



**МІЖНАРОДНИЙ НАУКОВИЙ ЖУРНАЛ**  
**INTERNATIONAL SCIENTIFIC JOURNAL**  
**МЕЖДУНАРОДНЫЙ НАУЧНЫЙ ЖУРНАЛ**

*Свидетельство  
о государственной регистрации  
печатного средства массовой информации  
КВ № 20971-10771Р*

*Сборник научных трудов*

Выпуск 6

2 том

Киев 2016

ББК 1  
УДК 001  
М-43

В журнале опубликованы научные статьи по актуальным проблемам современной науки.  
Материалы публикуются на языке оригинала в авторской редакции.  
Редакция не всегда разделяет мнения и взгляды авторов. Ответственность за достоверность фактов, имен, географических названий, цитат, цифр и других сведений несут авторы публикаций.  
При использовании научных идей и материалов этого сборника, ссылки на авторов и издания являются обязательными.

© Авторы статей, 2016  
© Международный научный журнал, 2016

Полное библиографическое описание всех статей Международного научного журнала представлено в: НЭБ Elibrary.ru, Polish Scholarly Bibliography.

Журнал зарегистрирован в международных каталогах научных изданий и наукометрических базах данных: РИНЦ; Open Academic Journals Index; ResearchBib; Scientific Indexing Services; Turkish Education Index; Electronic Journals Library; Staats- und Universitätsbibliothek Hamburg Carl von Ossietzky; RePEc; InfoBase Index; International Institute of Organized Research; CiteFactor; Open J-Gate, Cosmos Impact Factor.

### *Редакционная коллегия*

Главный редактор: **Коваленко Дмитрий Иванович** — кандидат экономических наук, доцент

Заместитель главного редактора: **Золковер Андрей Александрович** — кандидат экономических наук, доцент

Заместитель главного редактора: **Безверхий Константин Викторович** — кандидат экономических наук, доцент

Глава редакционной коллегии: **Тарасенко Ирина Алексеевна** — доктор экономических наук, профессор

Член редакционной коллегии: **Чабан Виталий Васильевич** — доктор технических наук, профессор

Член редакционной коллегии: **Румянцев Анатолий Александрович** — доктор технических наук, профессор

Член редакционной коллегии: **Сергейчук Олег Васильевич** — доктор технических наук, профессор

Член редакционной коллегии: **Беликов Анатолий Серафимович** — доктор технических наук, профессор

Член редакционной коллегии: **Мельник Виктория Николаевна** — доктор технических наук, профессор

Член редакционной коллегии: **Наумов Владимир Аркадьевич** — доктор технических наук, профессор (Российская Федерация)

Член редакционной коллегии: **Луценко Игорь Анатольевич** — доктор технических наук, профессор

Член редакционной коллегии: **Пахрутдинов Шукриддин Илесович** — доктор политических наук, профессор (Республика Узбекистан)

Член редакционной коллегии: **Степанов Виктор Юрьевич** — доктор наук по государственному управлению, профессор

Член редакционной коллегии: **Дегтярь Андрей Олегович** — доктор наук по государственному управлению, профессор

Член редакционной коллегии: **Дегтярь Олег Андреевич** — доктор наук по государственному управлению, доцент

Член редакционной коллегии: **Колтун Виктория Семеновна** — доктор наук по государственному управлению, доцент

Член редакционной коллегии: **Щербан Татьяна Дмитриевна** — доктор психологических наук, профессор, Заслуженный работник образования Украины, ректор Мукачевского государственного университета

Член редакционной коллегии: **Цахаева Анжелика Амировна** — доктор психологических наук, профессор (Российская Федерация, Республика Дагестан)

Член редакционной коллегии: **Сунцова Алеся Александровна** — доктор экономических наук, профессор, академик Академии экономических наук Украины

Член редакционной коллегии: **Денисенко Николай Павлович** — доктор экономических наук, профессор, член-корреспондент Международной академии инвестиций и экономики строительства, академик Академии строительства Украины и Украинской технологической академии

Член редакционной коллегии: **Кухленко Олег Васильевич** — доктор экономических наук, профессор

Член редакционной коллегии: **Чубукова Ольга Юрьевна** — доктор экономических наук, профессор

Член редакционной коллегии: **Драган Елена Ивановна** — доктор экономических наук, профессор

Член редакционной коллегии: **Захарин Сергей Владимирович** — доктор экономических наук, старший научный сотрудник, профессор

Член редакционной коллегии: **Лойко Валерия Викторовна** — доктор экономических наук, профессор

Член редакционной коллегии: **Скрипник Маргарита Ивановна** — доктор экономических наук, профессор

Член редакционной коллегии: **Селиверстова Людмила Сергеевна** — доктор экономических наук, профессор

Член редакционной коллегии: **Ефименко Надежда Анатольевна** — доктор экономических наук, профессор

Член редакционной коллегии: **Мигус Ирина Петровна** — доктор экономических наук, профессор

Член редакционной коллегии: **Смолин Игорь Валентинович** — доктор экономических наук, профессор

Член редакционной коллегии: **Шинкарук Лидия Васильевна** — доктор экономических наук, профессор, член-корреспондент Национальной академии наук Украины

Член редакционной коллегии: **Гоблик Владимир Васильевич** — доктор экономических наук, кандидат философских наук, доцент, Заслуженный экономист Украины

Член редакционной коллегии: **Заруцкая Елена Павловна** — доктор экономических наук, профессор

Член редакционной коллегии: **Вдовенко Наталия Михайловна** — доктор экономических наук, профессор

Член редакционной коллегии: **Копилюк Оксана Ивановна** — доктор экономических наук, профессор

- Член редакционной коллегии: **Ниценко Виталий Сергеевич** — доктор экономических наук, доцент
- Член редакционной коллегии: **Шевчук Ярослав Васильевич** — доктор экономических наук, старший научный сотрудник, доцент
- Член редакционной коллегии: **Мухсинова Лейла Хасановна** — доктор экономических наук, доцент (Российская Федерация)
- Член редакционной коллегии: **Задерей Петр Васильевич** — доктор физико-математических наук, профессор
- Член редакционной коллегии: **Байчоров Александр Мухтарович** — доктор философских наук, профессор (Республика Беларусь)
- Член редакционной коллегии: **Ильина Антонина Анатольевна** — доктор философских наук, доцент
- Член редакционной коллегии: **Сутужко Валерий Валериевич** — доктор философских наук, доцент (Российская Федерация)
- Член редакционной коллегии: **Стеблук Всеволод Владимирович** — доктор медицинских наук, профессор криминалистики и судебной медицины, Народный Герой Украины, Заслуженный врач Украины
- Член редакционной коллегии: **Щуров Владимир Алексеевич** — доктор медицинских наук, профессор, главный научный сотрудник лаборатории коррекции деформаций и удлинения конечностей (Российская Федерация)
- Член редакционной коллегии: **Свиридов Николай Васильевич** — доктор медицинских наук, главный научный сотрудник отдела эндокринологической хирургии, руководитель Центра диабетической стопы
- Член редакционной коллегии: **Иоелович Михаил Яковлевич** — доктор химических наук, профессор (Израиль)
- Член редакционной коллегии: **Сопов Александр Валентинович** — доктор исторических наук, профессор (Российская Федерация)
- Член редакционной коллегии: **Свинухов Владимир Геннадьевич** — доктор географических наук, профессор (Российская Федерация)
- Член редакционной коллегии: **Сенотрсова Светлана Валентиновна** — доктор биологических наук, доцент (Российская Федерация)
- Член редакционной коллегии: **Кузава Ирина Борисовна** — доктор педагогических наук, доцент
- Член редакционной коллегии: **Бабина Юлианна Ивановна** — докторантка (Республика Молдова)
- Член редакционной коллегии: **Коньков Георгий Игоревич** — кандидат технических наук, профессор
- Член редакционной коллегии: **Рамский Андрей Юрьевич** — кандидат экономических наук, профессор
- Член редакционной коллегии: **Чаленко Надежда Владимировна** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Бугас Наталия Валериевна** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Русина Юлия Александровна** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Беялов Талат Энверович** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Бадзым Александр Сергеевич** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Квасова Ольга Петровна** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Сибирянская Юлия Владимировна** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Щепанский Эдуард Валерьевич** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Криволапов Василий Сергеевич** — кандидат экономических наук, доцент (Российская Федерация)
- Член редакционной коллегии: **Саньков Петр Николаевич** — кандидат технических наук, доцент
- Член редакционной коллегии: **Артюхов Артем Евгеньевич** — кандидат технических наук, доцент
- Член редакционной коллегии: **Филева-Русева Красимира Георгиева** — кандидат психологических наук, доцент (Республика Болгария)
- Член редакционной коллегии: **Баула Ольга Петровна** — кандидат химических наук, доцент
- Член редакционной коллегии: **Вавилова Елена Васильевна** — кандидат сельскохозяйственных наук, доцент (Российская Федерация)
- Член редакционной коллегии: **Вицентий Александр Владимирович** — кандидат математических наук, доцент (Российская Федерация)
- Член редакционной коллегии: **Мулик Екатерина Витальевна** — кандидат наук по физическому воспитанию и спорту, доцент
- Член редакционной коллегии: **Олейник Анатолий Ефимович** — кандидат юридических наук, профессор
- Член редакционной коллегии: **Химич Ольга Николаевна** — кандидат юридических наук
- Член редакционной коллегии: **Фархитдинова Ольга Михайловна** — кандидат философских наук

**ЗМІСТ****CONTENTS****СОДЕРЖАНИЕ****ТЕХНИЧЕСКИЕ НАУКИ**

<b>Ковалюк Захар Дмитриевич, Юрценюк Сидор Прокопович, Семенчук Иван Иванович</b> ОБ ИНДУКТИВНО-ПОДОБНОМ ЭФФЕКТЕ В СУПЕРКОНДЕНСАТОРЕ .....	7
<b>Stepanov A. V.</b> AN AUTOMOBILE SAFETY: PROBLEMS AND PERSPECTIVES .....	11
<b>Унгул Володимир Валерійович, Проценко Михайло Михайлович</b> МЕТОД ПОЛІПШЕННЯ ЯКОСТІ ЗОБРАЖЕННЯ ВІДБИТКІВ ПАЛЬЦІВ ЗА ДОПОМОГОЮ ФІЛЬТРА ГАБОРА.....	15
<b>Унгул Володимир Валерійович</b> АНАЛІЗ ТА РОЗРОБКА МЕТОДІВ ДЕОБФУСКАЦІЇ СКРИПТІВ ДЛЯ ВИЯВЛЕННЯ ЗАГРОЗ ІНФОРМАЦІЙНОЇ СТІЙКОСТІ КОМП'ЮТЕРА .....	19
<b>Ханенко Олександр Андрійович, Медведський Андрій Миколайович</b> СИНХРОНІЗАЦІЯ ФАЙЛІВ І РЕЗЕРВНЕ КОПІЮВАННЯ ЗА ДОПОМОГОЮ RSYNC.....	28
<b>Ханенко Олександр Андрійович</b> ОГЛЯД МОЖЛИВОСТЕЙ PYTHON ЯК ЗАСОБУ ДЛЯ СТВОРЕННЯ ВЕБ-ПАРСЕРА.....	31
<b>Шеренковський Артем Олегович</b> АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СТВОРЕННЯ ІНТЕРНЕТ-РАДІОМОВЛЕННЯ.....	34
<b>Шилова Тетяна Олександрівна</b> ПРИНЦИПИ ПРОЕКТУВАННЯ МІСЬКИХ І ЗАМІСЬКИХ ДОРІГ .....	37
<b>Якимець Роман Вікторович, Яременко Костянтин Миколайович</b> МАСШТАБУВАННЯ НАВАНТАЖЕННЯ WEB-ДОДАТКІВ .....	45
<b>Якимець Роман Вікторович</b> МЕТОДИ КЛАСТЕРИЗАЦІЇ ТА ЇХ КЛАСИФІКАЦІЯ.....	48

<b>Якимів Йосип Васильович, Люта Наталя Вікторівна, Бортняк Олена Михайлівна</b> ВПЛИВ ТЕПЛА ТЕРТЯ ПОТОКУ НА ТЕМПЕРАТУРНИЙ РЕЖИМ ПЕРЕКАЧУВАННЯ ВИСОКОВ'ЯЗКИХ НАФТ .....	51
<b>Яковець Михайло Вікторвич</b> ФОРМУВАННЯ РЕКОМЕНДАЦІЙ НА ОСНОВІ МОДЕЛІ ПРИХОВАНИХ ФАКТОРІВ .....	54
<b>Яременко Костянтин Миколайович</b> РОЗПІЗНАВАННЯ ЖЕСТІВ РУКИ.....	57
<b>Бондаренко ОлексійСергійович, Івченко Дмитро Анатолійович</b> ТЕХНОЛОГІЇ АНАЛІЗУ РІВНЯ ГЛЮКОЗИ В ОРГАНІЗМІ ЛЮДИНИ .....	60
<b>Бондаренко Олексій Сергійович</b> МОБІЛЬНІ ДОДАТКИ ДЛЯ МЕДИЦИНИ.....	66
<b>Сергеев Єгор Ігорович</b> РОЗРОБКА КРОС-ПЛАТФОРМНИХ ДОДАТКІВ З ВИКОРИСТАННЯМ ФРЕЙМВОРКА XAMARIN .....	70
<b>Піпіч Артем Андрійович</b> КОНФЛІКТИ В КОНЦЕПТУАЛЬНИХ ПРОШАРКАХ КОНТЕКСТНО-ЗАЛЕЖНИХ СИСТЕМ.....	74

**Ковалюк Захар Дмитриевич**

*д-р физ.-мат. наук, профессор*

**Юрценюк Сидор Прокопович**

*главный конструктор*

**Семенчук Иван Иванович**

*аспирант*

*Институт проблем материаловедения им. И.Н. Францевича*

*Национальной Академии наук Украины, Черновицкое отделение*

**Zakhar D. Kovalyuk**

*Dr. Phys. & Math. Sc., Professor*

**Sidor P. Yurtsenyuk**

*chief designer*

**Ivan I. Semenchuk**

*postgraduate*

*Institute of Materials Science, I.M. Frantsevich*

*National Academy of Sciences of Ukraine, Chernivtsi Department*

## ОБ ИНДУКТИВНО-ПОДОБНОМ ЭФФЕКТЕ В СУПЕРКОНДЕНСАТОРЕ

### ABOUT THE INDUCTIVE-LIKE EFFECTS IN THE SUPERCAPACITOR

**Аннотация.** В работе исследуются пористые углеродные материалы, полученные методом пиролиза из органического сырья растительного происхождения активированного в среде гидроксида калия, в качестве электродной компоненты суперконденсаторов (СК). Обнаружен эффект, проявление которого подобно поведению индуктивности в цепи переменного тока.

**Ключевые слова:** суперконденсатор, растительное сырье, индуктивность, электрод.

**Summary.** Porous carbon materials are investigated as an electrode component of supercapacitors (SC). The materials are obtained from vegetable organic raw materials by using a technique of pyrolysis with a subsequent activation in a potassium hydroxide solution. We have found an effect of which is similar to that of inductance in a.c. circuits.

**Keywords:** supercapacitor, plant material, inductance, electrode.

Научный и практический интерес к накопителям электрической энергии на базе суперконденсаторов (СК) обусловлен отсутствием альтернативных устройств, заполняющих нишу между классическими химическими аккумуляторами и электролитическими конденсаторами. Основой формирования электродов СК является активный электродный материал, в качестве которого используются в настоящее время различные пористые материалы для достижения высоких значений эффективной внутренней поверхности, необходимой для достижения высокой удельной емкости [1].

В настоящее время наиболее широко для этой цели используются активированные нанопористые угле-

родные материалы [2]. В отдельную группу можно выделить активированные объемно-пористые углеродные материалы, полученные пиролизом органического сырья растительного происхождения. Благодаря многообразию доступности, экологической чистоте, естественной воспроизводимости, веществом растительного происхождения в данное время уделяется повышенное внимание со стороны разработчиков СК [3].

С целью выбора наиболее эффективно работающих в качестве поляризационной компоненты электродов СК нами проводятся активированных объемно-пористых углеродных материалов, получаемых методом термической деструкции разных видов растительного сырья [4].



Пиролиз (карбонизацию) растительного сырья проводили в кварцевой реторте в невысоком вакууме (~10 Па) при постоянной откачке реакционного объема на протяжении 50–60 мин, при температуре 650–660 °С. Применяется метод резкого нагрева (термоудар).

Реторту, загруженную сырьем и откачанную до рабочего давления, помещали в предварительно нагретую до рабочей температуры трубчатую электропечь. На всем протяжении процесса рабочий объем находился под постоянной откачкой. По завершении процесса карбонизации реторту извлекали из печи и при постоянной откачке поддерживали естественному охлаждению до комнатной температуры.

Процесс активации проводили при температуре ≈900 °С в щелочной среде. Карбонизованный материал пропитывали 30% водным раствором щелочи – КОН, в открытой ампуле загружали в реторту рабочий объем которой откачивали до остаточного давления 50–70 Па. Затем реторту, не отключая от вытяжной системы, помещали в предварительно нагретую до рабочей температуры трубчатую электропечь. Время выдержки в печи определяли по остаточному давлению 3–4 Па (зависит от рабочего объема реторты и количества загруженного материала). Охлаждение – как при карбонизации.

Исследования основных качественных характеристик полученных активированных материалов проводили на макетных образцах СК с водным раствором электролита (6М КОН). В разборной ячейке помещали два электрода из чистого (без добавок) углеродного материала весом по 0,02 г. Электроды разделены пропитанной раствором электролита сепараторной бумагой БАХИТ-48 толщиной 48–52 мкм изготовленной из асбеста АХО-2 по ТУ 21-22-3-81 в два слоя.

Процесс циклирования (заряд/разряд) проводили на установках SERIES2000 BATTERY TEST SYSTEM фирмы MACCOR (США), обеспечивающей программные задания различных режимов циклирования, а также время переключения ≤ 0,01 с.

Исследуя пористые углеродные материалы, полученные методом пиролиза из органического сырья растительного происхождения активированного в среде гидроксида калия, в качестве электродной компоненты суперконденсаторов (СК), обнаружен эффект, проявление которого подобно поведению индуктивности в электроцепи переменного тока. В моменты переключений от заряда на разряд (или наоборот) происходит выброс напряжения на величину ΔU выше заданной величины зарядного напряжения (или – наоборот ниже минимального значения разрядного напряжения), что аналогично наличию в цепи индуктивности [5, 6].

Проведенные измерения зависимости величины ΔU от величины тока заряд/разряда (при заряд/раз-

ряде постоянным током  $I_3 = I_p = \text{const}$ ) указывают на уменьшение ΔU при возрастании I, причем эта зависимость носит линейный характер и при некотором значении I ΔU=0. При дальнейшем увеличении I скачек напряжения происходит в обратную сторону, что подобно падению напряжения на внутреннем сопротивлении СК. При переходе от разряда к заряду наблюдается аналогичная ситуация с обратным знаком.

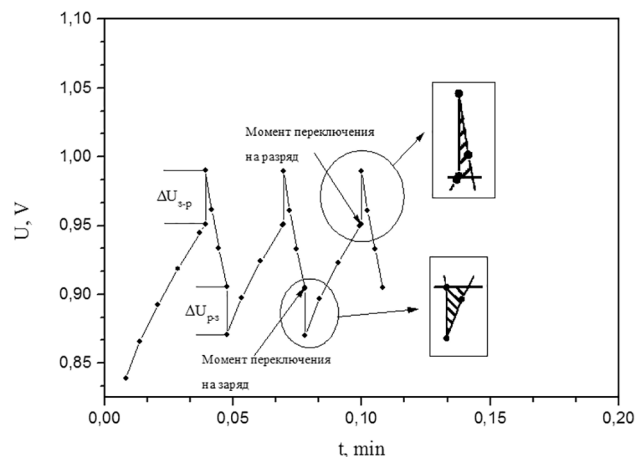


Рис. 1: Циклирование СК с водным раствором электролита (6М КОН) с электродами полученного пористого углеродного материала: заряд  $I_3 = 0,002$  А, разряд  $R_H = 300$  Ом

На рис. 1 представлено циклограмму заряд/разряда СК с водным раствором электролита (6М КОН) и электродами по 0,032 г пористого углеродного материала в режиме: заряд током  $I_3 = 0,002$  А, разряд на постоянную нагрузку  $R_H = 300$  Ом. Как видно из рис. 1, СК заряжается до заданного напряжения 0,95 В, но в момент переключения на разряд наблюдается скачек напряжения выше заданного на величину  $\Delta U_{з-р}$ . Аналогично при переходе от разряда на заряд происходит скачек напряжения, ниже минимального напряжения до которого разрядился СК, на величину  $\Delta U_{р-з}$ .

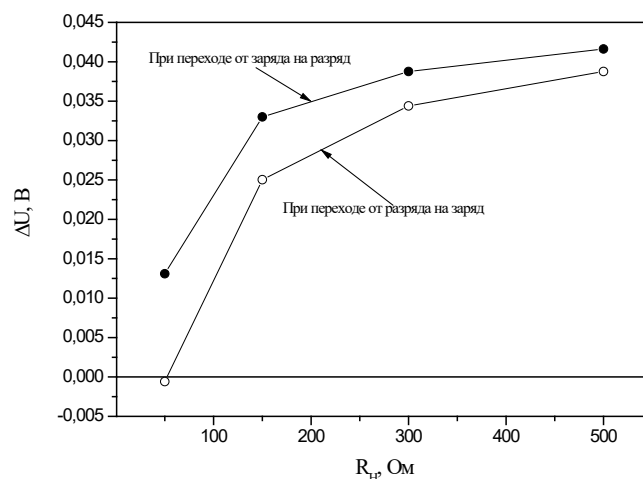


Рис. 2: Зависимость  $\Delta U_{з-р}$  и  $\Delta U_{р-з}$  от величины  $R_H$

На рис. 2 представлено зависимости  $\Delta U_{з-р}$  и  $\Delta U_{р-з}$  от величины  $R_H$ . Как видно, эти зависимости имеют нелинейный характер, так как при разряде СК на  $R_H=const$  напряжение на нем уменьшается по экспоненциальному закону. При перерасчете на соответствующие значения мгновенных токов эти зависимости приобретают линейный характер.

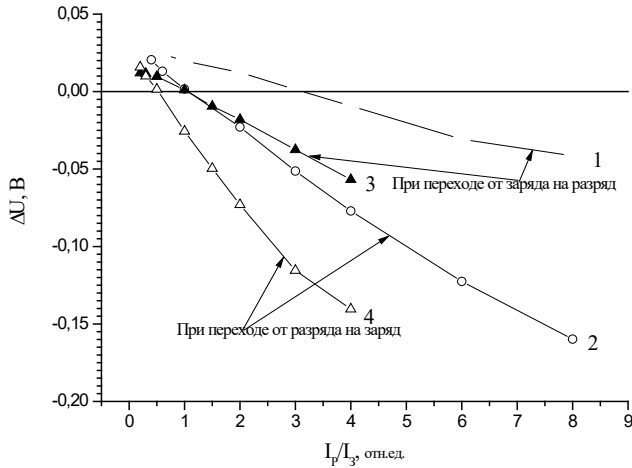


Рис. 3: Зависимость  $\Delta U_{з-р}$  и  $\Delta U_{р-з}$  от соотношения  $I_p/I_3$ .

На рис. 3 представлено зависимости  $\Delta U_{з-р}$  и  $\Delta U_{р-з}$  от соотношения  $I_p/I_3$ . Как видно из рис. 3 при изменении соотношения  $I_p/I_3$  изменяется наклон к оси абсцисс.

Заряд и разряд СК проводили также в режиме постоянного тока  $I_3=const, I_p=const$ . Однако, в моменты переключений этот режим нарушается, так как происходит изменение направления тока, а время такого изменения не может быть бесконечно малым. Каким бы коротким не было время переключения, оно имеет свою протяженность. Обозначим время переключения от заряда на разряд (или наоборот) —  $\Delta t_{пер}$ . За это короткое время в электроцепи, в которой находится испытуемый СК, происходит изменение направления тока. Такой процесс адекватный переменному току некоторой частоты. Напряжение в электроцепи, при включении СК на разряд, согласно законам электрических цепей должно распределиться на сопротивлении нагрузки и на внутреннем сопротивлении СК. Однако, как видно из эксперимента, происходит резкий скачок напряжения выше «э.д.с.» СК (в данном случае «э.д.с.» СК означает заданное максимальное значение напряжения до которого зарядился СК). Этот эффект наблюдается как при переключении на разряд в режиме  $I_3 = I_p = const$ , так и при переключении на разряд на большую нагрузку (1 КОм). Кроме того, как видно из рис. 1, разряд СК начинается не с величины напряжения соответствующей «э.д.с.» СК, а с точки соответствующей напряжению — «э.д.с.» СК+ $\Delta U$  и при различных режимах разряда напряже-

ние на СК уменьшается до значения «э.д.с.» СК на протяжении различного времени — от нескольких секунд (при  $I_3 = I_p = const$ ) до 1–2 мин (при  $R_H = const$ ). То есть, происходит накопление энергии выше заданного значения на емкостной составляющей. Если предположить, что данный эффект носит индуктивный характер, то эту излишнюю энергию (см. вставку рис. 1) можно описать уравнением:

$$W = \frac{LI^2}{2} \tag{1}$$

При заряд/разряде постоянным током ( $I=const$ ) из (1), учитывая, что  $U=f(t)$  получим:

$$L = \frac{2}{I} \int U(t) dt \tag{2}$$

Согласно (2) проведена оценка величины индуктивности проявляемой в моменты переключений от заряда на разряд и наоборот. Таким образом, получены значения индуктивности при переходе от заряда на разряд  $L_{з-р} \approx 4 \cdot 10^{-2}$  Н, при переходе от разряда на заряд  $L_{р-з} \approx 9,7 \cdot 10^{-2}$  Н.

Данные значения индуктивности получены из результатов измерений в режиме: заряд постоянным током, разряд на постоянную нагрузку. При этом режиме в процессе разряда происходит изменение как величины напряжения на СК, так и изменение величины разрядного тока. Поэтому затруднительно установить зависимость скачка напряжения от величины тока. Разность в величине индуктивности  $L_{з-р}$  и  $L_{р-з}$  может быть обусловлена разностью  $\Delta U_{з-р}$   $\Delta U_{р-з}$ , т.е. в (2) необходимо учитывать, что

$$\Delta U_{з-р} = U(t) = \Delta U_u - \Delta U_r \tag{3}$$

где  $\Delta U_u$  — скачек напряжения за счет индуктивности,  $\Delta U_r$  — падение напряжения на активном сопротивлении СК.

В случае перехода от разряда на заряд имеем

$$\Delta U_{р-з} = U(t) = \Delta U_u + \Delta U_r \tag{4}$$

Таким образом, для каждого режима измерений будем иметь конкретные значения  $\Delta U_u$  и  $\Delta U_r$ . Так для проведенного выше расчета  $L_{з-р}$  и  $L_{р-з}$  без учета  $\Delta U_r$  среднее значение  $L = 6,85$  Н, что соответствует, в рамках допустимых ошибок, значению полученному с использованием формул (3) и (4).

Таким образом, формулу (2) следует записать в виде:

$$L = \frac{2}{I} \int (\Delta U_u \pm \Delta U_r) dt \tag{5}$$

Из (5) видно, что при увеличении тока разряда уменьшается вклад индуктивности и в некоторый момент наступает ситуация, когда вклад падения напряжения на активном сопротивлении СК становится

преобладающим. Тогда рассматривается как бы «классический» случай: распределение падений напряжений на активном сопротивлении СК и внешней нагрузки ( $R_{\text{н}}$ ). Но проведенные в данной работе исследования, показали, что такой подход приводит к ошибочному определению внутреннего сопротивления СК.

#### Литература

1. В.Е Conway, Electrochemical supercapacitors. Scientific fundamentals and technological applications (N. Y.: Kluwer Academic / Plenum Publ.: 1999).
2. Бухаров В. А., Ковалюк З. Д., Нетяга В. В., Юрценюк С. П. Углеродный материал из растительного сырья для электродов суперконденсаторов // Электрохимическая энергетика. — 2008. — Т. 8, № 2. — С. 111–114.
3. Kovalyuk Z. D., Yurtsenyuk S. P., Buharov V. A., Savchuk A. I. New electrode materials for supercapacitors // E-MRS IUMRS ISEM 2006 Spring Meeting (Nice, France). — 2006.
4. D. Angin, Production and characterization of activated carbon from sour cherry stones by zinc chloride, Fuel 115 (2014) 804–811.
5. Компан М. Е., Малышкин В. Г., Кузнецов В. П., Микрюкова М. А. Проявление индуктивной компоненты импеданса пористого интерфейса суперконденсатора // Письма в ЖТФ, 2013, Т. 39, В.11, С. 80–86.
6. Ковалюк З. Д., Юрценюк С. П., Семенчук И. И. Наблюдение индуктивно-подобного эффекта в суперконденсаторе // Письма в ЖТФ, 2016, Т. 42, В.6, С. 91–96.

**Stepanov A. V.**  
*PhD in Technical Sciences,*  
*Associate Professor,*  
*Kharkiv National Automobile and Highway University*

## AN AUTOMOBILE SAFETY: PROBLEMS AND PERSPECTIVES

**Abstract.** *The article examines special aspects of the concept of motor vehicle safety and the concept of transport safety. The problems and prospects of motor vehicle safety were given herein. A wide range of internal and external threats, which impair transport safety of the country were observed. Along with the concept of transport safety threat, a classification of transport safety threats was introduced. It was concluded, that further development of transport system and provision of its stable operation in conditions of transport safety refer to strategic tasks of the State.*

**Key words:** *automobile transport, motor vehicle safety, transport safety, threats.*

**Introduction.** Automobile transport is one of the most important sectors of the national economy that serves as a kind of circulatory system in a complex organism of the country. Besides motor vehicles play a major role in the economy. As an integral and extremely substantial part of the infrastructure, automobile transport allows for delivery of export and import cargoes of various purposes in the interest of international cultural and technical cooperation, as well as integration into the world economy. It accounts for more than a half of the passenger traffic.

In recent years, the problem of traffic safety has aggravated. Due to the large number and mobility of motor vehicles, the problems of automobile transport safety are acute. There is a tendency to rapid deterioration of the situation, which requires an urgent response. According to general estimates of the United Nations Economic Commission for Europe (UNECE) Inland Transport Committee experts, the minimum estimated loss from death and injury of people, damage and destruction of vehicle cargo is 3–4% of the value of the gross national product [10].

The current views and knowledge of the issue of motor vehicle safety allow making conclusions about the perception and understanding of the problem, thus causing a change in the paradigm of traffic safety. Consequently, vehicle safety is a serious social and economic problem. Further development of the transport system and provision of its stable operation in conditions of transport safety refer to strategic tasks of the State. The solution to this problem should be complex and should take into account the possibilities of all links of the “driver – vehicle – road – environment” system.

The rapid growth of car park that takes place in recent years significantly increased the loading level on the road network and all transport infrastructure, which were not designed for such a number of vehicles, drivers and pedes-

trians we observe these days. As a result of this situation, the number of people dying each year in car accidents has increased by about 45% between 1994 and 2004. Over the last decade, more than 2 million people were killed and more than 17 million were injured. [6]

The category of vehicle safety was studied from various points of view in scientific literature, therefore this definition has different meanings [6–10, 12, 13]. For instance, investigation of transport safety through the basic principles of vehicle efficiency and reliability was presented in the works of V. Ya. Anilovich, O. V. Bazhinova, V. M. Varfolomeev, E. S. Wentsel, E. M. Getsovych, M. Ya. Govoruschenko, O. S. Grynchenko, I. E. Dyumin, M. S. Zhdanovsky, I. V. Kragelsky, V. G. Kukhtov, A. T. Lebedev, V. V. Nichke, M. A. Podryhalo, O. S. Polyansky, A. A. Staroselsky, A. M. Turenko, V. N. Tkachov and of many other authors. It should be noted, that despite the large number of general works, statements and studies, there is still not enough research on the complexity and contradiction of modern motor vehicle safety.

**The aim of the study, problem statement.** The aim of this study is to consider the problems of motor vehicle safety and prospects for their solutions. To achieve this goal it is necessary to examine the automobile transport safety improvement, the transport safety concept and possible threats to transport safety.

**Research materials.** Providing safety on different types of vehicles became one of the priority requirements to transport systems nowadays. Active safety systems for automobiles, namely components and assemblies, which prevent traffic accidents: brake systems, steering control, lighting and visibility systems, information systems, antispin and anti-lock braking systems, tires and so on, are being constantly improved. An intensive development and prevalent adoption of passive safety systems happen,

namely devices that limit or even eliminate the negative consequences of an accident, such as seat belts, front and side airbags, collapsible steering tubes and pedal boxes, fire resistible fuel systems, folding mirrors, robust designs of door locks, door handles, etc. Active use of the above implements allows to reduce the severity of consequences and the number of deaths in car accidents in some Western European countries [14].

The issue of road safety overtook a special place among the main problems of the entire world. The following complex decisions related to effective organization of traffic are applied in European countries, Japan, USA and Canada: replacement old traffic lights with light-emitting-diode (LED) ones, introduction of traffic lanes for “slow” vehicles, installation of locators to separate oncoming flows, systematization of routes for different types of transport, implementation of high technologies of road accidents prevention. For example, in Europe and the US connected to vehicles GPS-tracking systems are widely used. In the event of a car accident, the tracker records the vehicle position, the number of passengers, strength and geometry of the crush, air bags operation and sends this information to emergency services. In their turn, car manufacturers try to make their cars as safe as possible both for passengers and for other road users [14].

While considering the transport safety we must distinguish between the concepts of “motor vehicle safety” and “traffic safety”. In many languages to refer to both of the concepts the same word “security” is used. When applied to the motor vehicle industry as a whole, these two terms are explained as follows [11].

We must understand motor vehicle safety as a level of safety or a collection of measures to prevent illegal activities (fraud, attacks on drivers and vehicles, theft of goods and cars) as well as terrorist attacks. And the traffic safety implies a level of safety and collection of measures to prevent accidents on public roads that lead or do not lead to property damage, injuries or fatalities. Both of these states are of great political significance and the situation in this field bears extreme importance for the population of the country.

The motor vehicle safety measures and traffic safety measures should not be done in parallel and in isolation one from the other. Programs to improve the situation in these two areas should be implemented together in order to get mutual benefits, given the obvious connection between them.

It should also be borne in mind that there is no zero risk in natural environment and it is impossible to guarantee complete safety. However, the automobile industry is strongly interested in safety advancement. States and their authorities have the leading role in its provision [1–5]. Resourcefulness and active participation of the

motor vehicle sector is an essential element of ensuring the success of any measures aimed at improving safety. It is necessary to strengthen cooperation between the public and private sectors in safety providing, which can be highly effective.

The currently existing means of safety and effectiveness ensuring, which provide benefits of the both should be used to the maximum extent. These funds include the TIR system of the United Nations (Customs Convention on the International Transport of Goods under Cover of TIR Carnets – the international agreement adopted in 1975 under the auspices of the European Economic Commission of the United Nations. The goal of the Convention is the creation of goods transporting system, General Transit system of EU and of other international legal instruments that aim to simplify procedures of goods registration during borders crossing [11].

It is obvious that safety policy should be based on relevant information. Rational and effective measures of safety improvement should be based on reliable information and understanding of the situation in the field of international criminality and terrorism, and recording of safety-associated risks. Herewith it is always necessary to clarify terminology. For example, when it comes to the notion of “transport safety” acts of unlawful interference in transportation system should be mentioned, since it is a wider term than anti-terrorist threat. For instance, a false message about a bomb causes evacuation of thousands of people and schedule or flight delays.

Today, the concept of transport safety is mainly interpreted as transport terrorism prevention. The anti-terrorist imperative of transport safety is objective and generally triggered by a significant increase of terrorist attacks in the world, as well as the degree of its danger directly to the transport sector. The dramatic events of a number of acts of terror in recent years where vehicles filled with explosives were used as a weapon by terrorists. However, given the fact that protection of an individual, society and the state from terrorism, including in the transport sector, was announced a priority task nowadays, we should keep in mind that preventing and countering terrorist attacks on transport is only a part of the problem of transport safety providing in the country as a whole. Its additional integral fragment is the protection of transport sector from other fields, including criminal forms of unlawful interference with transport, and also from various kinds of emergencies.

In the most general understanding, the concept of “transport safety” [13] can be defined as: system of prevention, opposition and suspension of crimes, including terrorism, in the transport sector; system of avoidance of natural and human-caused emergencies in the transport sector; system of prevention or minimization of material

and moral damages in transport sector from crimes and emergencies; system aimed at increasing environmental safety of transportation and ecological sustainability of the transport system; system of implementation of national security purposes of in the transport sector as a whole.

The systemic nature of the concept of transport safety determines the need for a comprehensive, systemic solution to the problems existing in this area.

Transport safety aims to protect passengers, owners, addressees and carriers of goods, owners and operators of vehicles, transport sector and its workers, the economy and budget of the country as well as environment from threats in the transport sector.

Transport security is designed to ensure safe conditions of transportation for life and health of passengers; goods, load and cargo transportation safety; performance and operation safety of transport facilities and means; economic safety, including foreign economic; environmental safety; information security; fire safety; sanitary safety; chemical, biological, nuclear and radiation safety; mobilization readiness of the transport industry sectors.

Large-scale range of different reasons of natural, technical and social nature causes existence of a wide variety of internal and external threats that weaken transport security of the country. Transport safety threats refer to illegal actions or intentions to commit such acts and natural or man-made processes or their combination that obstruct the implementation of vital interests of an individual, society and the state in the transport sector, causing or may cause accidents in transport complex [13]. Transport safety threats are classified in a number of aspects: according to the degree of importance, the nature of threats, the areas and types of manifestation and so on.

The main threats to transportation include: terrorist acts and sabotage (hijacking or commandeering of aircrafts, sea and river vessels, railway trains, vehicles, explosions at railway stations and public roads, sabotage against hydrotechnical constructions, etc.); other cases of unlawful interference with the functioning of transport (imposing foreign objects on the rails, breaking of

railroad devices, telephone "terrorism", illegal blocking of airports and major highways) that threaten the safety of passengers' lives and health, bring direct harm to the transport sector and generate negative socio-political, economic and psychological effects in society; criminal actions against passengers; criminal actions against goods and cargoes; events (accidents), caused by malfunction of technical systems of transport (depreciation, degradation, disrepair, high accident rate, imperfection), violation of operation rules of technical systems, including regulatory requirements for environmental safety during transportation, as well as by natural factors, create an alarm situation and entail material losses and casualties.

The threats should include the negative consequences of insufficient elaboration of normative legal framework governing relations in the transport sector, as well as imperfections in law application.

**Conclusions.** The situation in transport safety interacts with the fundamental interests of individuals, society and the state as well as interests of all physical and legal entities involved in the transport sector. Therefore, the questions of transport safety advancement cannot be of interest to public authorities only, and must concern everyone. Only unified forces are able to provide a stable and strong safety, including motor vehicle safety. The State cannot and should not solve this problem alone, because it requires significant physical, financial and human resources. The society has to participate actively in the financing of transport safety tasks execution, and each person in particular should strengthen his own moral position and maintain law and order, should respect lives and rights of others and as a result each one will be able to feel safe.

Effective and successful work of specialized organizations and entities of transport infrastructure as to the introduction and implementation of motor vehicles safety requirements on automobile transport is only possible upon condition of further improvement of the existing regulatory legal framework through its integration into the international security system.

### References

1. Law of Ukraine “On Transport”. – VVR – 1994.
2. Law of Ukraine “On the Automobile Transport”. Journal of Verkhovna Rada of Ukraine (VVR), 2001, N22, 105) {As amended by Law N3492-IV as of February 23, 2006, VVR, 2006, N32, art.273} with additions and amendments.
3. Decree of the President of Ukraine “On the Regulations of the State Inspectorate of Ukraine on Land Transport Safety, as amended by Presidential Decree N506/2013 as of September 11, 2013.
4. Resolution of the Cabinet of Ministers of Ukraine dated October 30, 2008 № 1384-r “On Approval of the Concept of the State Program of Traffic Safety Advancement for Years 2009–2012”. – Kyiv: Council, 2008.
5. Resolution of the Cabinet of Ministers of Ukraine dated October 20, 2010 № 2174 “On Approval of the Transport Strategy of Ukraine for the Period till 2020”.
6. Traffic Safety of Pedestrians: Overview information / Kyiv: SIC of Traffic Police Ministry of Internal Affairs of Ukraine, 2010. – Vol. 12 – page 28.
7. Belyi O. V. Concept of Integrated Traffic Safety/ Belyi O. V., D. A. Skorokhodov, A. L. Starichenkov // Miscellany of studies of All-Russian Scientific Conference “Transport of Russia: Problems and prospects. – 2009 “. Moscow: MSUT, 2009. – с. II-5 – I-6.
8. Ibadulayev V. A. Evaluation of Risks during Hazardous Cargoes Transportation / V. A. Ibadulayev, A. L. Starichenkov, I. V. Stepanov // Transport: Science, Technology, Management. Miscellany of general information, № 8, 2004.
9. A. V. Kurbatova. Prognostication of Transport Systems Development: Ideology, tools, calculations / Kurbatov A. V., Kuznetsova E. Yu. – Ekaterinburg: USTU, 2000. – page 185.
10. Provision of Traffic Safety in the countries of Asia and Pacific Region: General information. – Moscow: SIC Traffic Police the Ministry of Internal Affairs of Russia, 1997. – Vol. 4 – page 48.
11. Access: [www.iru.org](http://www.iru.org). – Date of application: April 02, 2015.
12. D. A. Skorokhodov. Motor Vehicle Safety Problems / D. A. Skorokhodov, A. L. Starichenkov // Transport Security and Technology. № 2 (3), 2005. – pages 24–27.
13. I. E. Surkhachev. “Transport Safety” / I. E. Surkhachev. – 2007. – page 270.
14. Proceedings of the Sixteenth Scientific and Technical Conference “Safety Systems” – SS-2007 of International Information Forum, October 25, 2007, Moscow. – Moscow: Academy SBS, 2007. – page 235.

**Унгул Володимир Валерійович**

*студент*

*Національний технічний університет України «Київський політехнічний інститут»*

**Проценко Михайло Михайлович**

*студент*

*Національний технічний університет України «Київський політехнічний інститут»*

**Унгул Владимир Валерьевич**

*студент*

*Национальный технический университет Украины «Киевский политехнический институт»*

**Проценко Михаил Михайлович**

*студент*

*Национальный технический университет Украины «Киевский политехнический институт»*

**Unhul Volodymyr V.**

*Student*

*National Technical University of Ukraine «Kyiv Polytechnic Institute»*

**Protsenko Mikhail M.**

*Student*

*National Technical University of Ukraine «Kyiv Polytechnic Institute»*

## МЕТОД ПОЛІПШЕННЯ ЯКОСТІ ЗОБРАЖЕННЯ ВІДБИТКІВ ПАЛЬЦІВ ЗА ДОПОМОГОЮ ФІЛЬТРА ГАБОРА

## МЕТОД УЛУЧШЕННЯ КАЧЕСТВА ИЗОБРАЖЕНИЯ ОТПЕЧАТКА ПАЛЬЦА С ПОМОЩЬЮ ФИЛЬТРА ГАБОРА

## METHOD OF IMPROVING IMAGE QUALITY FINGERPRINT USING GABOR FILTER

**Анотація.** В даній статті розглянуто алгоритм покращення якості вхідного зображення заснований на фільтрах Габора. Сформульована задача формалізації методів підбору параметрів фільтра Габора та запропоновано рішення для неї.

**Ключові слова:** поліпшення зображень, фільтр Габора, нормалізація зображення, локальні ознаки, відбитки пальців.

**Аннотация.** В данной статье рассмотрен алгоритм улучшения качества входного изображения, который основан на фильтрах Габора. Сформулирована задача формализации методов подбора параметров фильтра Габора и предложены решения для нее.

**Ключевые слова:** улучшение изображений, фильтр Габора, нормализация изображения, локальные признаки, отпечатки пальцев.

**Summary.** In this article the algorithm for improving the quality of the input image based on Gabor filter. The problem formalization methods of selection options Gabor filter and suggested solutions to it.

**Key words:** Adjusting images, Gabor filters, image normalization, local features, fingerprints.

### Вступ

Автоматичне співставлення відбитків пальців ґрунтується на порівнянні локальних ознак і їх комбінації для виконання ідентифікації особистості. Важливим етапом в зіставленні відбитків пальців є автоматично і достовірно отримати ознаки з вхідних

зображень відбитків пальців, що є важким завданням. Виконання алгоритму вилучення ознак залежить більшою мірою від якості вхідного зображення відбитка пальця.

Для забезпечення необхідної надійності виконання алгоритму вилучення ознак, незалежно від якості



зображення відбитка пальця, необхідний додатковий алгоритм, який покращує якість зображення. Вхідне зображення відбитка пальця в залежності від типу сканера може представлятися у вигляді бінарного зображення або ж зображення в градаціях сірого кольору (градієнтне зображення).

$$VAR = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (I(i, j) - M)^2 .$$

На Рис. 1. наведено приклад вхідного і нормалізованого зображення.

### Локальні і глобальні характеристики відбитка пальця

У кожному відбитку пальця можна визначити два типи ознак — глобальні та локальні. Глобальні ознаки — ті, які можна побачити неозброєним оком.

Інший тип ознак — локальні. Це локальні особливості папілярних ліній унікальні для кожного відбитка точки. Їх виділення пов'язано з тим, що лінії відбитків пальців не є прямими. Вони часто зламані, розгалужені, змінюють напрямок і мають розриви. Точки, в яких лінії закінчуються, розгалужуються або змінюють напрямок, називаються точками мінуціями. Ці точки забезпечують унікальну інформацію про відбиток пальця при ідентифікації особистості. Кожен відбиток містить до 70 мінуцій.

### Поліпшення якості зображення відбитку пальця за допомогою використання фільтра Габора

Фільтр Габора — лінійний фільтр, імпульсна перехідна характеристика якого представляється у вигляді добутку функції Гауса на гармонійну функцію:

$$g(x, y) = s(x, y) \cdot w_r(x, y) ,$$

де —  $s(x, y)$  комплексна синусоїда, а —  $w_r(x, y)$  огибає Гауса для двовимірного простору. Застосування фільтра Габора для зображень здійснюється за п'ять етапів.

#### Етап 1. Нормалізація зображення.

Нормалізація зображення необхідна для того, щоб поставити попередні середні значення і відхилення. Нормалізоване зображення  $G$  визначається як зображення, де  $G(i, j)$  — значення нормалізованої яскравості пікселя з координатами  $(i, j)$ . Нормалізоване зображення розраховується виходячи із середнього і середньоквадратичного відхилення вихідного зображення:

$$G(i, j) = \begin{cases} M_0 + \sqrt{\frac{VAR_0 \cdot (I(i, j) - M)^2}{VAR}} , \text{ якщо } I(i, j) > M \\ M_0 - \sqrt{\frac{VAR_0 \cdot (I(i, j) - M)^2}{VAR}} , \text{ инакше} \end{cases}$$

де  $M_0$  і  $VAR_0$  — задані значення середнього і середньоквадратичного відхилення відповідно,  $i$  — вихідні значення середнього і середньоквадратичного відхилення, обчислюються за формулами:

$$M = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i, j) ,$$



Рис. 1. Приклад вхідного (а) і нормалізованого (б) зображення образу відбитка пальця.

#### Етап 2. Розрахунок орієнтаційного зображення.

Орієнтаційне зображення  $O$  представляє собою матрицю  $N \times N$ , в якій кожна компонента  $O(i, j)$  показує локальну орієнтацію (кут нахилу в даній точці) лінії з координатами  $(i, j)$ .

$$O(i, j) = \frac{1}{2} \arctan \left( \frac{d_x^2(i, j) d_y^2(i, j)}{2 d_x(i, j) d_y(i, j)} \right)$$

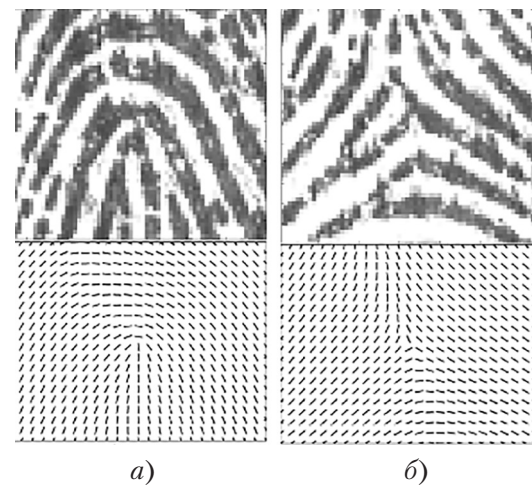


Рис. 2. Орієнтаційні зображення фрагментів відбитка пальця. На малюнку а) зображений фрагмент центру, на малюнку б) фрагмент дельти.

#### Етап 3. Розрахунок частотного зображення.

Частотне зображення являє собою матрицю розміру  $N \times N$ , в якій кожна компонента  $F(i, j)$  показує локальну частоту ліній в даній точці, яка визначається як частоту гребенів, спрямованих уздовж орієнтації виступу. У разі, якщо в якійсь точці неможливо визначити чітку синусоїдально-окреслену хвилю (наприклад, через наявність особливих точок в цих

координатах), частота визначається як середня величина частоти в сусідніх блоках.



Рис. 3 Приклад блоків, в яких неможливо чітко визначити чітку синусоїдально-окреслену хвилю.



Рис. 4. Хвильове представлення ліній в осередку

Розрахунок частоти в точці з координатою  $(i, j)$  розраховується наступним чином: якщо  $\lambda$  — кількість пікселів між двома сусідніми гребенями в блоці розмірністю  $W \times W$ , і центр блоку — піксель з координатами  $(i, j)$ , то частота в даній точці буде розраховуватися як  $F(i, j) = \frac{1}{\lambda}$ .

**Етап 4. Бінаризація зображення.**

Зображення  $V$  визначається як зображення, якщо кожен піксель приймає одне з двох можливих значень — нуля одиниці.

Одиниця відповідає гребеню відбитка, нуль — западині:

$$R(i, j) = \begin{cases} 1, & \text{если } G(i, j) > B_0 \\ 0, & \text{иначе} \end{cases}$$

де  $B_0$  — поріг маскування, а  $G(i, j)$  — інтенсивність пікселя зображення.

**Етап 5. Застосування до бінарним зображенню фільтрів Габора.**

Фільтр налаштовується на локальну орієнтацію виступів, застосовується до пікселів виступів і западин зображення.

$$H(i, j : \phi, f) = \exp \left\{ -\frac{1}{2} \left[ \frac{x_\phi^2}{\delta_x^2} + \frac{y_\phi^2}{\delta_y^2} \right] \right\} \cos(2\pi f x_\phi),$$

Де  $x_\phi = i \cdot \cos \phi + j \cdot \sin \phi$ ;  $y_\phi = -i \cdot \sin \phi + j \cdot \cos \phi$ ;  $\phi$  — Орієнтація фільтра Габора,  $f$  — частота, а  $\delta_x$  й  $\delta_y$  — просторові константи обвідної Гаусса уздовж осей  $x$  і  $y$  відповідно.

Для використання Фільтра Габора нам необхідно знати значення наступних величин:

- 1) напрямлення фільтра;
- 2) частоту синусоїдальної площинний хвилі;
- 3)  $\delta_x$  і  $\delta_y$  — середньоквадратичні відхилення обвідної Габора.

Частотна характеристика фільтра визначається з локальної частоти  $f$  виступів, напрямок визначається локальної орієнтацією. Значення  $\delta_x$  і  $\delta_y$  можна задати при реалізації алгоритму.

Чим більше будуть ці значення, тим більше фільтр буде стійкий до шумів, але, в той же час, буде вносити більше спотворень, створюючи неіснуючі виступи і западини. Якщо вибрати значення  $\delta_x$  і  $\delta_y$  низькими, фільтр не вноситиме спотворень, але його здатність фільтрувати значно знизиться, що призведе до неефективного усунення шумів.

Тому при підборі значень  $\delta_x$  і  $\delta_y$  намагаються знайти компроміс між ефективністю фільтра і відсутністю внесених фільтром спотворень. Як правило ці параметри підбираються емпіричним шляхом.



Рис. 5 Приклад відбитку пальця, обробленого фільтром Габора.

**Вибір параметрів алгоритму, математична постановка завдання**

Завдання: підбір параметрів для оптимальної роботи алгоритму з умови забезпечення якості фільтрації і збереження ключової інформації.

$M = \{m_1, \dots, m_k\}$  — множина ключових точок в зображенні;

$E = \{e_1, \dots, e_k\}$  — множина помилок в зображенні.

Необхідно вибрати оптимальний набір параметрів для фільтра зображення, заснованого перетворенні Габора. Буде проводиться підбір параметра, середньоквадратичної обвідної Гаусс, використовуваної в фільтрі.

Кожному значенню  $\delta$  ставляться у відповідність значення функцій  $P_1(\delta)$  і  $P_2(\delta)$ .

$P_1(\delta)$  — Функція ймовірності помилки першого роду, то є ймовірність внесення фільтром спотворень в ключову інформацію

$P_2(\delta)$  — Функція ймовірності помилки другого роду, тобто ймовірність ігнорування, тобто Не видалені шумів в зображенні.

Якщо прийняти  $f$  за якусь функцію, яка є лінійною комбінацією 2-х параметрів:

$$f(\delta) = a_1 P_1(\delta) + a_2 P_2(\delta)$$

де  $a_1$  і  $a_2$  – вагові коефіцієнти, то завдання можна сформулювати наступним чином – необхідно знайти таке значення параметра  $\delta$ , що  $f(\delta) = \min, \delta \in R$ .

#### Рішення поставленого математичного завдання

Для вирішення поставленого в роботі математичної задачі необхідно визначити діапазон значень, в якому буде проводитися підбір параметру  $\delta$  і кількість кроків підбору  $N$ .

Нехай  $[\delta_{\min}, \delta_{\max}]$  – обраний діапазон значень, тоді

$$\Delta = \frac{\delta_{\max} - \delta_{\min}}{N} \text{ – величина кроку. Значення } \delta \text{ на кроці}$$

і визначається за формулою:

$$\delta_i = \delta_{\min} + \frac{i}{N}(\delta_{\max} - \delta_{\min}), \quad i = \overline{1, N}.$$

Необхідно знайти таке  $i$ , при якому

$$f(\delta) = a_1 P_1(\delta) + a_2 P_2(\delta) \xrightarrow{\delta} \min.$$

Приймемо  $[\delta_{\min}, \delta_{\max}] = [0.3, 1.5]$ ,  $N = 24$ , величина

$$\text{кроку } \Delta = \frac{\delta_{\max} - \delta_{\min}}{N} = 0.05.$$

Результати експерименту показали, що значення функції  $f(\delta) = a_1 P_1(\delta) + a_2 P_2(\delta)$  мінімально при  $\delta = 1.05$ . На Рис. 6 проілюстрована робота алгоритму при різних значеннях параметра  $\delta$ .

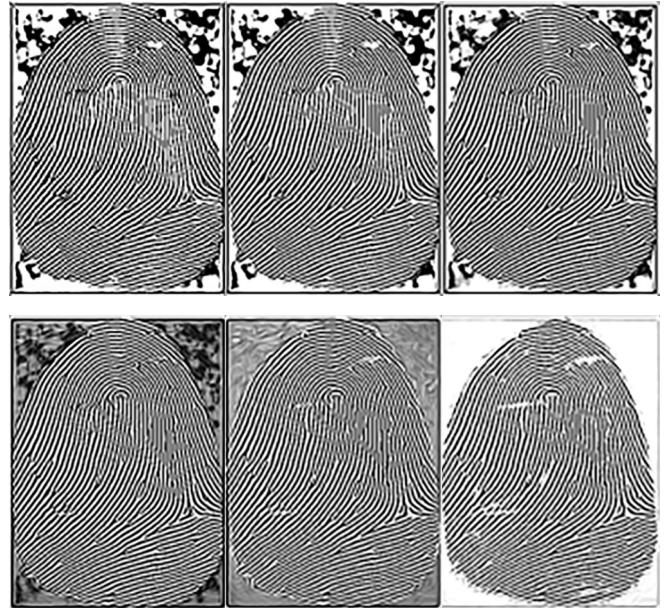


Рис. 6. Результати обробки зображення фільтром Габора при різних значеннях.

#### Висновки

У даній статті був оглянутий метод покращення якості вхідного зображення заснований на фільтрах Габора. Був детально вивчений математичний апарат даного методу, і розроблений алгоритм, який вирішує поставлене завдання. Так само була сформульована математична постановка задачі, яка полягає в формалізації методів підбору параметрів фільтра Габора.

#### Література

1. Maltoni D., Maio D., Jain A. K., Prabhakar S. Handbook of fingerprint recognition. – N. Y.: Springer, 2003. – 348 p.
2. Asker M. Bazen, Systematic methods for the computation of the directional fields and singular points of fingerprint/ Asker M. Bazen, Sabih H. Gerez // IEEE Transactions on pattern analysis and machine intelligence. – 2002 – № 7.
3. Hong L., Wan Y., Jain A. K. Fingerprint Image Enhancement: Algorithm and Performance Evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, vol. 20, pp. 777–789.
4. Yang J. W., Liu L. F., Jiang T. Z., Fan Y. A Modified Gabor Filter Design Method for Fingerprint Image Enhancement. Pattern Recognition Letters, 2003, vol. 24, no. 12, pp. 1805–1817.
5. Wang W., Li J., Huang F., Feng H. Design and Implementation of Log-Gabor Filter in Fingerprint Image Enhancement. Pattern Recognition Letters, 2008, vol. 29, no. 3, pp. 301–308.
6. Гонсалес Р. Цифровая обработка изображений. М.: Техносфера, 2005. 1072 с. [Gonzales R., Woods R. Tsifrovaya obrabotka izobrazheniy (Digital Processing of the Images). Moscow, Tekhnosfera Publ., 2005. 1072 p.]

**Унгул Володимир Валерійович**

*студент*

*Національний технічний університет України «Київський політехнічний інститут»*

**Унгул Владимир Валерьевич**

*студент*

*Национальный технический университет Украины «Киевский политехнический институт»*

**Unhul Volodymyr V.**

*Student*

## **АНАЛІЗ ТА РОЗРОБКА МЕТОДІВ ДЕОБФУСКАЦІЇ СКРИПТІВ ДЛЯ ВИЯВЛЕННЯ ЗАГРОЗ ІНФОРМАЦІЙНОЇ СТІЙКОСТІ КОМП'ЮТЕРА АНАЛИЗ И РАЗРАБОТКА МЕТОДОВ ДЕОБФУСКАЦИИ СКРИПТОВ ДЛЯ ВЫЯВЛЕНИЯ УГРОЗ ИНФОРМАЦИОННОЙ УСТОЙЧИВОСТИ КОМПЬЮТЕРА ANALYSIS AND DEVELOPMENT OF METHODS DEOBFUSKATSIYI SCRIPTS TO IDENTIFY THREATS TO INFORMATION COMPUTER SUSTAINABILITY**

**Анотація.** В роботі досліджуються проблемні питання в захисті інформаційних систем та пропонуються методи усунення цих проблем. Метою є покращення систем захисту комп'ютерної інформації від скриптових вірусів, які потрапляють до комп'ютера з мережі інтернет, а саме при перегляді WEB-сторінок із вбудованим вірусним програмним забезпеченням. Описані принципи створення та побудовані алгоритми протидії обфусцированим скриптовим вірусам.

**Ключові слова:** Обфускація, Деобфускація, Розшифровування, WEB-сторінки, JavaScript, скриптовий вірус.

**Аннотация.** В работе исследуются проблемные вопросы в защите информационных систем и предлагаются методы устранения этих проблем. Целью является улучшение систем защиты компьютерной информации от скриптовых вирусов, которые попадают к компьютеру из сети интернет, а именно при просмотре WEB-страниц с встроенным вирусным программным обеспечением. Описаны принципы создания и построены алгоритмы противодействия обфусцированным скриптовым вирусам.

**Ключевые слова:** Обфускация, Деобфускация, Расшифровка, WEB-страницы, JavaScript, скриптовый вирус.

**Summary.** It analyzes the problems in protecting information systems and proposes methods eliminate these problems. To improve the systems of information protection of scripting viruses that enter your computer from the Internet, such as when watching WEB-page in-viral software. Describes the creation and counter the algorithm obfustsyrovanyum scripting viruses.

**Key words:** Obfuskatsiya, Deobfuskatsiya, Decryption, WEB-page, JavaScript, script virus.

### **Вступ**

З кожним днем зростає кількість інформаційних комп'ютерних систем, які потребують захисту, надійності в обробці й використанні даних, разом із ними зростає кількість різновидів вірусів та вірусного коду, які можуть негативно вплинути або взагалі внести критичні зміни в роботу будь-якої інформаційної системи. В процесі виконання наукового дослідження винайдені та представлені принципово нові рішення по захисту інформаційної системи від зашифрованих скриптових комп'ютерних вірусів.

### **1. АНАЛІЗ ТА ОГЛЯД ПРОБЛЕМАТИКИ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ**

#### **1.1. Гармонізація понятійної бази**

В зв'язку з тим, що проблеми протидії зашифрованим скриптовим вірусам є досить новою, в науково-практичних джерелах є деяка розбіжність в термінології та поняттях. Тому на першому етапі досліджень проведена гармонізація понятійної бази.

Гармонізація базувалась на наступних принципах:

- Використані поняття та терміни повинні відповідати загальноновизнаним та не протирічити між собою.
- Використані поняття та терміни повинні відповідати меті дослідження.

В даній статті особлива увага приділяється обфускації програмного коду та веб-додатків написаних на скриптовій мові javascript. Обфускація — це приведення виконуваного програмного коду до виду, який зберігає його функціональність, але ускладнює аналіз, розуміння алгоритму роботи і модифікації при декомпіляції [1]. Іншими словами це заплутування коду. «Заплутування» коду може здійснюватися на рівні алгоритму, сирцевого тексту або асемблерного тексту. Існують спеціальні програми, що виробляють обфускацію, їх називають обфускаторами. Є деяка кількість шляхів обфускації, яка може реорганізувати програмний код в автоматичному режимі або в ручному.

Цілі обфускації:

- складнення декомпіляції та вивчення програм з метою виявлення функціональності;
- Ускладнення декомпіляції програм з метою запобігання обходу систем перевірки ліцензій;
- Порушення авторських прав програмістів і приховування авторства;
- Демонстрація неочевидних можливостей мови і кваліфікації програміста.

Останнім часом, в інтернет середовищі значно поширилась тенденція написання шкідливого коду і прихованість його за допомогою методів обфускації. Такі вірусні програми та скрипти легко обходять існуючі антивірусні засоби, тому що виявити шкідливий функціонал досить складно. Щоб дослідити такий код треба проводити деобфускацію. В деяких випадках її треба робити в декілька кроків, тому що код може бути обфусцированим декілька разів. Тобто вже обфусцирований код піддається новій обфускації і так далі.

Деобфускація — це процес, приведення обфусцированого коду до виду, який дозволить вивчати та аналізувати його функціонал. В даний момент часу, не має систем, які б змогли проаналізувати будь-який обфусцирований код, але існують деобфускатори (програми, які надають змогу проаналізувати функціонал або привести код до зрозумілого виду), які «заточені» тільки під певний різновид обфускації. Метою дослідження є можливість аналізувати будь-який обфусцирований код, незважаючи, якими засобами та методами він був обфусцирован.

### 1.2. Методи шифрування (обфускації) скриптовій мові JavaScript

Існує дуже велика кількість методів обфускації, яких із кожним днем стає дедалі більше. Ми спробували їх розділити на три категорії.

#### 1.2.1. Обфускація мінімізаторами

Існують готові програмні засоби, які надають змогу в автоматичному режим зробити «зтискання» коду. Таким видом обфускації користуються в більшій мірі

виробники програмного забезпечення. Код, який отримуємо на виході, можна зашифрувати ще одним методом, які будуть розглянуті пізніше в нашій роботі.

Серед загальновідомих мінімізаторів [2] можна виділити JS Packer, JSmin, YUI Compressor, Closure compiler та це далеко не повний їх список.

Написаний код на скриптовій мові JavaScript виду:

```
function MyClass(){
  this.foo = function(argument1, argument2){
    var addedArgs = parseInt(argument1)+parseInt(argument2);
    return addedArgs;
  }
  var anonymousInnerFunction = function(){
    // do stuff here!
  }
}
```

За допомогою мінімізаторів приймає такий вигляд:  
function MyClass(){this.foo=function(c, b){var d=parseInt(c)+parseInt(b); return d}; var a=function(){};};

або такий:

```
var _0xd799=["\x66\x6F\x6F"]; function MyClass() {this[_0xd799[0]]=function (_0xefca2,_0xefca3) {var _0xefca4=parseInt(_0xefca2)+parseInt(_0xefca3); return _0xefca4;}; var _0xefca5=function(){};};
```

або такий:

```
eval(function(p, a, c, k, e, d){e=function(c){return c}; if(!'.replace(/^\/, String))while(c-){d[c]=k[c]|c} k=[function(e){return d[e]}; e=function(){return"\w+"; c=1; while(c-){if(k[c]){p=p.replace(new RegExp('\b'+e(c)+'\b','g'), k[c])}}return p} ('40="35!";92(1){6(1+"\7"+0)}2("8");',10,10,'a|msg|MsgBox|Hello|var|World|alert|n|OK|function'.split('|'),0,{})).
```

З першого погляду отриманий код важко проаналізувати та все ж таки не існує нічого неможливого. Подібну обфускацію можна провести і в ручному режимі, що, можливо, навіть ускладнить його деобфускацію.

#### 1.2.2. Методи Braifuck

Такий підхід перетворює код до невпізнання [3]. Навіть незрозуміло, що перед нами виконуваний JavaScript скрипт.

Мізерний скрипт alert(0) можна перетворити в такий вид:

```
([][(!![]+[])!+[]!+[]!+[]!+[]!+(!![]+[][(!![]+[])[+[]]+(!![]+[]+[][[[]])!+[]+[]+[]]+(!![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]]+(!![]+[])[!+[]+!+[]]+(!![]+[])[!+[]+!+[]]]
```

[+!+[ ]+[ +[] ]+(!![ ]+[ ])[+!+[ ]]+(!![ ]+[ ])  
[ +[] ][( [ ][( [ ]+[ ])[ +[] ]+( [ ]+[ ]+[ [ ] ] )  
[+!+[ ]+[ +[] ]+(!![ ]+[ ])[!+[ ]+[ ]+(!![ ]+[ ])  
[ +[] ]+(!![ ]+[ ])[!+[ ]+[ ]+[ ]+[ ]+(!![ ]+[ ])  
[+!+[ ]]+( [ ]+[ ])[!+[ ]+[ ]+[ ]+[ ]+( [ ]+[ ])  
[+!+[ ]]+( [ ]+[ ])[!+[ ]+[ ]+( [ ]+[ ])[!+[ ]+[ ]]  
( [ ]+[ ])[+!+[ ]+( [ ]+[ ])[!+[ ]+[ ]+( [ ]+[ ])  
[!+[ ]+[ ]+[ ]+( [ ]+[ ])[+!+[ ]+( [ ]+[ ])(+ [ ] );  
або в такій:

```
° ω° )= / ~ m´ ) ) ~—— / /*∇ `*/ [ ' _ ] ; o=( ° - ° )  
= _=3; c=( ° Θ° )=( ° - ° )-( ° - ° ); ( ° Д° )=( ° Θ° )= (o^_o)/  
(o^_o);( ° Д° )=( ° Θ° : ' _ ' ; ° ω° ) : (( ° ω° )==3) + ' _ ' ] [ ° Θ° ]  
, ° - ° ) : ( ° ω° )+ ' _ '[o^_o-( ° Θ° )] ; ° Д° ) : (( ° - ° ==3) + ' _ ' )  
[ ° - ° ] ]; ( ° Д° ) [ ° Θ° ] =( ° ω° )==3) + ' _ ' [c^_o];( ° Д° )  
[ ' c ' ] = (( ° Д° )+ ' _ ' ) [ ( ° - ° )+ ( ° - ° )-( ° Θ° ) ];( ° Д° ) [ ' o ' ] =  
(( ° Д° )+ ' _ ' ) [ ° Θ° ] ;( ° o )=( ° Д° ) [ ' c ' ]+( ° Д° ) [ ' o ' ]+( ° ω° )  
+ ' _ '[ ° Θ° ]+ (( ° ω° )==3) + ' _ ' [ ° - ° ] + (( ° Д° ) + ' _ ' )  
[ ( ° - ° )+ ( ° - ° )]+ (( ° - ° ==3) + ' _ ' ) [ ° Θ° ]+( ° - ° ==3) + ' _ ' )  
[ ( ° - ° ) - ( ° Θ° )]+( ° Д° ) [ ' c ' ]+( ° Д° )+ ' _ ' [ ( ° - ° )+ ( ° - ° )]+  
( ° Д° ) [ ' o ' ]+( ° - ° ==3) + ' _ ' ) [ ° Θ° ] ;( ° Д° ) [ ' _ ' ] =(o^_o)  
[ ° o ] [ ° o ] ;( ° ε° )=(( ° - ° ==3) + ' _ ' ) [ ° Θ° ]+ ( ° Д° )  
. ° Д° )+( ° Д° )+ ' _ ' [ ( ° - ° ) + ( ° - ° )]+(( ° - ° ==3) + ' _ ' )  
[o^_o- ° Θ° ]+( ° - ° ==3) + ' _ ' ) [ ° Θ° ]+( ° ω° )+ ' _ ' ] [ ° Θ° ] ;  
( ° - ° )+( ° Θ° ) ; ( ° Д° ) [ ° ε° ]= '\\ ; ( ° Д° ). ° Θ° )=( ° Д° +  
° - ° )[o^_o-( ° Θ° )];( ° - ° o)=( ° ω° ) + ' _ ' ] [c^_o];( ° Д° )  
[ ° o ]=' \" ' ;( ° Д° ) [ ' _ ' ] ( ° Д° ) [ ' _ ' ] ( ° ε° + ° Д° ) [ ° o ]+  
( ° Д° ) [ ° ε° ]+( ° Θ° )+ ( ° - ° )+ ( ° Θ° )+ ( ° Д° ) [ ° ε° ]+( ° Θ° )+  
(( ° - ° )+( ° Θ° ))+( ° - ° )+( ° Д° ) [ ° ε° ]+( ° Θ° )+( ° - ° )+( ° - ° )  
+ ( ° Θ° )+( ° Д° ) [ ° ε° ]+( ° Θ° )+ ((o^_o) +(o^_o))+  
((o^_o) - ( ° Θ° ))+( ° Д° ) [ ° ε° ]+( ° Θ° )+ ((o^_o)  
+(o^_o))+ ( ° - ° )+( ° Д° ) [ ° ε° ]+( ° - ° )+( ° Θ° )+(c^_o)+  
( ° Д° ) [ ° ε° ]+(o^_o) +(o^_o))+ (c^_o)+ ( ° Д° )  
[ ° ε° ]+( ° - ° )+( ° Θ° ))+( ° Θ° )+( ° Д° ) [ ° o ] ( ° Θ° )) ( ' _ ' ) ;
```

де взагалі використовуються символи, яких немає навіть на розкладці клавіатури. Хоча з першого погляду все виглядає досить страшно та здебільшого до такого виду можна привести в автоматичному режимі. Існує деяка кількість Braifuck-подібних конверторів для JavaScript.

До даного виду обфускації також можна віднести написання коду на езотеричних мовах програмування таких як Whitespace, JAPH та інших.

### 1.2.3. Метод запечатування

В першому способі на виході ми отримували код схожий на JavaScript, у другому зовсім не схожий,

а метод запечатування його робить взагалі прихованим.

Обфусцирований код буде складатися з двох частин: видима частина (може використовуватися будь-який спосіб із вищеперерахованих для її обфускації) і прихована частина.

Реалізовується це таким чином: «Шкідливий код», який ми хочемо сховати, перетворюється на строку, яка складається зі знаків табуляції (біт 1) та пробілів (біт 0). Як результат, ми отримуємо в багато разів більше коду, ніж у нас було. Видима частина буде декодувати приховану частину та її виконувати: декодуватиме біти в числа, а числа перетворить в символ String.fromCharCode(), а далі виконає функцію eval. Приклад:

```
decodeAndEval(document.getElementById  
(«evilCode»).innerHTML);  
<div id=»evilCode»>  
</div>
```

До цього методу обфускації коду можна також віднести пакування скрипта JavaScript в CSS [4]. На сторінці користувача буде відображатися картинка, яка може бути навіть фоною, але при виконанні коду нашкочить вашому комп'ютеру. Такий спосіб шифровки реалізовується таким чином: відбувається зтискання коду та з'являється можливість реалізації інтелектуальних систем відновлення зображень винятково на растровій мові. Вказуючи просторове положення такого зображення-коду, ми можемо формувати дуже цікаві структуровані по принципу нарізання (tilling) бібліотеки практично безграничного розміру, високо-оптимізовані по трафіку та часу доступу до конкретної ділянки коду (рис. 1.1).

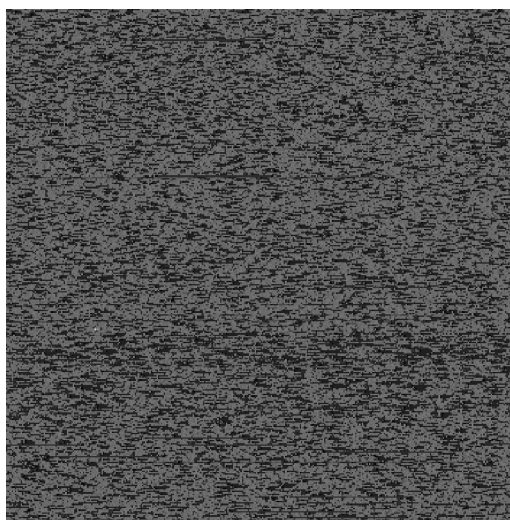


Рисунок 1.1. Приклад запечатаного скрипта в зображення

### 1.2.4. Узагальнена структурна схема шляхів обфускації

Із вище продемонстрованих та описаних категорій обфускації скриптів можна побудувати узагальнену структурну схему шляхів обфускації, яка зображена на рис. 1.2.

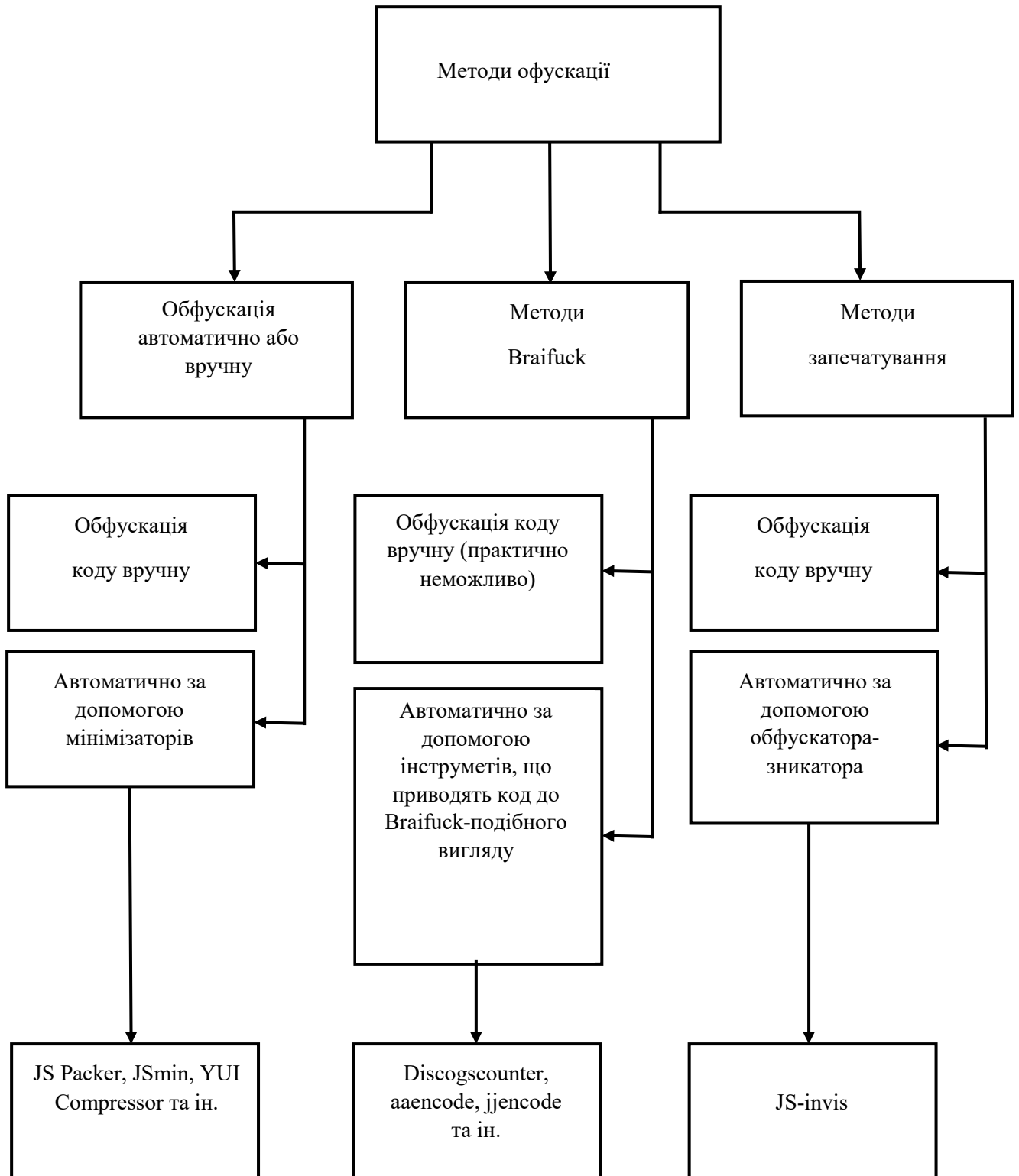


Рисунок 1.2. Узагальнена структурна схема шляхів обфускації

## 2. РОЗРОБКА МЕТОДІВ ДЕОБФУСКАЦІЇ JAVASCRIPT

Аналіз літературних джерел не виявив завершеного алгоритму деобфускації зашифрованого коду JavaScript. Знайдено тільки характерні приклади деобфускації. Тому на основі ідеї розшифрування скриптів та деяких запропонованих шляхів аналізу обфусцированого коду запропонуємо деякі методи дуобфускації JavaScript.

### 2.1. Розшифрування JavaScript, який було обфусцировано за допомогою мінімізатора

1) Визначаємо обфусцирований код JavaScript. Якщо ми хочемо дізнатися чи є на певній WEB-сторінці шкідливий код, нам треба її просканувати за допомогою інструментів.net fraimwork та виявити наявність JavaScript скриптів. При обфускації код може прийняти зовсім незрозумілий вигляд і втратити характерні ознаки та це не стане завадою для його коректного виконання на Вашому комп'ютері і як результат захований вірус може нанести чималу шкоду вашому електронному пристрою. Для визначення зашифрованого скрипта будемо використовувати типові приклади та результати роботи обфускаторів мінімізаторів. Маємо певний набір ключових слів та функцій, які вказують на те, що деякий фрагмент або фрагменти сторінки — це обфусцирований JavaScript.

2) Починаємо розбір зашифрованого коду. Завантажуємо підозрілий код у строкову змінну. Спочатку ініціалізуємо перший знайдений зашифрований фрагмент коду WEB-сторінки.

3) Аналізуємо підозрілий скрипт. Оскільки ми визначили, що шифрування відбувалося методом мінімізаторів, то для таких випадків характерним є упорядкування коду. Додаємо символ нового рядка після кожної крапки з комою.

4) Отримуємо імена функцій JavaScript. Для цього окремо скануємо кожний отриманий рядок. Об'явлені змінні в обфусцированому коді передаємо до функції alert(), для того щоб отримати реальні імена функцій JavaScript.

5) Аналізуємо отриманий функціонал на безпечність використання. Результатом попередніх дій стало виявлення функцій, які при певному наборі шляхом аналізу можуть дати відповідь на питання чи зашкодить виконуючий скрипт роботі нашої комп'ютерної системи.

6) Якщо на сторінці виявлено декілька зашифрованих скриптів, то результати першого просканованого коду зберігаємо у вихідний файл та переходимо знову до пункту 2, де розпочинаємо аналізування наступного підозрілого фрагменту.

7) Формування висновку про безпечність посилання. Тільки після того, як будуть розшифровані всі

підозрілі скрипти ми зможемо зробити висновок, наскільки є надійним певне посилання.

Для даного методу деобфускації розроблено наступний алгоритм, який зображено на рис. 2.1.

### 2.2. Розшифрування JavaScript, який було обфусцировано за допомогою BraiFuck-подібного методу

1) Визначаємо обфусцирований код JavaScript. Для того щоб визначити прихований JavaScript код BraiFuck-подібного вигляду треба використовувати xml-файл, який матиме в собі підбірку типових Braifuck-подібних скриптів та окремих символів.

2) Розпочинаємо розбір зашифрованого коду. Завантажуємо підозрілий фрагмент у строкову змінну.

3) Розшифруємо код. Запускаємо цикл FOR по всій довжині вмісту змінної.

4) За одну ітерацію оброблятимемо один символ.

5) Дістаємо значення зашифрованого тексту-коду. Як усім відомо, повернути значення певного Braifuck-подібного зашифрованого скрипта просто, слід лише використати правильний деобфускатор, але для цього треба виконати дуже глибокий аналіз. Тому пропонуємо вміст скрипту отримати універсальним способом. Визначаємо ASCII-код символа:

– Якщо код має значення між 36 і 61 (не включно), то додаємо до нього 25, а символ, відповідаючий результуючому коду, зберігаємо в нову змінну.

– Якщо код має значення між 61 і 86 (не включно), то віднімаємо від нього 25, а символ, відповідаючий результуючому коду, зберігаємо у ту ж таки нову змінну.

6) Декодуємо. Після того, як ми перейшли від Braifuck-подібного вигляду до більш зрозумілого нашій системі розпізнавання вірусів виду, отримане треба декодувати. Для цього нам треба реалізувати виклик функції eval() для запуску результату декодування.

7) Аналізуємо функціонал на безпечність використання. Можемо мати один із двох результатів:

– Якщо результатом деобфускації є код, в якому ми можемо проаналізувати функціонал, тобто ключеві слова, які визивають деякі функції, то переходимо до оцінки безпечності такого коду.

– Якщо результатом деобфускації є код, в якому проаналізувати функціонал не вдається, то треба перейти до попереднього методу деобфускації JavaScript, який було розглянуто в 2.1. Такий підхід демонструє, що обфускація була проведена у декілька кроків, де спочатку скористалися методом мінімізації, а потім методом приведення коду до Braifuck-подібного вигляду.

Для даного методу деобфускації розроблено наступний алгоритм, який зображено на рис. 2.2.



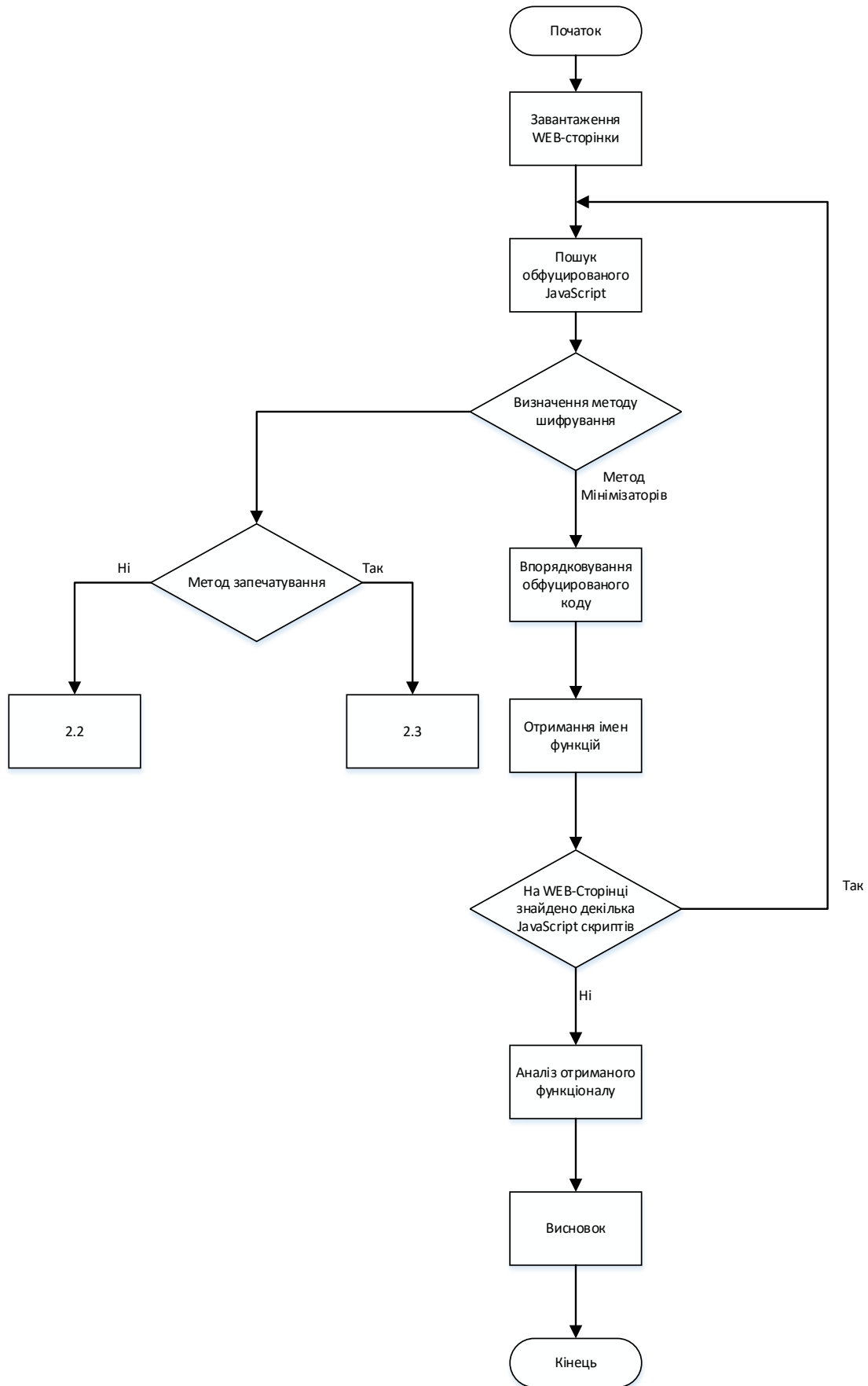


Рисунок 2.1. Алгоритм деобфускації зашифрованих скриптів мінімізаторами

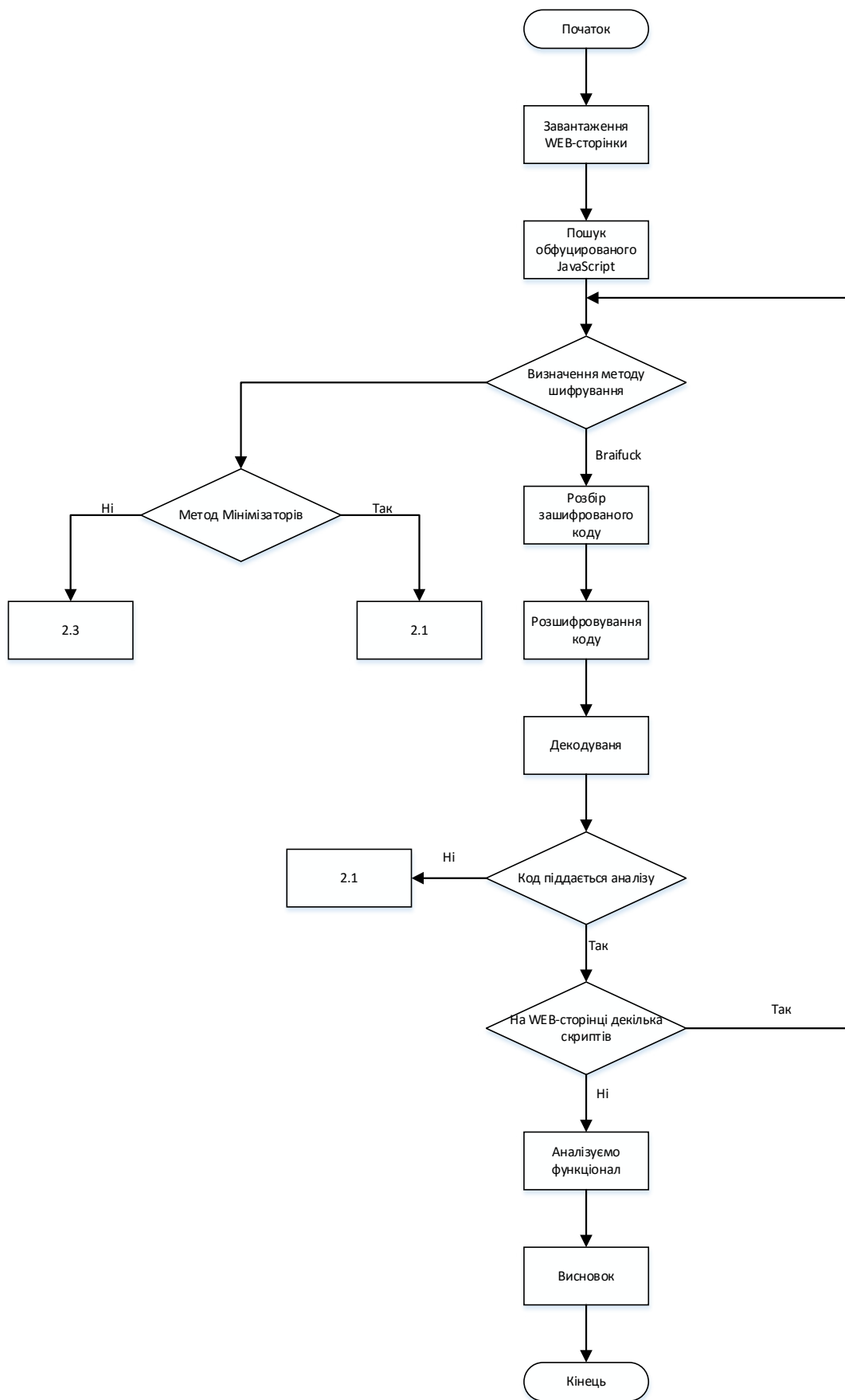


Рисунок 2.2. Алгоритм деобфускації зашифрованих скриптів Braifuck-подібного виду

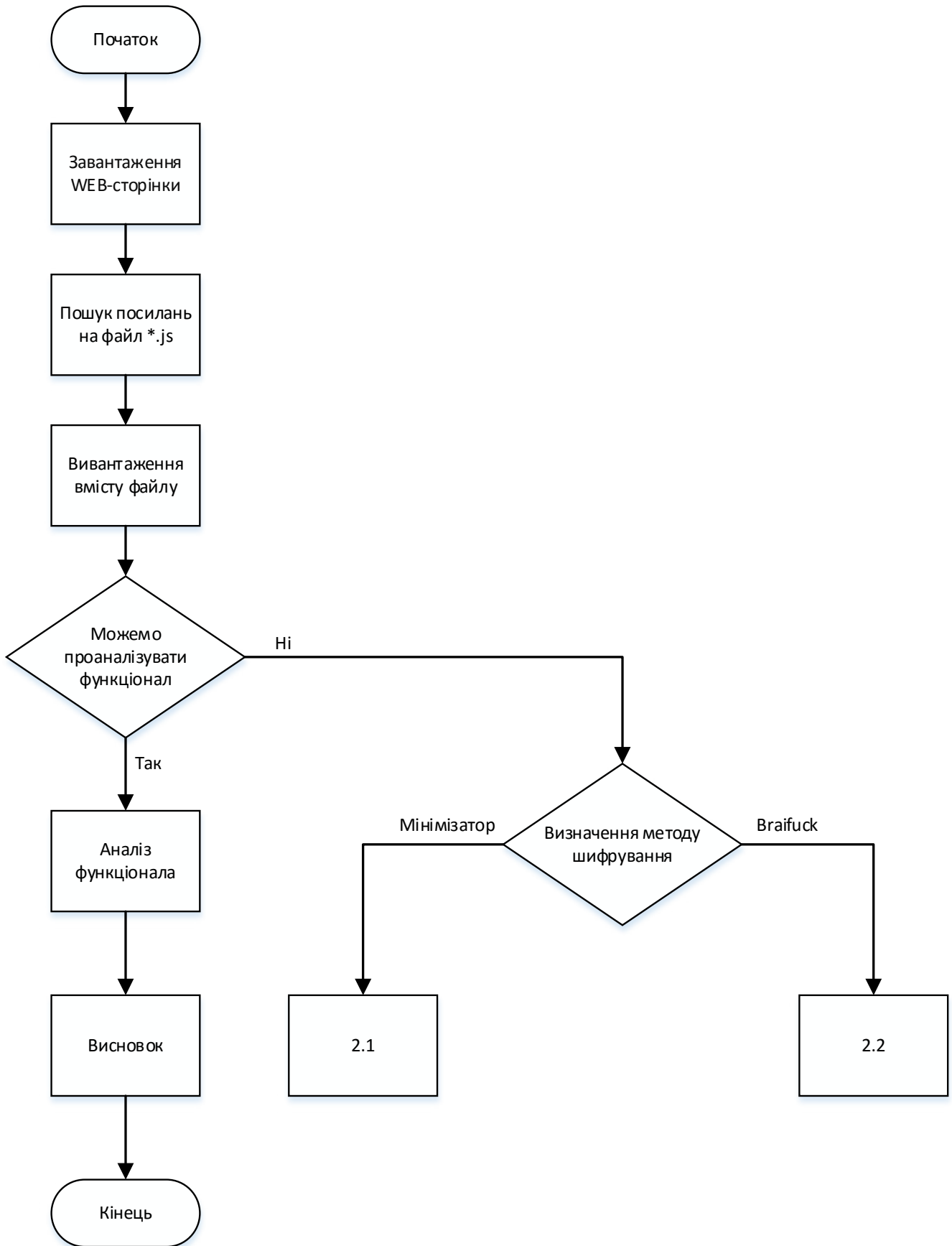


Рисунок 2.3. Алгоритм деобфускації зашифрованих скриптів методом запечатування

### **2.3. Розшифрування JavaScript, який було обфусцировано за допомогою методу запечатування**

1) Визначаємо посилання на файл JavaScript. Для цього скануємо сторінку і шукаємо посилання на файли із розширенням \*.js.

2) Розпочинаємо розшифрування коду. Копіюємо вміст файлу в змінну. Після чого розпочинається процес сканування отриманого коду.

3) Визначаємо чи можемо ми проаналізувати функціонал, який зберігається у файлі чи потрібно провести процес деобфускації.

4) Визначаємо метод обфускації або напямую проводимо аналіз функціоналу. Якщо неможливо провести оцінку надійності коду — значить він прихований, а тому можливо й небезпечний. Для цього треба використати методи розпізнавання обфусцированих скриптів із попередніх пунктів. У файлі може зберігатися обфусцирований код методом мінімізації або методом, який приводить до Braifuck-подібного вигляду. Тому визначимо який саме метод використовується та

перейдемо до розшифрування використовуючи методику запропоновану у пунктах 2.1 та 2.2.

5) Якщо на WEB-сторінці більше одного посилання на файл, то вони вистроюються в чергу і починаючи з пункту 2 проходять процес деобфускації.

Для даного методу деобфускації розроблено наступний алгоритм, який зображено на рис. 2.3.

### **Висновки**

Представлені алгоритми та методи деобфускації скриптів рекомендовано використовувати в програмному забезпеченні інтегрованому із браузерами, для того щоб виявляти вірусну активність на WEB-ресурсах в режимі онлайн. Звісно дані алгоритми потребують деякої модифікації для більш результативної роботи при їх впровадженні, але представлена ідея класифікації обфусцированого коду по категоріях є досить новою і надає можливість розробити універсальні підходи до аналізування та розшифрування скриптів.

### **Література**

1. Терейковський І. Нейронні мережі в засобах захисту комп'ютерної інформації / І. Терейковський. — К.: ПоліграфКонсалтинг. — 2007. — 198 с.
2. «Badass JavaScript»: інтернет блог, стаття: «Badass js is back white some badass obfuscation» — посилання: <http://badassjs.com/post/2929065287/obfuscation/>
3. «Adamcecc»: інтернет блог — посилання: <http://adamcecc.blogspot.com/>
4. «Ajaxian»: інтернет сайт, стаття «Want to pack JS and CSS really well? Convert it to a PNG and unpack it via Canvas», посилання: <http://ajaxian.com/>
5. Безруков Н. Н. Компьютерная вирусология / Н. Н. Безруков. — К. Инкомбук, 1990. — 450 с.
6. Біленчук П. Д. Комп'ютерна злочинність / П. Д. Біленчук, Б. В. Романюк. — К.: Атака, 2002. — 240 с.
7. Вилков А. С. Информационная безопасность персональных ЭВМ и мониторинг компьютерных сетей / А. С. Вилков. — М.: МИНИТ ФСБ России, 2005. — 210 с.

**Ханенко Олександр Андрійович**

*студент*

*Національний технічний університет України «Київський політехнічний інститут»*

**Медведський Андрій Миколайович**

*студент*

*Національний технічний університет України «Київський політехнічний інститут»*

**Ханенко Александр Андреевич**

*студент*

*Национальный технический университет Украины «Киевский политехнический институт»*

**Медведский Андрей Николаевич**

*студент*

*Национальный технический университет Украины «Киевский политехнический институт»*

**Khanenko O.**

*student*

*National Technical University of Ukraine “Kyiv Polytechnic Institute”*

**Medvedskyi A.**

*student*

*National Technical University of Ukraine “Kyiv Polytechnic Institute”*

## СИНХРОНІЗАЦІЯ ФАЙЛІВ І РЕЗЕРВНЕ КОПІЮВАННЯ ЗА ДОПОМОГОЮ RSYNC СИНХРОНИЗАЦИЯ ФАЙЛОВ И РЕЗЕРВНОЕ КОПИРОВАНИЕ С ПОМОЩЬЮ RSYNC FILE SYNCHRONIZATION AND BACKUP WITH RSYNC

**Анотація.** Мета даної роботи – розглянути можливості утиліти Rsync як засобу для синхронізації файлів та створення резервних копій на віддалених машинах.

**Ключові слова:** синхронізація, Rsync, резервне копіювання, віддалений доступ, SSH, Linux, GUI.

**Аннотация.** Цель данной работы – рассмотреть возможности утилиты Rsync как средства для синхронизации файлов и резервного копирования на удаленных машинах.

**Ключевые слова:** синхронизация, Rsync, резервное копирование, удаленный доступ, SSH, Linux, GUI.

**Summary.** The aim of this work is to consider the possibilities Rsync utility as a tool for file synchronization and backup on remote machines.

**Key words:** synchronization, Rsync, backup, remote access, SSH, Linux, GUI.

### Вступ

Одним з найбільш зручних інструментів, які встановлюються на Linux та інші \*NIX системи є Rsync. Як випливає з назви, Rsync використовується для синхронізації файлів на локальних і віддалених машинах. Rsync використовується для копіювання файлів на віддалені системи і навпаки, або для резервного копіювання на локально встановлені жорсткі

диски. На перший погляд користувача, Rsync може здаватись важким та незрозумілим засобом, однак він є одним з дійсно необхідних інструментів.

### Синхронізація файлів за допомогою Rsync

Загалом Rsync має одне застереження. Він є ефективним інструментом для переміщення даних з одного комп'ютера на інший, однак при неправильному

використанні, він є інструментом перезапису або видалення файлів. Відновлення файлів у Linux — дуже громісткий процес.

При запуску інструменту вперше, користувач має можливість скористатись опцією `-n` або, якщо перевага віддається довгому запису параметра, `-dry-run`. Виконання цієї команди призводить до пробного запуску без будь-яких реальних змін.

Для особистого використання, користувачі часто вибирають `Ifolder` або `Dropbox` для простих резервних копій. Ці інструменти відповідають вимогам клієнтів, серед яких:

- резервне копіювання;
- віддалений доступ;
- інші.

Однак ці сервіси відрізняються від `Rsync`. `Rsync` здебільшого використовується, щоб виконати резервне копіювання або складні операції синхронізації. Він містить великий стек функціоналу, який GUI інструменти не підтримують або підтримують, але роблять це погано. Прикладом такої функції є синхронізація файлу з домашньої директорії на локально примонтований USB накопичувач. Хоча й у GUI програм є також переваги: можливість загального доступу до файлів іншим користувачам чи групам, просте видалення файлів у папці (синхронізовано з віддаленою машиною).

Принциповим у `Rsync` є те, що файли копіюються не повністю, а лише на частину, яка відрізняється. Припустимо, у нас є два комп'ютери загального призначення `F` і `S`. Комп'ютер `F` має доступ до файлу `A` і `S` має доступ до файлів `B`, де `A` і `B` «схожі». Існує повільний канал зв'язку між `F` і `S`.

Алгоритм `Rsync` складається з наступних етапів:

1. `S` розбиває файл `B` в серію фіксованого розміру блоків, що не перекриваються розміром `L` байтів. Останній блок може бути коротше, ніж `L` байт.

2. Для кожного з цих блоків `L` обчислює дві контрольних суми: слабку 32-бітну контрольну суму і сильну 128-бітну за алгоритмом MD4.

3. `S` посилає ці контрольні суми до `F`.

4. `F` виконує пошук, щоб знайти всі блоки довжиною `L` байт (з будь-якого зсуву, а не тільки кратні `L`), які мають одну і ту ж слабку і сильну контрольні суми як одного з блоків `B`. Це може бути зроблено за один прохід дуже швидко, використовуючи нижче особливу властивість прокату контрольної суми.

5. `F` посилає `S` послідовність команд для побудови копії `A`. Кожна команда або являється посиланням на блок `B`, або буквенними даними. Буквальні дані передаються тільки для тих розділів, які не відповідають жодному з блоків `B`.

Кінцевим результатом є те, що `S` отримує копію, а тільки шматки, які не знайдені в `B` (плюс невелика

кількість даних для контрольних сум і блокових індексів), передаються по посиланню. Алгоритм зводить до мінімуму вплив затримки лінії зв'язку.

Найбільш важливі деталі алгоритму є прокат контрольної суми і пов'язані з ними кілька альтернативних механізмів пошуку, які дозволяють реалізувати дуже швидкий пошук контрольних сум.

Більшість дистрибутивів Linux має `Rsync`, встановлений за замовчуванням. Якщо немає, то варто використати менеджер пакунків дистрибутиву для установки `Rsync`. Наприклад, на OpenSUSE потрібно ввести:

```
sudo zypper in rsync
```

Наприклад, щоб скопіювати файли з домашньої директорії на диск USB:

```
rsync -avh --exclude="*.iso" /home/user/bin/ /media/diskid/user_backup/
```

`Rsync` дозволяє комбінувати опції після дефісу. Це один з пунктів, які роблять його зручним у використанні. В даному випадку після команди `rsync` чотири опції і один аргумент. Ключ `(-a)` говорить `Rsync` копіювати файли рекурсивно і зберегти групу і власності користувача, коли він копіює файли. Опція `(-v)` описує в терміналі дії `Rsync` більш детально. Це зручно використовувати для тестування команди, перш ніж додавати її в сценарій. При використанні цієї опції `Rsync` буде друкувати список змінених файлів, час відправки, обсяг даних і т.д. Однак кількість інформації буде обмеженою і вона буде неповною. І при потребі повністю вимкнути інформацію, варто передати ключ `-q`. Під час копіювання `Rsync` виводить інформацію, яка не є читабельною для користувача. Для того, щоб зробити її доступною, варто передати також ключ `-h`.

Ще однією командою, вказаною в прикладі є `--exclude`. Вона зазначає, які файли повинні бути проігноровані, в даному випадку це файли з ізо розширенням.

### **Створення резервних копій**

Як було описано вище, однією з ключових властивостей інструменту є можливість створення резервних копій. Зазвичай вони створюються на віддалених комп'ютерах, тому необхідно розглянути саме цей випадок.

`Rsync` може бути налаштований декількома різними способами для створення зовнішніх резервних копій, але варто розглянути найбільш практичний (також найпростіший і безпечний) метод тунельного `Rsync` через SSH. Більшість серверів і навіть багато клієнтів вже мають SSH, і він може бути використаний для резервного копіювання `Rsync`. Необхідно показати процес, щоб отримати одну машину Linux для резервного копіювання на іншу по локальній мережі.

Процес буде точно такий же, якщо робити це в Інтернеті. Варто лиш звернути увагу, що порт 22 (або будь-який інший порт, на який налаштований SSH), повинні бути спрямовані на будь-якому мережевому обладнанні на стороні сервера.

Для цього використовується та ж команда, що і для використання Rsync на локальному комп'ютері, але вона включатиме необхідні доповнення для тунелювання Rsync через SSH до сервера в своїй локальній мережі. Для користувача «user» підключення до «192.168.235.137» і, використовуючи такі перемикачі (-av -delete) ми виконаємо наступне:

```
$ rsync -av -delete -e ssh /Directory1/
user@192.168.235.137:/Directory2/
```

Якщо SSH прослуховування на якомусь іншому порту, а не 22, то необхідно буде вказати номер порту, наприклад:

```
$ rsync -av -delete -e 'ssh -p 12345' /
Directory1/ user@192.168.235.137:/Direc-
tory2/
```

Виконання резервного копіювання по мережі в значній мірі є таким же, як при створенні резервної копії на локальному рівні. Єдине, що змінюється, так це команда, яка використовується. Необхідно також зауважити, що присутній запит на введення пароля. Це для аутентифікації через SSH. Однак можна налаштувати ключі RSA, щоб пропустити цей процес і це в значній мірі спростить автоматизацію Rsync.

В Linux може бути використаний інструмент Cron, щоб автоматизувати виконання команд, таких як Rsync. Використовуючи його, є можливість запускати резервне копіювання щоночі, або так часто, як це потрібно. Для редагування файлу конфігурації Cron потрібно ввести:

```
crontab -e
```

Cron використовує такий синтаксис: хвилини, годину дня, день місяця, місяць року, день тижня, команду. Наступна команда буде запускати команду Rsync щоночі 22:00 вечора:

```
0 22 * * * rsync -av -delete /Directory1/
/Directory2/
```

Перший «0» вказує хвилини години, а «22» вказує на 10 годин вечора. Так як необхідно, щоб ця команда виконується щодня, потрібно залишити інші поля із зірочками, а потім вставити команду Rsync.

Є кілька різних способів, за допомогою яких можна шифрувати резервні копії Rsync. Найпростіший спосіб встановити шифрування на самому жорсткому диску (той, де будуть зберігатись копії). Інший спосіб для шифрування файлів перед їх відправкою на віддалений сервер. В обох випадках Rsync виявляється одним з найбільш ефективних і універсальних інструментів резервного копіювання на сьогоднішній день, і навіть простий Rsync сценарій може врятувати користувача від втрати даних.

### Література

1. Get to Know rsync [Електронний ресурс] — Режим доступу до ресурсу: <https://www.linux.com/learn/get-know-rsync>. — Дата доступу: 05.06.2016.
2. How to Use rsync to Backup Your Data on Linux [Електронний ресурс] — Режим доступу до ресурсу: <http://www.howtogeek.com/135533/how-to-use-rsync-to-backup-your-data-on-linux/>. — Дата доступу: 28.05.2016.

**Ханенко Олександр Андрійович**

*студент*

*Національний технічний університет України «Київський політехнічний інститут»*

**Ханенко Александр Андреевич**

*студент*

*Национальный технический университет Украины «Киевский политехнический институт»*

**Khanenko O.**

*student*

*National Technical University of Ukraine "Kyiv Polytechnic Institute"*

**ОГЛЯД МОЖЛИВОСТЕЙ PYTHON ЯК ЗАСОБУ  
ДЛЯ СТВОРЕННЯ ВЕБ-ПАРСЕРА  
ОБЗОР ВОЗМОЖНОСТЕЙ PYTHON КАК СРЕДСТВА  
ДЛЯ СОЗДАНИЯ ВЕБ-ПАРСЕРА  
REVIEW CAPABILITIES PYTHON AS A MEANS  
TO CREATE WEB PARSER**

**Анотація.** Мета даної роботи – розглянути можливості мови програмування Python для створення веб-парсера. Дана робота зосереджується на виборі оптимального модуля для реалізації програмного продукту.

**Ключові слова:** парсинг, Python, модуль, URL, HTML, LXML, запит, синтаксичний аналіз, веб-сайт.

**Аннотация.** Цель данной работы – рассмотреть возможности языка программирования Python для создания веб-парсера. Данная работа сосредотачивается на выборе оптимального модуля для реализации программного продукта.

**Ключевые слова:** парсинг, Python, модуль, URL, HTML, LXML, запрос, синтаксический анализ, веб-сайт.

**Summary.** The aim of this work is to consider the capabilities Python programming language for creating web parser. This work focuses on the selection of optimal module for software implementation.

**Key words:** parsing, Python, module, URL, HTML, LXML, request, syntactic analysis, website.

**Вступ**

Python — високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читання коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Основні архітектурні риси — динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень і зручні високорівневі структури даних. Код в Python організовується у функції та класи, які можуть об'єднуватися в модулі (вони в свою чергу можуть бути об'єднані в пакети).

Як наслідок, на цій мові створено немало зарекомендованих бібліотек, серед них і ті, які прекрасно справляються з парсингом веб-сторінок.

**Етапи парсинга**

Парсинг html-сторінки вдає із себе процес, який можна розбити на три етапи:

1. Отримання початкового коду веб-сторінки — скачати програмний код тієї сторінки сайту, з якої необхідно витягти інформацію.
2. Синтаксичний аналіз html-коду.
3. Фіксація результату (експорт даних).

**Завантаження веб-сторінок**

Для завантаження веб-сторінок в Python доступні бібліотеки urllib, urllib2, які виконують URL запити, але пропонують різні функціональні можливості:

- urllib2 може приймати об'єкт Request, щоб встановити заголовки для запиту URL, urllib приймає тільки URL.



- urllib надає метод urlencode, який використовується для генерації рядків запиту GET, urllib2 не має такої функції. Це одна з причин того, чому urllib часто використовується разом з urllib2.

При завантаженні веб-сторінок може виникнути багато проблем, що призупиняють виконання програми або повернуть некоректні дані. Основні з них наведені нижче:

- Сервер не відповідає.
- Веб-сайт блокує агента користувача робота.
- Доступ до сайту дозволений лише в окремих країнах.
- Веб-сторінки з динамічним контентом.
- Взаємодія з формами.
- Обхід CAPTCHA.

Всі ці проблеми можливо обійти, використовуючи готові модулі в Python та написавши декілька функцій.

У разі помилки 5xx коли сервер не відповідає, потрібно реалізувати функцію, яка буде повторно звертатися до сервера через певний проміжок часу. Потрібно використовувати ідентифіковану назву агента користувача, для того, щоб веб-сайт не розпізнав програми-робота, передавши у функцію urllib2.Request() установку headers = {'User-agent': 'wswp'}.

Деякі веб-сайти блокують користувачів, які знаходяться за межами даної країни, в такому разі отримати доступ можна через проксі-сервер. Реалізувати підтримку проксі можливо з urllib2.

Існують сторінки з динамічним контентом, коли JavaScript динамічно генерує контент веб-сторінки, це може варіюватися від простих форм до SPA (Single Page Application). Для вирішення цієї проблеми можна використати Selenium WebDriver, — це програмна бібліотека для управління браузерами. За допомогою Selenium можна з легкістю сфільтрувати дані чи виконати пошук по заданих критеріях, загрузити сторінку і далі аналізувати лише корисні нам дані.

З формами, що відправляють дані до сервера шляхом GET і POST запитів, легко справляється такий модуль як Mechanize. Поля форм є легко доступними, непотрібно управляти куками, Mechanize виконає це самостійно.

Якщо CAPTCHA (рис. 1) прості їх можна легко обійти за допомогою таких модулів як Pillow і Tesseract. В Pillow реалізований клас Image, в якому доступні методи для обробки зображень. Отже перше, що потрібно це забрати фон із зображення. А потім, підключивши модуль Tesseract, за допомогою pytesseract.image\_to\_string(img) перетворити зображення в строку.

У випадку складних CAPTCHA можемо скористатись спеціальними онлайн сервісами, що розпізнають

CAPTCHA зображення, де працюють реальні люди. Зареєструвавшись, отримуємо особистий ключ до їхньої API, пишемо програму, яка буде відправляти дані на сервер та отримувати їх.



Рисунок 1

### Аналіз та обробка даних

Три різні підходи в Python для реалізації синтаксичного аналізу:

- 1) регулярні вирази;
- 2) BeautifulSoup модуль;
- 3) lxml модуль.

Регулярні вирази доволі швидко працюють, але їх важко побудувати так як вони нечитабельні. Крім того, якщо розробники допишуть атрибути до тегів, то регулярні вирази прийдеться переписувати.

BeautifulSoup модуль доволі простий в реалізації і зрозумілий, в змозі правильно інтерпретувати відсутні лапки атрибутів і закриваючі теги, а також додати <html> і <body> теги, щоб сформувати повний HTML-документ. Мінус в тому, що він дуже повільно обробляє інформацію.

LXML є Python обгорткою поверх libxml2 XML-бібліотеки, написаної на C, яка допомагає зробити це швидше, ніж BeautifulSoup. Переваги lxml:

- Швидка і гнучка бібліотека для обробки XML/HTML на Python.
- Надає доступ до декларативного синтаксису XPath (рис. 2), зберігаючи при цьому загальну функціональність, доступну в Python
- Простота в написанні коду.
- Висока продуктивність при обробці дуже великих обсягів XML/HTML-даних.

Lxml.etree включає в себе клас ElementTree, що надає метод xpath() для підтримки синтаксису XPath, а також розширені функції.

### Висновок

Для коректної роботи програми-парсера рекомендовано написати універсальні методи для завантаження та обробки HTML-документа. Особливо в функції



Рисунок 2

завантаження врахувати всі можливі перешкоди, що можуть виникнути при спробі отримати веб-сторінку, більш детально ознайомитись з існуючими наведеними вище модулями. Для обробки веб-сторінки найкраще використовувати lxml в Python так як вона

одна з найшвидших бібліотек, яка справляється з аналізом даних. В ній реалізовано безліч різноманітних методів, а також вона дозволяє використовувати специфікацію Document Object Model XPath, синтаксис якої досить простий і доступний.

### Література

1. Grune D. Parsing Techniques – A Practical Guide / D. Grune, C. Jacobs. – Chichester: Originally published by Ellis Horwood, 1990. – 320 с.
2. Aho A. V. The theory of parsing, translation, and compiling / A. V. Aho, J. D. Ullman. – USA: Prentice-Hall, 1972. – 121 с.
3. Mitchell R. Web Scraping with Python / Ryan Mitchell. – Boston: O'Reilly Media, 2015. – 256 с.
4. Lxml – XML and HTML with Python [Електронний ресурс] – Режим доступу до ресурсу: <http://lxml.de/>. – Дата доступу: 28.05.2016.
5. BeautifulSoup 4 Python [Електронний ресурс] – Режим доступу до ресурсу: <http://www.pythonforbeginners.com/beautifulsoup/beautifulsoup-4-python>. – Дата доступу: 01.06.2016.

**Шеренковський Артем Олегович**

*Студент*

*Національний технічний університет України «Київський політехнічний інститут»*

**Шеренковский Артем Олегович**

*Национальный технический университет Украины «Киевский политехнический институт»*

**Sherenkovskiy A.**

*Student*

*National Technical University of Ukraine “Kyiv Polytechnic Institute”*

## АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СТВОРЕННЯ ІНТЕРНЕТ-РАДІОМОВЛЕННЯ

## АНАЛИЗ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СОЗДАНИЯ ИНТЕРНЕТ-РАДИОВЕЩАНИЯ

## ANALYSIS OF SOFTWARE FOR INTERNET BROADCASTING

**Анотація.** Проаналізоване програмне забезпечення для компонентів Інтернет-радіо.

**Ключові слова:** Інтернет-радіо, медіа-сервер, аудіо, потокова передача даних, плеєр.

**Аннотация.** Проанализировано программное обеспечение для компонентов Интернет-радио.

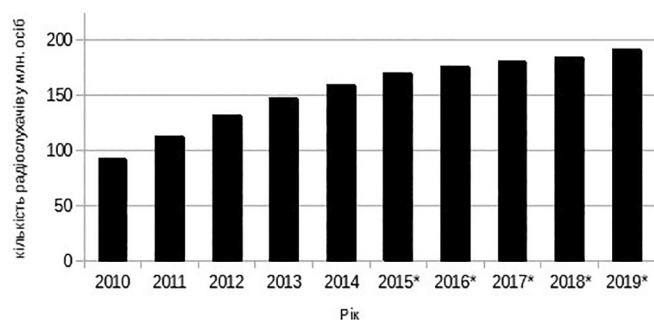
**Ключевые слова:** Интернет-радио, медиа-сервер, аудио, потоковая передача данных, плеер.

**Summary.** Software for Internet-radio components were analyzed.

**Key words:** Internet-radio, media-server, audio, streaming content transmission, player.

Незважаючи на велику кількість медіа-контенту в мережі Інтернет, радіомовлення через всесвітню павутину у сьогоденні не лише не втрачає своїх позицій, а й навпаки знаходить все більше й більше прихильників.

За даними статистики, кількість Інтернет-радіослухачів у США станом на 2014 рік була близькою до 160 мільйонів осіб, що складає близько 50% всього населення Сполучених Штатів, та близько 67% американців, що користуються Інтернетом. Разом із тим зростає час, який користувачі проводять слухаючи онлайн-радіо [1, с. 4].



Малюнок 1. Графік збільшення кількості Інтернет-радіослухачів (\* — прогнозоване збільшення) [2]

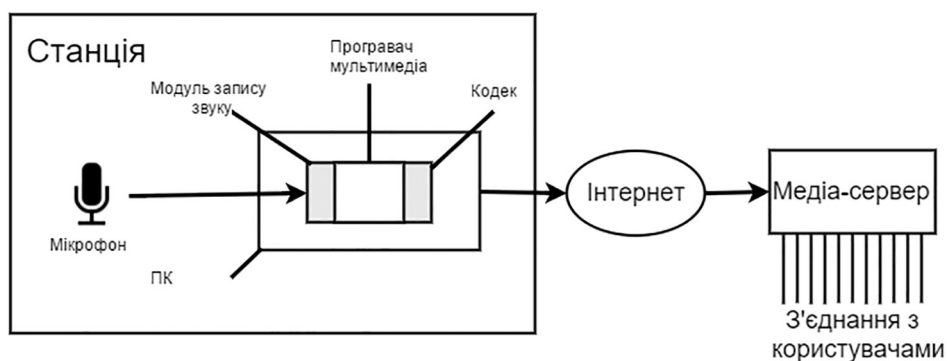
### Структура

У Інтернет-радіомовленні є 3 основних компоненти:

- Станція (радіостанція) — генерує потік аудіо-даних, та направляє його на сервер. Потребує малої пропускної здатності мережі, бо створює лише 1 потік(на сервер).
- Сервер (медіа-сервер) — приймає потік аудіо-даних від станції, та направляє його всім підключеним клієнтам. Потребує найбільшої пропускної здатності мережі, бо генерує окремий потік до кожного клієнту.
- Клієнт — приймає потік аудіо-даних від сервера та перетворює його на аудіо-сигнал, який сприймається кінцевим користувачем. Потребує малої пропускної здатності як і станція, бо лише приймає один потік від серверу.

### Програмне забезпечення компонентів Інтернет-радіо

**Радіостанція.** Першим, що необхідно будь-якій радіостанції у мережі Інтернет це програмне забезпечення, що зможе створювати та віддавати свій аудіо-потік у мережу. Найбільш розповсюдженими рішеннями є:



Малюнок 2. Структура з'єднання компонентів Інтернет-радіо [3, с. 767]

- Для комп'ютерів під управлінням ОС Windows:
  - RadioBOSS;
  - Winamp + плагін до SHOUTcast(програмне забезпечення медіа-серверу).
- Для комп'ютерів під управлінням ОС Linux:
  - Ices;
  - PulseAudio.

У таблиці 1 надано порівняльну характеристику даного програмного забезпечення.

**Медіа-сервер.** Найважливішою частиною Інтернет-радіомовлення є медіа-сервер – програмне забезпечення, що до якого під'єднуються усі потоки аудіо-даних, як вхідні, так і вихідні.

Існують два найпопулярніших медіа-сервера для радіомовлення: SHOUTcast та IceCast.

- **SHOUTcast** – це багатоплатформне програмне забезпечення для потокової передачі даних через мережу Інтернет, що поширюється у вільному доступі, проте з закритим кодом. Плюсами даного продукту є його легкість у налаштуванні та простота використання. Для цього ПЗ наявна велика кількість додаткових скриптів та інших додатків. Дає змогу потокового мовлення аудіо або відео потоків у форматах MP3, AAC, AACplus, NSV. SHOUTcast сервери і клієнти доступні для FreeBSD, Linux, Mac OS X, Microsoft Windows і Solaris. Також існують

клієнтські версії для таких операційних систем, як: Android, iOS, Windows Mobile, Blackberry OS та ін. Можливо транслювати аудіо потоки, що йдуть від декількох радіостанцій. Наявна повноцінна документація на англійській мові. Основним мінусом даного продукту є велика завантаженість пам'яті, CPU та використання великої кількості трафіку. Дуже мало вбудованих функцій та функцій адміністративного центру, а сам сервер продовжує працювати навіть при відсутності користувачів.

- **IceCast** – безкоштовне програмне забезпечення від Xiph.Org Foundation. IceCast також є багатоплатформним, проте на відміну від SHOUTcast має відкритий, що дає змогу самостійно змінювати, додавати або видаляти певний функціонал. У наявності користувача є велика кількість вбудованих функцій, а також великий набір зручних функцій адміністративної панелі. Підтримує велику кількість аудіо та відео форматів, серед яких, як розповсюдженні MP3, AAC, так і менш відомі Ogg Vorbis, Opus, Theora та NSV. На відміну від SHOUTcast, менша завантаженість CPU(в деяких ситуаціях до 10 разів), оперативної пам'яті та мережевого трафіку, а мовлення відбувається лише при наявності слухачів. Як і у аналога є можливість створення декількох паралельних аудіо-потоків для радіомовлення різних станцій та є

Таблиця 1

Порівняльна характеристика програмного забезпечення для створення аудіо-потoku

Критерії	Програвачі			
	RadioBOSS	Winamp	Ices	PulseAudio
Операційна система	Windows	Windows, Mac OS, Android	Linux	Windows, Linux, Solaris
Можливість роботи з декількома джерелами	+	-	+	+
Безкоштовність	-	+	+	+
Додаткові можливості	Створення звітів, генератор плейстів, рекламний планувальник	Багато плагінів, зручний ширококутовий еквалайзер	Мала необхідність у ресурсах	Синхронізація потоків, придушення акустичного відлуння

зручна онлайн документація англійською мовою. Мінусами даного програмного забезпечення є більша складність в установці та налаштуванні, а встановлення можливе лише на UNIX-подібні операційні системи, або на операційну систему Windows. Наявно мало додатків та скриптів.

**Клієнт.** В якості програмного забезпечення клієнта може виступати будь-який програвач, що може приймати дані з мережі Інтернет. Надалі ж було розглянуто три програвачі, а саме:

- **Winamp** — безкоштовний плеєр, що має реалізацію для ОС Windows, Mac OS, та Android. Плеєр відрізняється підтримкою великої кількості форматів та наявністю великої кількості плагінів, що дозволяють налаштувати функціонал плеєру під необхідності кожного користувача. Еквалайзер плеєру має широкий спектр налаштувань для різних видів музики, що покращує якість відтворення музики.
- **AIMP** — плеєр з підтримкою всіх відомих форматів аудіо. Має пакети для встановлення на ОС Windows та Android. Має у наявності велику кількість модулів. Дозволяє обробляти звук у якості 32 біти. Дозволяє як прослуховувати радіостанції, так і створювати власний аудіо-потік для відправки на медіа сервер.
- **Amarok** — найкращий плеєр для ОС Linux, хоча має реалізацію для ОС Windows та Mac OS. Відмінністю даного програвача є тісна інтеграція з Вікіпедією, що дозволяє швидко знаходити детальну інформацію про виконавця та треки, що програвються.

### Висновки

Виходячи із результатів порівняння можна сказати, що при виборі ПЗ для створення аудіо-потіку для комерційного користування, що потребує звітності і планування реклами кращим вибором буде RadioBOSS, що має цей функціонал. Для любительської радіостанції краще підійде PulseAudio, що є безкоштовним та багатоплатформовим і має можливості придушення відлуння та синхронізації вихідних аудіо-потоків.

Серед розглянутих варіантів реалізації медіа-серверу усі підтримують розповсюджені формати аудіо, мають можливість роботи з декількома джерелами звуку та мають повноцінну онлайн документацію англійською мовою. Проте мала необхідність у ресурсах надає перевагу саме медіа-серверу IceCast.

Щодо програм-програвачів, то всі розглянуті можуть відтворювати аудіопотік, що йде з мережі, та мають еквалайзери для покращення звучання композиції, тож питання обрання певного з них переходить до питання зручності використання для кожного індивідуального користувача. Окремо можна відмітити Амагок, єдиний з розглянутих варіантів, що має дистрибутиви для ОС Linux, та має тісну інтеграцію з Вікіпедією, що надає можливість швидкого пошуку інформації про композицію, альбом чи автора.

### Література

1. Internet Radio Trends Report 2015, [Електронний ресурс]. — Режим доступу: [http://xappmedia.com/wp-content/uploads/2015/01/Internet-Radio-Trends-Report-2015\\_january.pdf](http://xappmedia.com/wp-content/uploads/2015/01/Internet-Radio-Trends-Report-2015_january.pdf)
2. Number of monthly digital radio listeners in the United States from 2010 to 2019, [Електронний ресурс]. — Режим доступу: <http://www.statista.com/statistics/191657/us-weekly-internet-radio-listeners-since-2009/>
3. Таненбаум Э. Компьютерные сети 5-е издание / Эндрю Таненбаум. — Санкт-Петербург: «Питер», 2012.

**Шилова Тетяна Олександрівна**

*кандидат технічних наук, доцент, доцент кафедри міського будівництва  
Київського національного університету будівництва і архітектури*

**Шилова Татьяна Александровна**

*кандидат технических наук, доцент, доцент кафедры городского строительства  
Киевского национального университета строительства и архитектуры*

**Shilova T. O.**

*Candidate of Technical Sciences (Ph. D.), Docent, Associate Professor at the Department of City  
Development*

*Kyiv National University of Construction and Architecture*

## ПРИНЦИПИ ПРОЕКТУВАННЯ МІСЬКИХ І ЗАМІСЬКИХ ДОРІГ

## ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ ГОРОДСКИХ И ВНЕГОРОДСКИХ ДОРОГ

## PRINCIPLES OF DESIGN OF URBAN AND EXTRA-URBAN ROADS

**Анотація.** Виконаний аналіз та зіставлення нормативів і методів проектування міських і заміських доріг з метою удосконалення засобів та підходів до проектування обох типів доріг.

**Ключові слова:** автомобільна дорога, міські вулиці та дороги, поздовжній профіль, поперечний профіль, дорожній одяг, водовідведення, зовнішнє освітлення, ландшафтне проектування, інженерний благоустрій.

**Аннотация.** Выполнен анализ и сопоставление нормативов и методов проектирования городских и внегородских дорог с целью усовершенствования средств и подходов к проектированию обоих типов дорог.

**Ключевые слова:** автомобильная дорога, городские улицы и дороги, продольный профиль, поперечный профиль, дорожная одежда, водоотвод, внешнее освещение, ландшафтное проектирование, инженерное благоустройство.

**Summary.** Analysis and comparison of the norms and methods for a design of urban and extra-urban roads was conducted to improve tools and approaches used for the design of both types of roads.

**Key words:** highway, urban streets and roads, longitudinal profile, cross-section, road pavement, drainage, outdoor lighting, landscape design, land improvement.

Автомобільні дороги та вулиці і дороги населених пунктів мають утворювати єдину цілісну мережу транспортних комунікацій країни. Для цього на периферії міст — там, де відбувається стикування цих різних шляхів сполучення, автомобільні дороги та міські вулиці повинні мати однакові або близькі параметри. Але навіть на погляд пересічного мешканця, не кажучи вже про фахівця-містобудівника, ці види комунікацій мають багато відмінностей. Тому здалося цікавим зіставити принципи проектування міських і заміських доріг.

**Автомобільна дорога** — лінійний комплекс інженерних та допоміжних споруд, призначених для безперервного, безпечного та зручного проїзду транспортних засобів з нормативними технічними характеристиками та навантаженнями, що забезпечує внутрішньодержавні та міжнародні перевезення пасажирів і вантажів.

**Міська вулиця** — обмежений принаймні з одного боку рядом будинків простір в межах міста або іншого населеного пункту, призначений для проїзду транспорту та ходіння. Зазвичай вулиця обмежена рядами будинків з обох сторін проїзної частини.

Як критерії порівняння обрані такі:

- Вихідні дані для проектування — інтенсивність руху.
- Принципи проектування поздовжнього профілю.
- Принципи проектування поперечного профілю.
- Водовідведення.
- Ландшафтне проектування.
- Дорожній одяг.
- Зовнішнє освітлення.
- Інженерний благоустрій.

Спочатку ми відмітили різницю у функціональному призначенні цих різних шляхів сполучення.

Так, автомобільна дорога призначена для пропуску лише транспортних потоків за межами міста. Наявність пішоходів на ній не припустима. Громадський транспорт — лише автомобільний (автобуси), інші види — швидкісні — на уособленому полотні за межами земляного полотна дороги. На міських вулицях, за винятком магістралей безперервного руху, зазвичай передбачається наявність різних видів громадського транспорту та пішоходів.

Автомобільні дороги загального користування згідно з Законом України «Про автомобільні дороги» поділяються на дороги державного та місцевого значення. Автомобільні дороги державного значення, в свою чергу, поділяються на міжнародні, національні та регіональні. Автомобільні дороги місцевого значення поділяються на територіальні, обласні та районні [4].

Відповідно до цього поділу здійснюється технічна класифікація автомобільних доріг за категоріями залежно від розрахункової середньорічної добової перспективної інтенсивності руху.

Вулично-дорожня мережа міст, як відомо, за функціональним призначенням поділяється на:

- магістральні дороги безперервного та регульованого руху;
- магістральні вулиці загальноміського значення теж безперервного або регульованого руху;
- магістральні вулиці районного значення;
- вулиці та дороги місцевого значення, в тому числі: житлові вулиці, дороги промислових і комунально-складських зон, проїзди [1, 2, 3].

Селищні та сільські вулиці та дороги поділяються на такі категорії:

- селищні дороги;
- головні вулиці;
- житлові вулиці;
- дороги виробничого призначення;
- проїзди [1, 2].

Якщо порівняти розрахункову інтенсивність руху транспортних потоків, яка визначається категорією шляхів сполучення, перерахувавши в однакові одиниці, то можна побачити, що транспортне навантаження на міські вулиці в 2–3 рази більше, ніж на автомобільну дорогу. А розрахункові швидкості одиночного автомобіля на транспортних комунікаціях близьких категорій — майже однакові. При цьому геометричні параметри відповідних шляхів сполучення встановлюються залежно від розрахункової швидкості. І тому, зрозуміло, що міські вулиці працюють в більш напружених умовах, ніж автомобільні дороги. До того ж на міських вулицях додаються ще проблеми з організацією пішохідного руху, пропусканням громадського транспорту, прокладкою численних комунікацій. Все це, звісно,

наклало відбиток на підходи до проектування. Проектувальникам міських вулиць доводиться вирішувати більше проблем не лише суто технічного характеру, але й архітектурного, пов'язаних з виконанням вулицями репрезентативної функції, бо вулиці є осями формування забудови, скелетом міста. Крім того, необхідно зважати на те, що вулиці є джерелом екологічного дискомфорту, місцем концентрації негативних чинників, що супроводжують бурхливий розвиток автомобілізації — забруднення повітряного басейну, підвищеного рівня шуму, вібрації, електромагнітного випромінювання тощо. Тому містобудівникам доводиться ще намагатися захищати мешканців міста від цих негативних чинників, забезпечити газообмін між містом та приміською зоною коридорами, що утворюються вулицями, проектуючи їхнє відповідне озеленення та облаштування. До того ж проектувальникам міських вулиць доводиться, як правило, працювати в стиснених умовах вже сформованої усталеної міської забудови, де важко різко змінити геометричні параметри вулиць, довівши їх до сучасних нормативів та забезпечивши пропускання інтенсивних транспортних потоків з високими динамічними якостями. Часті перехрестя, необхідність забезпечити закриті водовідведення з міських вулиць ускладнюють вирішення питань вертикального розпланування та інженерного облаштування міських шляхів сполучення.

На заміських дорогах — свої проблеми. Інженеру-дорожнику важко запроектувати земляне полотно зі забезпеченням необхідного водно-теплового режиму та морозостійкості, надати можливість сучасним транспортним засобам реалізувати свої динамічні якості (це мають забезпечити геометричні параметри доріг), організувати проходження автомобільних доріг загальної мережі через населені пункти, створити зручні та безпечні вузли перетину доріг між собою.

Як бачимо, нам здалося цікавим і корисним проаналізувати підходи до проектування автомобільних доріг та міських вулиць, порівняти нормативи, вивчити досвід проектування. Це дасть можливість удосконалити нормативи та методики проектування міських та заміських доріг.

В цій роботі ми провели аналіз проектування автомобільних доріг та міських вулиць за кожним з обраних критеріїв.

Проекту лінії поздовжнього профілю осі магістралі міських вулиць зазвичай проектують з окремих ланок, які складаються з ділянки прямої лінії та відповідної ділянки вертикальної спряженої кривої (рис. 1), найчастіше, колової.

Задачу проектування вважають вирішеною, якщо між кінцями вертикальних кривих є прямі ділянки, особливо між протилежними за характером.

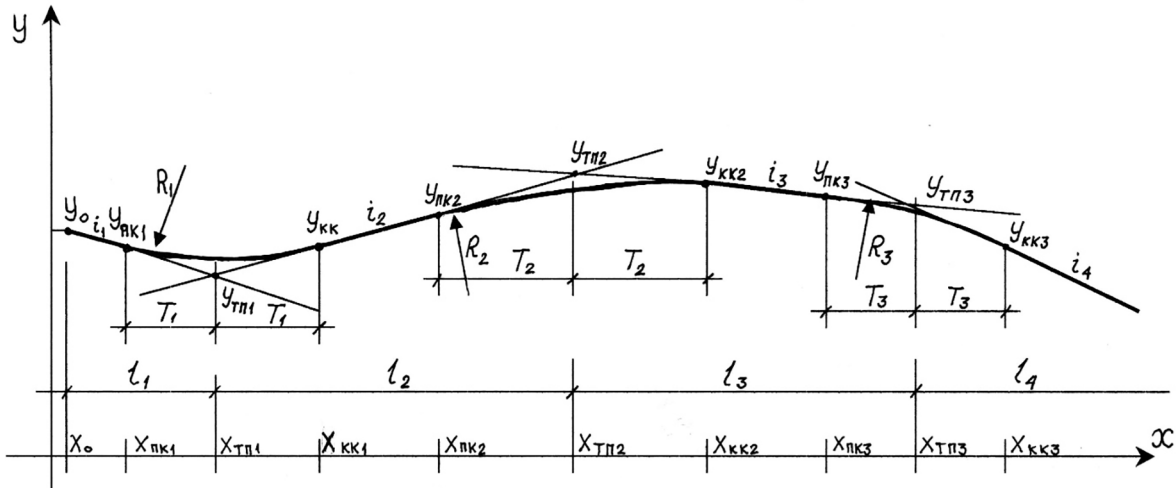


Рис. 1. Принцип проектування поздовжнього профілю міської вулиці

Трасу ж автомобільної дороги проектують як плавну лінію у просторі з ув'язкою елементів плану, поздовжнього та поперечного профілів між собою, з навколишнім ландшафтом і з оцінкою їх впливу на умови руху та зорове сприйняття дороги. При цьому план і поперечний профіль дороги проектують одночасно, виходячи з інтенсивності руху, умов забезпечення комфортності руху та безпеки транспортних засобів і враховуючи можливості реконструкції дороги за межею термінів перспективного розрахункового періоду (табл. 1).

Ділянки прямих і кривих у плані при радіусі кривої у плані 2000м та менше дорожники з'єднують перехідними кривими. Найменші довжини перехідних кривих у метрах наведені в табл. 2.

Проектувальники автомобільних доріг у своїй діяльності керуються чіткими правилами, головними з яких є:

- Криві у плані і поздовжньому профілі доцільно суміщати. При цьому криві у плані повинні бути на 100–150 м довші за криві в поздовжньому профілі.
- Слід уникати сполучення кінців кривих у плані з початком кривих у поздовжньому профілі. Відстань між ними рекомендується приймати не менше 150 м.
- Не рекомендується коротка пряма вставка між двома кривими в плані, які спрямовані в один бік. При довжині її менше 100м рекомендується замінювати обидві криві однією більшого радіуса.

Таблиця 1

**Параметри елементів плану і поздовжнього профілю, що залежать від розрахункової швидкості**

Найменування елементів	Параметри залежно від розрахункових швидкостей, км/год									
	150	140	120	110	100	90	80	60	50	30
Найбільший поздовжній уклон, %.	30	35	40	45	50	55	60	70	80	100
Найменший радіус кривої у плані, м	1200	1100	800	700	600	450	300	150	100	30
Найменший радіус кривої у поздовжньому профілі, м;										
– опуклої;	30000	25000	15000	12500	10000	7500	5000	2500	1500	600
– увігнутої	8000	7000	5000	4000	3000	2500	2000	1500	1200	600
Найменша відстань видимості, м:										
– для зупинки автомобіля;	300	300	250	225	200	175	150	85	75	45
– зустрічного автомобіля	-	-	450	400	350	300	250	170	130	90

Таблиця 2

**Найменші довжини перехідних кривих**

Радіус колової кривої	30	50	60	80	100	150	200	250	300	400	500	600–1000	1000–2000
Довжина перехідної кривої	30	35	40	45	50	60	70	80	90	100	110	120	100



- При довжині прямої вставки від 100 м до 400 м вставку доцільно замінювати перехідними кривими. Прямая вставка як самостійний елемент траси допускається для доріг I-а, I-б та II категорій при її довжині понад 700 м, для доріг III та IV категорій — понад 400 м.
- Максимальну довжину прямих вставок у поздовжньому профілі необхідно обмежувати залежно від алгебраїчної різниці поздовжніх уклонів та радіусів увігнутих кривих.
- У місцях зламу проектної лінії в поздовжньому профілі при алгебраїчній різниці уклонів 5‰ і більше на дорогах I-а, I-б і II категорій, 10‰ і більше на дорогах III категорії та 20‰ і більше на дорогах IV і V категорій слід застосовувати вертикальні криві.

На рис. 2 показаний приклад проектування поздовжнього профіля автомобільної дороги, а на рис. 3 — міської вулиці.

Значно відрізняються принципи проектування та типові поперечні профілі автомобільних доріг та міських вулиць.

Так, в перелік елементів поперечного профіля міської вулиці входять крім проїзної частини ще розділювальні та технічні смуги різного призначення, тротуари (ширина яких встановлюється згідно з категорією магістралі), трамвайне полотно (суміщене чи уособлене, розташоване в центрі або по боках проїзної частини), велодоріжки, кишені для влаштування зупинок громадського транспорту, місцеві проїзди, смуги озеленення, огорожі. Крім того, під різними елементами вулиць розміщується велика кількість інженерних мереж. При цьому інженерні мережі розміщуються таким чином: під тротуарами і розділювальними смугами — інженерні мережі в колекторах, каналах або тунелях; у межах розділювальних смуг — теплові мережі, водопровід, газопровід, господарсько-побутова й дощова каналізація (рис. 4).

Елементами поперечного профілю автомобільної дороги є: проїзна частина, узбіччя (в тому числі укріплена смуга узбіччя зі зупинковою смугою), розділювальна смуга (в тому числі укріплена смуга), земляне полотно (рис. 5).

Тобто, як бачимо, поперечні профілі автомобільної дороги відрізняються наявністю узбіччя і здебільшого відсутністю тротуарів та інженерних мереж.

Для міських вулиць влаштовується замкнена система водовідведення зі застосуванням дощоприймальних та оглядових колодязів та водонепроникних покриттів. Збір та відведення поверхневих вод забезпечується засобами вертикального планування вулиці та прилеглої території таким чином, що вода з прилеглої території збирається на вулиці, звідки відводиться в дощоприймальну каналізацію.

На автомобільній дорозі для захисту земляного полотна від перезволоження поверхневими водами, розмиву, а також для забезпечення виконання робіт на час спорудження земляного полотна передбачається система поверхневого водовідведення (планування території, влаштування каналів, лотків, перепадів, швидкотоків, випаровувальних басейнів, поглинальних колодязів тощо). Рівні ґрунтових і поверхневих вод, що впливають на міцність та стійкість земляного полотна або на умови виконання будівельних робіт, знижують, а воду відводять за межі земляного полотна.

Проектувальники автомобільних доріг більше уваги приділяють ландшафтному проектуванню, гарному вписуванню дороги в рельєф та прилеглий ландшафт, засобами озеленення вирішують специфічні задачі.

Насадження вздовж автомобільних доріг створюють для архітектурно-художнього оформлення (декоративне озеленення), захисту проїзної частини від снігових заносів (снігозахисне озеленення), а також для запобігання розмиванню доріг, зсувам, сильним вітрам та пиловим бурям (протиерозійне озеленення). Метою благоустрою та озеленення автомобільних доріг є створення комфортних умов для учасників дорожнього руху. Декоративне озеленення згідно з існуючими садово-парковими стилями та місцевими умовами створюється такими прийомами:

- регулярним — лінійні (алеїні або рядові) посадки дерев та чагарників, а також живоплоти;
- ландшафтним або вільним — групові посадки дерев та чагарників в поєднанні з прилеглим до дороги ландшафтом;
- мішаним — поєднання регулярних та вільних посадок, а також комплексні посадки біля перехресть, автобусних зупинок, шляхопроводів, при підході до лісових масивів тощо.

Декоративні зелені насадження розміщують за межами земляного полотна в залежності від категорії дороги, але не ближче, м: 14,0 для I категорії, 11,25 — II, 9,5 — III, 9,0 — IV, 8,25 — V від кромки проїзної частини.

Практика декоративного озеленення доріг передбачає також створення посадок зорового орієнтування, які можна поділити на три групи:

- напрямні посадки, які вказують напрям проїзду, — створюються лінійним прийомом. На крутих поворотах доріг повинно бути дві окремі ділянки лінійних посадок;
- бар'єрні посадки вказують на неможливість продовження руху в одному і тому ж напрямі. Вони можуть бути як лінійними, так і груповими;
- акцентуючі посадки орієнтують зір водія на найважливіші місця розв'язок, комплексів обслуговування тощо.

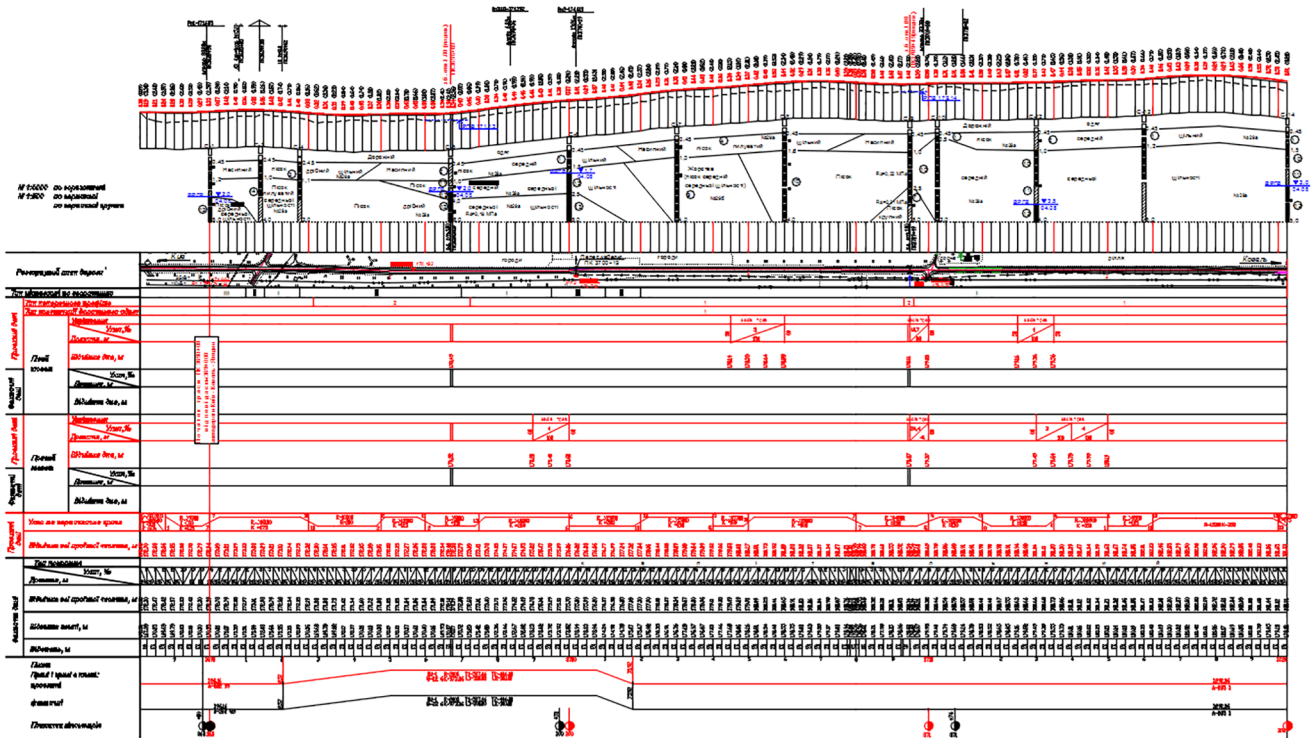


Рис. 2. Приклад проектування поздовжнього профілю автомобільної дороги

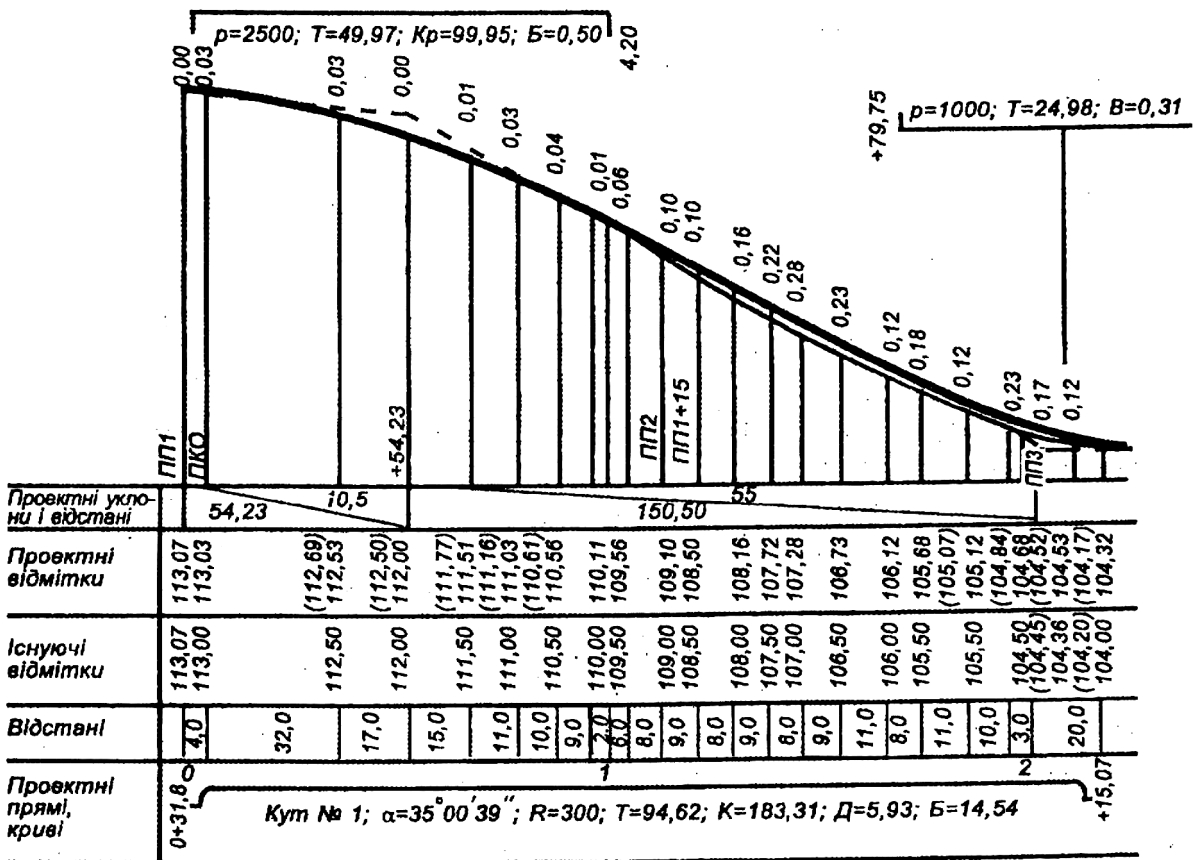


Рис. 3. Приклад проектування поздовжнього профілю міської вулиці

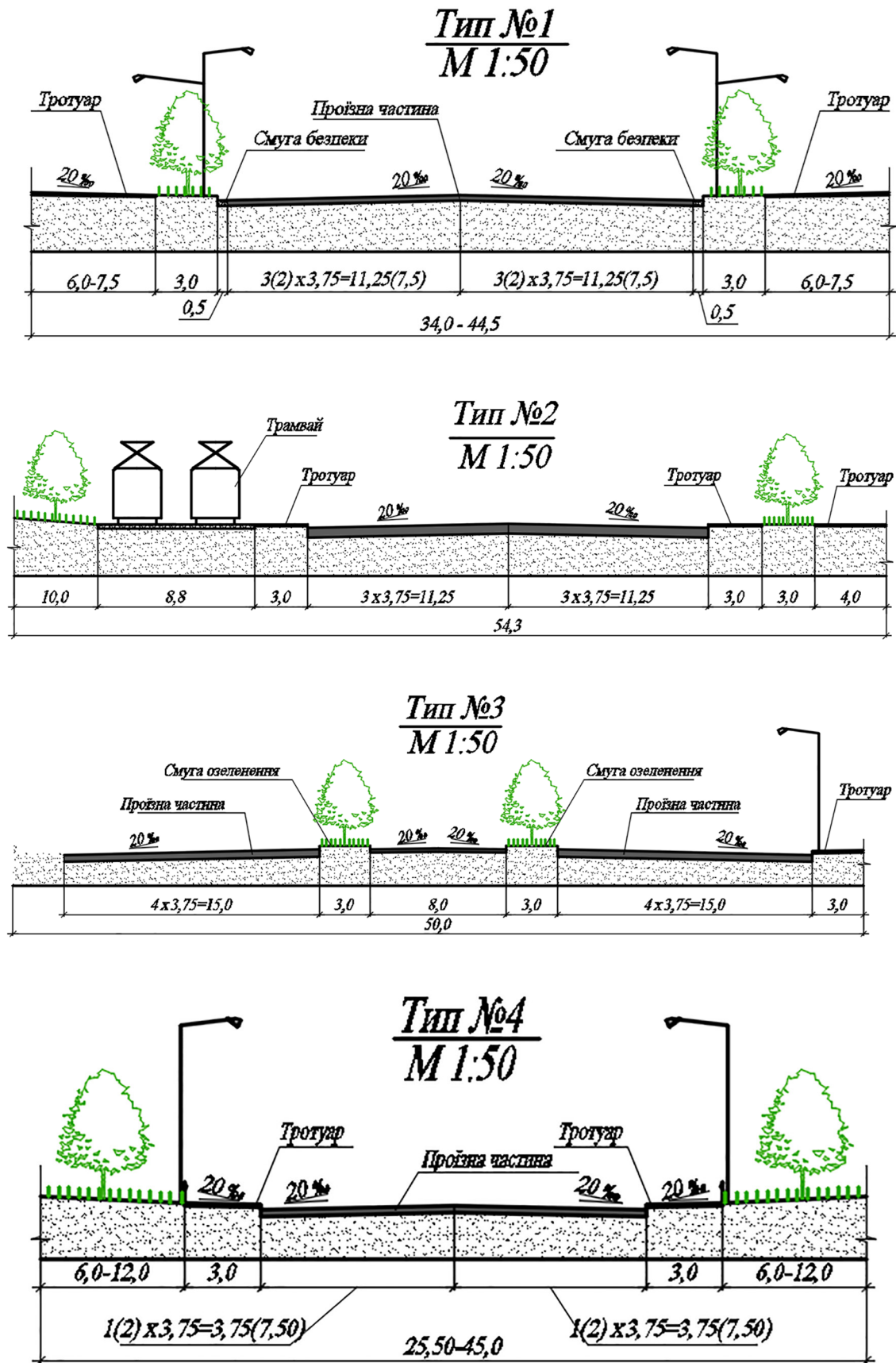


Рис. 4. Типові поперечні профілі міських вулиць

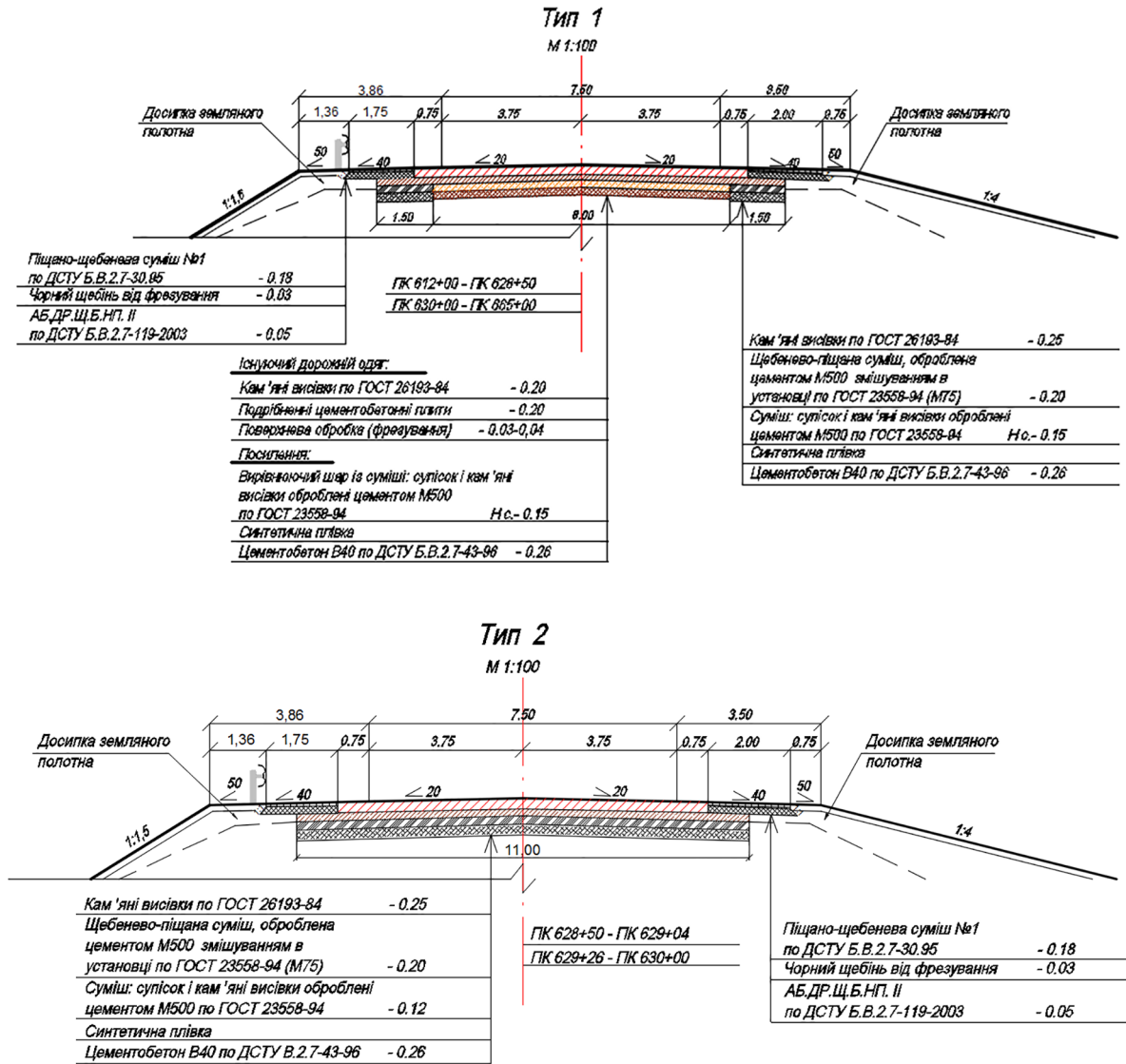


Рис. 5. Типові поперечні профілі автомобільних доріг

Конструкції дорожнього одягу різних типів доріг також істотно відрізняються. Для того, щоби можна було користуватися одним нормативним документом [5] для розрахунків конструкцій дорожнього одягу обох типів доріг передбачений умовний перехід від категорій автомобільних доріг загальної мережі до категорій міських доріг і вулиць (табл. 3). Звісно, що міські вулиці мають свої особливості, а більша інтенсивність руху та наявність громадського пасажирського транспорту різних типів (не лише автобусів, а й тролейбусів), ускладнюють розрахунки.

Зовнішнє освітлення здебільшого застосовується лише на міських вулицях, а на автомобільних дорогах — тільки в межах населених пунктів. Основна задача освітлення — це створення сприятливих умов для безпечного руху транспорту і пішоходів.

У місті розрізняють такі види постійних освітлювальних установок:

- для вуличного освітлення (забезпечення освітленості, необхідної для безпеки руху транспорту і пішоходів);
- для архітектурно-художнього освітлення (створення світлової архітектури міста у вечірні години з виявленням найбільш цінних в архітектурному, історичному та художньому відношенні будівель, споруд, пам'яників, фонтанів тощо, а також цілих комплексів);
- для рекламного освітлення (інформація населення про торговельні, побутові і культурні новини, оформлення вітрин магазинів, кіосків та ін.);
- для світлових сигналів (показники транспорту і пішоходам напрямів руху, місць зупинок, стоянок, переходів тощо).

Таблиця 3

**Умовний перехід від категорій автомобільних доріг загальної мережі до категорій міських вулиць і доріг**

№ п/п	Категорія вулиць і доріг	Аналог категорій доріг загальної мережі
1	Магістральні дороги, магістральні вулиці загальноміського значення, дороги вантажного руху	I, II
2	Магістральні вулиці районного значення	II
3	Вулиці та дороги місцевого значення, дороги промислових і складських районів	III
4	Житлові вулиці та проїзди, селищні вулиці та дороги	IV, V

Всі види установок повинні працювати у взаємодії одне з одним, враховуючи яскравість дорожніх покриттів вулиць, площ і тротуарів, яскравість вітрин, світлової реклами і світильників, а також освітлених пам'яників і фонтанів, ступінь блискості, що виникає в полі зору людини [6,7].

Проведений всебічний аналіз нормативів та методів проектування автомобільних доріг та міських вулиць дозволив нам зробити наступні висновки:

- на міських вулицях значно більша інтенсивність руху транспорту, пішоходів та громадського транспорту;
- проектування поздовжнього профілю міської вулиці ведеться прямими ділянками, що сполучаються коловими кривими, а автомобільної дороги — перехідними кривими великих радіусів з нечисленними прямими вставками;
- в поперечному профілі міської вулиці багато різноманітних елементів, зокрема, трамвайне полотно, велодоріжки, розділювальні та технічні смуги з великою кількістю комунікацій під ними, а поперечний профіль автомобільної дороги відрізняється наявністю земляного полотна та узбіччя;
- на міських вулицях прийнята замкнена система водовідведення, на автомобільних дорогах — відкрита;
- стиснені умови проектування міських вулиць;

– дорожній одяг міських вулиць більш складних конструкцій, має бути розрахований на велику кількість гальмувань та розгонів транспортних засобів і громадського пасажирського транспорту;

– міські вулиці відрізняються наявністю більшої кількості елементів інженерного благоустрою та оздоблення: пішохідні переходи, штучні споруди, озеленення, огорожі, рекламні носії;

– при вертикальному розплануванні міської вулиці складніше вирішуються місця перетинів магістралей зі складними поперечними профілями;

– на міських вулицях більш жорсткі вимоги до зовнішнього освітлення;

– на автомобільних дорогах значно більші вимоги до зорової плавності та ландшафтного проектування доріг.

Отже, бачимо, що аналіз методів проектування автомобільних доріг та міських вулиць дає змогу виявити кращі риси методів проектування тих чи інших доріг та запозичити прогресивний досвід. Скажімо, містобудівникам, хоча вони й працюють в складніших умовах, варто повчитися тому, як дорожники одночасно проектують поздовжній та поперечний профіль, ув'язуючи їх на попередній стадії проектування та гарно вписуючи магістраль в рельєф.

**Література**

1. ДБН 360-92\*. Містобудування. Планування і забудова міських і сільських поселень. — К.: Мінбудархітектури України. 1993. — 109 с.
2. ДБН 360-92\*\*. 2002. — 113 с.
3. ДБН В.2.3-5-2001. Споруди транспорту. Вулиці та дороги населених пунктів. — К.: Держбуд України, 2001. — 50 с.
4. ДБН В.2.3-4-2000. Споруди транспорту. Автомобільні дороги. — К.: Держбуд України, 2000. — 117 с.
5. ВБН В.2.3-218-186-2004. Споруди транспорту. Дорожній одяг нежорсткого типу. — К.: Укравтодор, 2004. — 151 с.
6. Інженерне обладнання та облаштування вулиць: навчальний посібник у 2-х ч. — Ч. 1 / М. М. Осетрін, Т. О. Шилова, П. П. Чередніченко. — К.: КНУБА, 2011. — 96 с.
7. Інженерне обладнання та облаштування вулиць: навчальний посібник у 2-х ч. — Ч. 2 / М. М. Осетрін, Т. О. Шилова, П. П. Чередніченко. — К.: КНУБА, 2012. — 96 с.

**Якимець Роман Вікторович**

*студент*

*Національний технічний університет України «Київський політехнічний інститут»*

**Яременко Костянтин Миколайович**

*студент*

*Національний технічний університет України «Київський політехнічний інститут»*

**Якимець Роман Викторович**

*студент*

*Национальный технический университет Украины «Киевский политехнический институт»*

**Яременко Константин Николаевич**

*студент*

*Национальный технический университет Украины «Киевский политехнический институт»*

**Yakymets Roman V.**

*Student*

*National Technical University of Ukraine «Kyiv Polytechnic Institute»*

**Yaremenko K. N.**

*Student*

*National Technical University of Ukraine «Kyiv Polytechnic Institute»*

## МАСШТАБУВАННЯ НАВАНТАЖЕННЯ WEB-ДОДАТКІВ МАСШТАБИРОВАНИЕ НАГРУЗКИ WEB-ПРИЛОЖЕНИЙ SCALING OF LOAD WEB-APPLICATIONS

**Анотація.** Досліджені принципи зменшення навантаження на web-додаток шляхом його оптимізації та масштабування. Розглянуті методи масштабування та способи їх застосування.

**Ключові слова:** зменшення навантаження, оптимізація, масштабування, web-додаток, хост.

**Аннотация.** Исследованы принципы уменьшения нагрузки на web-приложение путем его оптимизации и масштабирования. Рассмотрены методы масштабирования и способы их применения.

**Ключевые слова:** уменьшение нагрузки, оптимизация, масштабирование, web-приложение, хост.

**Summary.** Researched principles of reduce the load on web-application through optimization and scaling. There was considered the methods of scaling and methods of their application.

**Key words:** load reduction, optimization, scale, web-application, host.

### Вступ

Із зростанням популярності web-додатку його підтримка неминуче починає вимагати все більших і більших ресурсів. Перший час з навантаженням можна боротися шляхом оптимізації алгоритмів і архітектури самого додатка. Однак якщо все, що можна було оптимізувати, вже оптимізовано, а додаток все одно не справляється з навантаженням потрібно масштабувати web-додаток.

### Оптимізація

Насамперед необхідно перевірити наступні положення:

- Оптимальність запитів до БД.
- Правильність збереження даних в БД.
- Використання кешування.
- Відсутність зайвих запитів до ФС чи БД.
- Оптимальність алгоритмів обробки даних.

Про кожен з цих пунктів можна написати окрему статтю, так що їх детальний розгляд в рамках даної статті явно надлишковий. Важливо лише розуміти, що перед тим як приступити до масштабування додатку, вкрай бажано максимально оптимізувати його роботу — адже можливо тоді ніякого масштабування і не буде потрібно.

### Масштабування

Припустимо, що оптимізація вже проведена, але додаток все одно не справляється з навантаженням. В такому випадку рішенням проблеми, може бути його розподілення на декілька хостів, з метою збільшення загальної продуктивності додатку за рахунок збільшення доступних ресурсів. Такий підхід має офіційну назву — «масштабування» (scale) додатку. Точніше кажучи, під «масштабністю» (scalability) називається можливість системи збільшувати свою продуктивність при збільшенні кількості виділених для неї ресурсів. Розрізняють два способи масштабування: вертикальний і горизонтальний. Вертикальне масштабування має на увазі збільшення продуктивності додатка при додаванні ресурсів (процесора, пам'яті, диска) в рамках одного вузла (хоста). Горизонтальне масштабування характерне для розподілених додатків і має на увазі зростання продуктивності додатка при додаванні ще одного вузла (хоста).

Найпростішим способом буде просте оновлення заліза (процесора, пам'яті, диска) — тобто вертикальне масштабування. Крім того, цей підхід не вимагає ніяких доопрацювань додатка. Однак, вертикальне масштабування дуже швидко досягає своєї межі, після чого розробнику та адміністратору нічого не залишається окрім як перейти до горизонтального масштабування додатка.

### Архітектура програми

Більшість web-додатків є розподіленими, так як в їх архітектурі можна виділити мінімум три шари: web-сервер, бізнес-логіка, дані (БД, статика).

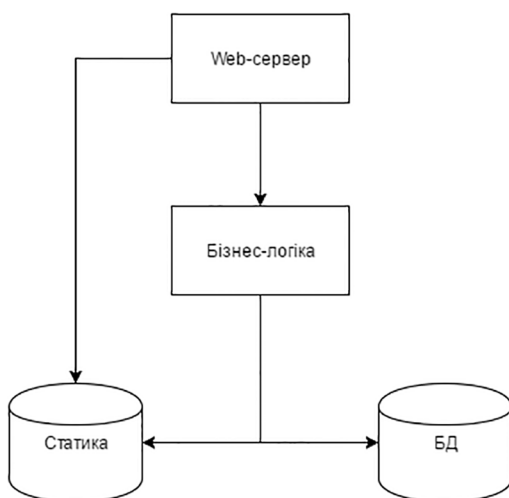


Рисунок 1. Архітектура програми

Кожен з цих шарів може бути масштабований. Тому якщо у системі бізнес-логіка і БД живуть на одному хості — першим кроком повинно стати розподілення їх на різні хости.

### Вузьке місце

Приступаючи до масштабування системи, перш за все варто визначити, який з шарів є «вузьким місцем» — тобто працює повільніше за решту системи. Для початку можна скористатися банальними утилітами типу top (htop) для оцінки споживання процесора/пам'яті і df, iostat для оцінки споживання диска. Однак, бажано виділити окремий хост, з емуляцією бойового навантаження (з допомогою AB або Jmeter), на якому можна буде профілювати роботу програми за допомогою таких утиліт як xdebug, orprofile і так далі. Для виявлення вузьких запитів до БД можна скористатися утилітами типу pgFouine.

Зазвичай все залежить від архітектури додатку, але найбільш ймовірними кандидатами на «вузьке місце» в загальному випадку є БД і код. Якщо додаток працює з великим об'ємом призначених для користувача даних, то «вузьким місцем», відповідно, швидше за все буде зберігання статички.

### Масштабування БД

Як вже говорилося вище, найчастіше вузьким місцем в сучасних додатках є БД. Проблеми з нею діляться, як правило, на два класи: продуктивність і необхідність зберігання великої кількості даних.

Знизити навантаження на БД можна розділивши її на кілька хостів. При цьому гостро встає проблема синхронізації між ними, вирішити яку можна шляхом реалізації схеми master/slave з синхронної або асинхронної реплікацією. У випадку з PostgreSQL реалізувати синхронну реплікацію можна за допомогою Slony-I, асинхронну — PGPool-II або WAL (9.0). Вирішити проблему поділу запитів читання і запису, а так само балансування навантаження між наявними slave'ами, можна за допомогою настройки спеціального шару доступу до БД (PGPool-II).

Проблему зберігання великого обсягу даних в разі використання реляційних СУБД можна вирішити за допомогою механізму партиціонування («поділ» в PostgreSQL), або розгортаючи БД на розподілених ФС типу Hadoop DFS.

Проте, для зберігання великих обсягів даних найкращим рішенням буде «шардінг» (Sharding) даних, який є вбудованим перевагою більшості NoSQL БД (наприклад, MongoDB).

Крім того, NoSQL БД в загальному працюють швидше своїх SQL-братів за рахунок відсутності

overhead'a на розбір / оптимізацію запиту, перевірки цілісності структури даних і т.д.

Окремо варто відзначити досвід Facebook, який використовують MySQL без JOIN-вибірок. Така стратегія дозволяє їм значно легше масштабувати БД, переносючи при цьому навантаження з БД на код, масштабується простіше БД.

### **Масштабування коду**

Складнощі з масштабуванням коду залежать від того, скільки розподілених ресурсів необхідно хостам для роботи вашої програми. Чи будуть це тільки сесії, або буде потрібно загальний кеш і файли? У будь-якому випадку в першу чергу потрібно запустити копії програми на декількох хостах з однаковим оточенням.

Далі необхідно налаштувати балансування навантаження / запитів між цими хостами. Зробити це можна як на рівні TCP (HAProxy), так і на HTTP (Nginx) або DNS.

Наступним кроком потрібно зробити так, що б файли статички, кеш і сесії веб-додатки були доступні на кожному хості. Для сесій можна використовувати сервер, який працює через мережу (наприклад, Memcached). Як сервер кеша цілком розумно використовувати той же Memcached, але на іншому хості.

Файли статички можна змонтувати з якогось загального файлового сховища по NFS / CIFS або використовувати розподілену (HDFS ФС, GlusterFS, Ceph).

Так само можна зберігати файли в БД (наприклад, Mongo GridFS), вирішуючи тим самим проблеми доступності та масштабованості (з урахуванням того, що для NoSQL БД проблема масштабованості вирішена за рахунок шардінга).

Окремо варто відзначити проблему розміщення на кілька хостів. Найпростішим рішенням того щоб споживач, натискаючи «Оновити», не бачив різні версії програми буде виключення з конфігураційного фай-

ла балансувальника навантаження (веб-сервера) не оновлених хостів, і послідовного їх включення в міру оновлення. Так само можна прив'язати користувачів до конкретних хостам по куки або IP. Якщо ж оновлення вимагає значних змін у БД, найпростіше, взагалі тимчасово закрити проект.

### **Масштабування ФС**

При необхідності зберігання великого обсягу статички можна виділити дві проблеми: брак місця і швидкість доступу до даних. Як вже було написано вище, проблему з нестачею місця можна вирішити як мінімум трьома шляхами: розподілена ФС, зберігання даних в БД з підтримкою шардінга і організація шардінга «вручну» на рівні коду.

При цьому варто розуміти, що роздача статички теж не найпростіше завдання, коли мова йде про високі навантаження. Тому в цілком резонно мати безліч серверів призначених для роздачі статички. При цьому, якщо ми маємо загальне сховище даних (розподілена ФС або БД), при збереженні файлу ми можемо зберігати його ім'я без урахування хоста, а ім'я хоста підставляти випадковим чином при формуванні сторінки. У разі, коли шардінг реалізується вручну (тобто, за вибір хоста, на який будуть залиті дані, відповідає логіка в коді), інформація про хоста заливки повинна або обчислюватися на основі самого файлу, або генеруватися на підставі інших даних (інформація про користувача, кількості місця на дисках-сховищах) і зберігатися разом з ім'ям файлу в БД.

### **Висновок**

В статті коротко розглянуто безліч варіантів рішень проблем масштабування веб-додатків. Кожен з них має свої переваги і недоліки. Не існує деякого рецепта, як зробити все добре і відразу — для кожного завдання знайдеться безліч рішень зі своїми плюсами і мінусами.

### **Література**

1. John H. Howard. Scale and performance in a distributed file system/ John H. Howard, Michael L. Kazar, Sherri G. Menees, David A. Nichols, M. Satyanarayanan, Robert N. Sidebotham, and Michael J. ACM Transactions on Computer Systems, 1988.
2. Title: RESTful Web Services By: Leonard Richardson, Sam Ruby Publisher: O'Reilly Media May, 2007 P. 454.
3. Clifford Neuman. Scale in Distributed Systems, 1994.
4. Martin Kalin Java Web Services: Up and Running, 2nd Edition, O'Reilly, August 2013, P. 360.



**Якимець Роман Вікторович**

*студент*

*Національний технічний університет України «Київський політехнічний інститут»*

**Якимець Роман Викторович**

*студент*

*Национальный технический университет Украины «Киевский политехнический институт»*

**Yakymets Roman V.**

*student*

*National Technical University of Ukraine «Kyiv Polytechnic Institute»*

## МЕТОДИ КЛАСТЕРИЗАЦІЇ ТА ЇХ КЛАСИФІКАЦІЯ

## МЕТОДЫ КЛАСТЕРИЗАЦИИ И ИХ КЛАССИФИКАЦИЯ

## METHODS OF CLUSTERING AND CLASSIFICATION

**Анотація.** Описано суть кластеризації. Досліджені методи кластеризації та їх особливості.

**Ключові слова:** кластеризація, добування даних, методи кластеризації, K-середніх.

**Аннотация.** Описана суть кластеризации. Исследованы методы кластеризации и их особенности.

**Ключевые слова:** кластеризация, добыча данных, методы кластеризации, K-средних.

**Summary.** Describe the essence of clustering. Investigated clustering methods and their features.

**Key words:** clustering, data mining, clustering methods, K-Means.

### Вступ

Існує безліч способів застосування кластерного аналізу. Найчастіше він виступає як інструмент, що дозволяє поглянути на дані в цілому. Також кластерний аналіз може використовуватись для попередньої обробки або як проміжний етап інших алгоритмів, таких як класифікації або прогнозування, чи для data mining. В задачах data mining за допомогою кластерного аналізу створюється комплексне зведення даних для класифікації, відбувається виявлення шаблонів, формування і перевірка гіпотез і т.і. Крім того, кластерний аналіз часто застосовується для виявлення даних, що «вибиваються» з-поміж інших, оскільки таким даним відповідають точки, розташовані на відстані від будь-якого кластера. Також кластерний аналіз використовується для стиснення та узагальнення даних.

### Кластерний аналіз

Кластер-колекція об'єктів даних містить схожі об'єкти в одному кластері. Це означає, що об'єкти є аналогічними один до одного в межах однієї групи, і в той же час вони досить різні, або пов'язані з об'єктами в іншій групі або в інших кластерах. Кластерний аналіз також називають кластеризацією або сегмента-

цією даних. Кластерний аналіз розподіляє даний набір точок даних в набір кластерів або груп. Ці точки даних якомога більше схожі в межах однієї групи та віддалені наскільки це можливо від інших груп. Кластерний аналіз відноситься до навчання без вчителя (unsupervised learning) з огляду на те, що на початку немає визначених класів. Це суттєво відрізняє його від класифікації, де потребується навчання з учителем (supervised learning) або завдання міток класу для побудови моделі класифікації.

### Класифікація алгоритмів кластеризації

Існує дві основні класифікації алгоритмів кластеризації:

1. Ієрархічні і неієрархічні (плоскі). Ієрархічні алгоритми будують систему вкладених розбиттів, тобто на виході алгоритму представляється дерево кластерів, з коренем у якості всієї вибірки і листками — у якості найменших кластерів. Неієрархічні алгоритми будують лише одне розбиття об'єктів на кластери.

2. Чіткі і нечіткі.

Чіткі алгоритми надають всім об'єктам вибірки відповідний номер кластера, що означає, що кожен об'єкт повинен відноситись лише до одного кластеру.

Нечіткі алгоритми надають кожному об'єкту у відповідність набір значень, які демонструють ступінь належності об'єкта до кластерів. Отже, кожен об'єкт відноситься до кожного кластеру з певною ймовірністю.

**Плоскі методи на прикладі K-Means**

Метод K-Means полягає в тому, щоб виявити угруповання в даних. Вхідна множина розділяється на K груп, при цьому мінімізується функція, що визначає відстані як суми квадратів помилок — Sum of Squared Errors (SSE):

$$SSE(C) = \sum_{j=1}^K \sum_{i=1}^{n_j} \|x_i^j - c_j\|^2.$$

Після цього ітеративно оптимізується якість такого поділу. Таким чином, K-секціонування — це метод, що розділяє набір даних D з n об'єктів в набір K кластерів.

Кожний кластер представляється центром кластера. Для K кластерів метод K-Means працює наступним чином:

1. Обирає K точок центроїдами.
2. В циклі виконує наступні дії до того моменту, поки не досягає критерія збіжності:
  - a. Формує K кластерів шляхом присвоєння кожної точки до найближчого до неї центроїда.
  - b. Перевизначає центроїди.
3. Алгоритм може використовувати різні міри відстані, наприклад Манхеттенську, Евклідову відстані.

Особливості методу полягають в наступному:

- обчислювальна складність  $O(tKn)$ , де n — це кількість об'єктів, K — кількість кластерів, t — кількість ітерацій. Звичайно  $K, t \ll n$ , тобто метод є ефективним;
- кластеризація може завершитись на локальному оптимумі, тому для високоякісного результату необхідна початкова ініціалізація;
- необхідно заздалегідь задати K — кількість кластерів;
- чутливість до «шумних» даних та значень, що сильно відрізняються;
- можливе застосування тільки для чисельних даних;
- неможливо будувати кластери неопуклої форми.

На даний час існує багато варіацій цього методу, що частково усувають недоліки, серед них: K-Medoids, K-Medians, K-Modes, K-means++, Intelligent K-Means, Genetic K-Means.

**Ієрархічні методи кластеризації**

Ієрархічна кластеризація — це така кластеризація, за якої, починаючи з кластера, що складається з одного елемента, кластери ітеративно зливаються в кластери вищого рівня. Також можливо починати з єдиного великого макрокластера, який ітеративно розділяється на маленькі кластери. Таким чином формуються ієрархія кластерів. Для їх формування не потрібно задавати кількість кластерів K, такий тип кластеризації є більш детермінованим та не потребує ітеративних уточнень.

Ієрархічні методи кластеризації включають в себе дві категорії алгоритмів. Перша категорія має назву агломераційної. Вона починається з одноелементного кластера, що зливаються два кластери, щоб побудувати ієрархію кластерів «знизу вверх». Друга група — Divisive methods — методом розділення великий макрокластер, що містить всі елементи, розділяється на дві групи, кожна з них також на дві групи і так далі. Таким чином генерується ієрархія кластерів «зверху вниз». На рисунку 1 зображено обидва підходи.

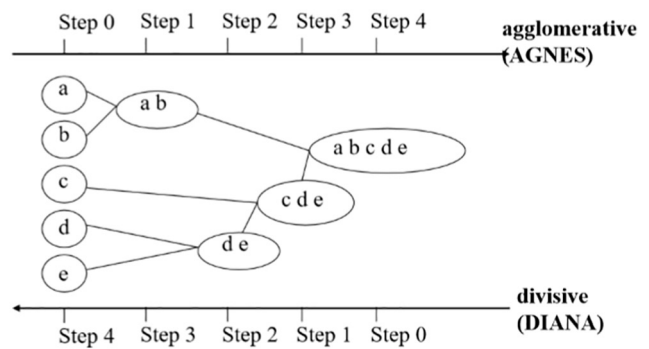


Рисунок 1. Хід алгоритмів AGNES, DIANA

**Агломеративні методи ієрархічної кластеризації**

Агломеративні алгоритми — це такі, що кластеризують «знизу вверх». На початку алгоритму кожна точка розглядається як кластер, потім алгоритм намагається об'єднати найближчі сусідні точки в один більший кластер і так далі, щоб зрештою об'єднати всі кластери в один великий кластер. Агломеративні алгоритми також називають AGNES (AGglomerative NESTing). Хід алгоритму виглядає наступним чином(рис.2):

- використовується метод одноканального зв'язку «найближчий сусід» та матриця відмінностей;
- вузли, що мають найменші відмінності, зливаються;
- всі вузли об'єднуються в один кластер.

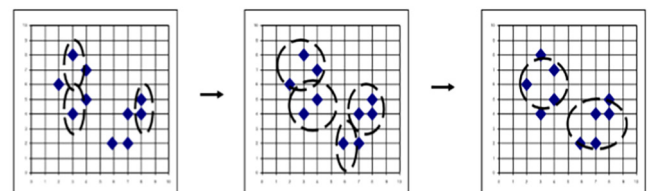


Рисунок 2. Хід алгоритму AGNES

Агломеративна кластеризація залежить від використання мір подібності кластерів:

- одноканальний зв'язок (найближчий сусід);
- повний зв'язок (діаметр);
- середній зв'язок (середнє по групі);
- центроїдний зв'язок (подібність центроїдів).

Одноканальний зв'язок (найближчий сусід):

- подібність двох кластерів — це подібність між їх найбільш подібними членами (найближчий сусід);
- приділяється увага найближчим точкам, ігнорується структура кластера;
- можливість будувати кластери неправильної форми;
- такий вид зв'язку чутливий до даних з шумами та значень, що вибиваються з множини. Схематичне зображення наведено на рисунку 3.

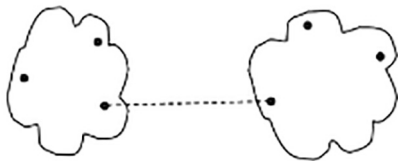


Рисунок 3. Одноканальний зв'язок

Повний зв'язок:

- подібність двох кластерів рахується як подібність їх найменш подібних членів;
- два кластери об'єднуються формують кластер з що найменшим діаметром;
- на виході — кластери компактної форми;
- чутливий до значень, що суттєво відрізняються.

Схематичне зображення наведено на рисунку 4.

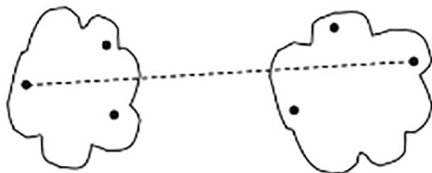


Рисунок 4. Повний зв'язок

Середній зв'язок — середня відстань між елементами в парі кластерів (рис. 5). Особливістю є затратне обчислення.

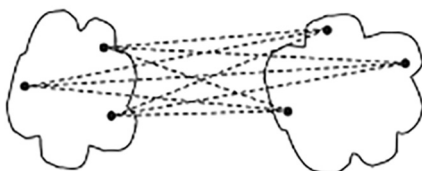


Рисунок 5. Середній зв'язок

Центроїдний зв'язок — відстань між центроїдами двох кластерів (рис. 6). Даний алгоритм не потребує задання кількості кластерів та дозволяє об'єднувати в кластери дані без істотних змін через значення, що вибиваються, та шуми.

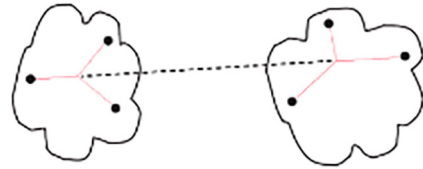


Рисунок 6. Центроїдний зв'язок

### Методи ієрархічної кластеризації розділенням

DIANA (Divisive Analysis) — зворотній порядок дій від AGNES: в результаті кожний елемент представляє собою кластер. На рис. 7 зображено хід алгоритму DIANA.

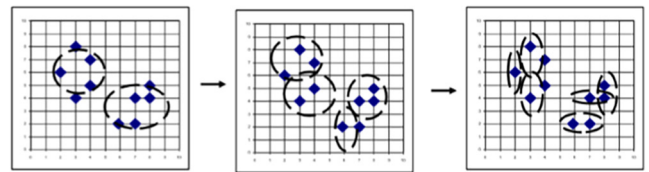


Рисунок 7. Хід алгоритму DIANA

Ієрархічна кластеризація розділенням — це підхід «зверху вниз»:

- процес починається з кореня, розглядаючи всі точки множини як кластер;
- кластери вищого рівня рекурсивно розщеплюються для побудови діаграми;
- може розглядатись в якості глобального підходу;
- може вважатись ефективнішим, але більш чутливим до шумів за AGNES.

### Висновки

В даній статті були описані та класифіковані деякі методи кластеризації. А саме, плоскі методи на прикладі K-Means та ієрархічні методи кластеризації. Також описані особливості цих методів.

### Література

1. A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988.
2. R. Ng and J. Han. Efficient and Effective Clustering Method for Spatial Data Mining. VLDB'9.
3. L. Kaufman and P.J. Rousseeuw. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, 199.
4. Котов А. Кластеризация данных./ Котов А., Красильников Н. 2006.

**Якимів Йосип Васильович**

*Кандидат технічних наук, доцент, доцент кафедри транспорту та зберігання нафти і газу  
Івано-Франківський національний технічний університет нафти і газу*

**Люта Наталія Вікторівна**

*Кандидат технічних наук, доцент, доцент кафедри транспорту та зберігання нафти і газу  
Івано-Франківський національний технічний університет нафти і газу*

**Бортняк Олена Михайлівна**

*Кандидат технічних наук, доцент, доцент кафедри транспорту та зберігання нафти і газу  
Івано-Франківський національний технічний університет нафти і газу*

**Якимив Иосиф Васильевич**

*Кандидат технических наук, доцент, доцент кафедры транспорта и хранения нефти и газа  
Ивано-Франковский национальный технический университет нефти и газа*

**Люта Наталия Викторовна**

*Кандидат технических наук, доцент, доцент кафедры транспорта и хранения нефти и газа  
Ивано-Франковский национальный технический университет нефти и газа*

**Бортняк Елена Михайловна**

*Кандидат технических наук, доцент, доцент кафедры транспорта и хранения нефти и газа  
Ивано-Франковский национальный технический университет нефти и газа*

**Yakymiv Yosyp**

*Ph.D., associate professor, department of transportation and storing of oil and gas,  
Ivano-Frankivsk National Technical University of Oil and Gas*

**Lyuta Nataliya**

*Ph.D., associate professor, department of transportation and storing of oil and gas,  
Ivano-Frankivsk National Technical University of Oil and Gas*

**Bortnyak Olena**

*Ph.D., associate professor, department of transportation and storing of oil and gas,  
Ivano-Frankivsk National Technical University of Oil and Gas*

**ВПЛИВ ТЕПЛА ТЕРТЯ ПОТОКУ НА ТЕМПЕРАТУРНИЙ  
РЕЖИМ ПЕРЕКАЧУВАННЯ ВИСОКОВ'ЯЗКИХ НАФТ  
ВЛИЯНИЕ ТЕПЛА ТРЕНИЯ ПОТОКА НА ТЕМПЕРАТУРНЫЙ  
РЕЖИМ ПЕРЕКАЧКИ ВЫСОКОВЯЗКИХ НЕФТЕЙ  
THE EFFECT OF THE HEAT OF FRICTION OF THE FLOW ON  
THE TEMPERATURE CONDITIONS WHEN PUMPING HIGH-VISCOSITY OILS**

**Анотація.** Досліджено вплив тепла тертя потоку на зміну температурного режиму при перекачуванні високов'язких нафт магістральними трубопроводами.

**Ключові слова:** тепло тертя потоку, коефіцієнт теплопередачі, магістральний трубопровід, високов'язка нафта, втрати напору на тертя.

**Аннотация.** Исследовано влияние тепла трения потока на изменение температурного режима при перекачке высоковязких нефтей по магистральным трубопроводам.

**Ключевые слова:** тепло трения потока, коэффициент теплопередачи, магистральный трубопровод, высоковязкая нефть, потери напора на трение.

**Summary.** The article explores the effect of the heat of friction of the flow on the change of the temperature conditions of pumping high-viscosity oils through crude oil pipelines.

**Keywords:** heat of friction of the flow, full heat transfer coefficient, main pipeline, high-viscosity oil, loss of the head due to friction

Частка високов'язких нафт в загальному обсязі видобування складає біля 10%. Економічним способом доставки таких нафт від місць видобування до нафтопереробних заводів є трубопровідний транспорт. Основним методом доставки нафт є перекачування їх трубопроводами з підігрівом. Суть способу полягає у нагріванні нафти на початку трубопроводу до певної температури, внаслідок чого зменшується її в'язкість, і наступного закачування її в трубопровід. При переміщенні нагрітої нафти трубопроводом її температура поступово зменшується, за рахунок чого в'язкість нафти збільшується. Швидкість зменшення температури залежить від теплообміну з оточуючим середовищем. Зниження температури дещо сповільнюється за рахунок тертя, що виникає при русі між шарами рідини та тертя рідини об стінки трубопроводу. Питання теплової взаємодії трубопроводу, яким перекачується нафта з підігрівом, з оточуючим середовищем та гідравлічних втрат висвітлені в роботах [1, с. 370, 2, с. 226, 3, с. 96–100]. За рахунок тепла тертя потоку зменшуються теплові та гідравлічні втрати, що недостатньо висвітлено в роботах. Тому виникає потреба у додатковому дослідженні впливу величини повного коефіцієнта теплопередачі на температурний та гідравлічний режим трубопроводів при врахуванні тепла тертя потоку.

Рівняння теплового балансу для ділянки трубопроводу довжиною  $dx$  із врахуванням тепла тертя потоку має вигляд [1, с. 385]

$$K\pi D(t - t_o)dx = -\rho Qc dt + \rho Qgidx, \quad (1)$$

де  $K\pi D(t - t_o)dx$  — тепло, яке передається в оточуюче середовище;

$\rho Qc dt$  — тепло, яке віддає нафта;

$\rho Qgidx$  — тепло тертя потоку;

$K$  — повний коефіцієнт теплопередачі;

$D$  — внутрішній діаметр трубопроводу;

$\pi D dx$  — поверхня, через яку проходить теплопередача;

$t$  — температура нафти на віддалі  $x$  від початку трубопроводу;

$t_o$  — температура оточуючого середовища на глибині укладання трубопроводу;

$\rho$  — густина нафти;

$Q$  — об'ємна витрата;

$c$  — масова питома теплоємність нафти;

$g$  — прискорення вільного падіння;

$i$  — гідравлічний нахил;

$idx$  — втрати напору на тертя на ділянці довжиною  $dx$ .

Згрупувавши члени в (1) і ввівши позначення

$$\Theta = \frac{\rho Qg}{K\pi D} i = \frac{\rho Qg}{K\pi D} \Delta_r \beta \frac{Q^{2-m} v^m}{D^{5-m}} = \phi v^m, \quad (2)$$

де  $\Delta_r$  — поправка на неізотермічність по радіусу труби;

$\beta$  і  $m$  — показники режиму руху рідини у формулі Лейбензона;

$v$  — кінематична в'язкість нафти;

$$\phi = \frac{\rho Q^{3-m} g}{K\pi D^{6-m}} \Delta_r \beta,$$

перепишемо вираз (1) із врахуванням (2) у вигляді

$$K\pi D(t - t_o - \Theta)dx = -\rho Qc dt. \quad (3)$$

У виразі (3) величина  $\Theta$  враховує тепло тертя потоку і має одиницю вимірювання температури.

Для зони гідравлічно гладких труб ( $m = 0,25$ ), в якій найчастіше рухаються високов'язкі нафти при перекачуванні з підігрівом, при незначній зміні в'язкості, тобто для трубопроводів невеликої протяжності величину  $\Theta$  приймають при середньому значенні температури і вважають постійною величиною  $\Theta = \Theta_{cp} = \text{const}$ . За такої умови після відокремлення змінних у виразі (3) і інтегрування одержано вираз

$$\frac{K\pi D}{\rho Qc} L = \ln \frac{t_n - t_o - \Theta}{t_x - t_o - \Theta}, \quad (4)$$

де  $L$  — довжина трубопроводу;

$t_n$  — температура нафти на початку трубопроводу (температура підігріву);

$t_x$  — температура нафти в кінці трубопроводу.

Із виразу (4) можна визначити температуру в кінці трубопроводу з врахуванням тепла тертя потоку

$$t_x = t_o + \Theta + (t_n - t_o - \Theta)e^{-\text{Шу}}, \quad (5)$$

де  $\text{Шу}$  — параметр Шухова,

$$\text{Шу} = \frac{K\pi D}{\rho Qc} L. \quad (6)$$

Відповідно температура в будь-якому перерізі на віддалі  $x$  від пункту підігріву буде дорівнювати

$$t = t_o + \Theta + (t_n - t_o - \Theta)e^{-\frac{K\pi Dx}{\rho Qc}}. \quad (7)$$

Температура нафти в кінці трубопроводу без врахування тепла тертя потоку дорівнює

$$t'_k = t_o + (t_n - t_o)e^{-\mu y}. \quad (8)$$

Кінцева температура нафти  $t_k$  при врахуванні тепла тертя потоку завжди більша за температуру  $t'_k$ , що одержана за формулою (8).

Використовуючи формулу (4), можна визначити відстань між пунктами підігріву

$$l_m = \frac{\rho Q c}{K \pi D} \ln \frac{t_n - t_o - \Theta}{t_k - t_o - \Theta}. \quad (9)$$

Необхідна кількість пунктів підігріву на трубопроводі дорівнює

$$n_m = \frac{L}{l_m}. \quad (10)$$

З температурним режимом трубопроводу безпосередньо пов'язані гідравлічні втрати в трубопроводі, визначення яких є напрямком наступних досліджень.

Запропонований алгоритм розрахунків може бути використаний проектними організаціями для прогнозування температурного режиму під час проектування трубопроводів для перекачування високов'язких і застигаючих нафт. Падіння температури нафти по довжині трубопроводу є менш інтенсивним за умов урахування тепла тертя потоку, тому під час експлуатації можуть бути вибрані енергоефективні способи регулювання режиму роботи такого нафтопроводу.

#### Література

1. Середюк М. Д. Трубопровідний транспорт нафти і нафтопродуктів: [підруч. для студ. вищ. навч. закл.] / М. Д. Середюк, Й. В. Якимів, В. П. Лісафін. — Івано-Франківськ: Кременчук, 2001. — 517 с.
2. Якимів Й. В. Типові технологічні розрахунки трубопровідного транспорту нафти і нафтопродуктів: [навч. посіб. для студ. вищ. навч. закл.] / Й. В. Якимів. — Івано-Франківськ: Факел, 2006. — 366 с.
3. Середюк М. Д. Дослідження особливостей гідравлічної характеристики неізотермічного нафтопроводу / М. Д. Середюк, В. Т. Болонний // Розвідка та розробка нафтових та газових родовищ. — 2003. — № 1(6). — С. 96–100.

**Яковець Михайло Вікторвич***студент**Національний технічний університет України «Київський політехнічний інститут»***Яковець Михаил Викторович***студент**Национальный технический университет Украины «Киевский политехнический институт»***Yakovets M.***student**National Technical University of Ukraine «Kyiv Polytechnic Institute»*

**ФОРМУВАННЯ РЕКОМЕНДАЦІЙ НА ОСНОВІ  
МОДЕЛІ ПРИХОВАНИХ ФАКТОРІВ  
ФОРМИРОВАНИЕ РЕКОМЕНДАЦИЙ НА ОСНОВЕ  
МОДЕЛИ СКРЫТЫХ ФАКТОРОВ  
FORMATION OF RECOMMENDATIONS BASED  
ON THE LATENT FACTOR MODEL**

**Анотація.** В статті проведено огляд існуючих підходів до формування рекомендацій. Запропоновано алгоритм формування рекомендацій на основі моделі прихованих факторів. Проведено порівняння запропонованого алгоритму з існуючими алгоритмами.

**Ключові слова:** машинне навчання, рекомендаційний алгоритм, колаборативна фільтрація, приховані фактори.

**Аннотация.** В статье проведен обзор существующих подходов к формированию рекомендаций. Предложен алгоритм формирования рекомендаций на основе модели скрытых факторов. Проведено сравнение предложенного алгоритма с существующими алгоритмами.

**Ключевые слова:** машинное обучение, рекомендательный алгоритм, колаборативных фильтрация, скрытые факторы.

**Summary.** In the article a review of existing approaches to the formation of recommendations was conducted. Formation of recommendations algorithm based on the model of latent factors was developed. The comparison of the proposed algorithm with existing algorithms was conducted.

**Key words:** machine learning, recommendation algorithm, collaborative filtering, latent factors.

**Вступ.** Основне завдання рекомендаційної системи — це надання персоналізованих рекомендацій користувачу, які враховують його уподобання при виборі предметів. Задача підвищення якості рекомендацій важлива тим, що з ростом обсягів даних, які зберігаються у мережі і пропонуються користувачу, зростає необхідність полегшення пошуку потенційно корисної інформації. Крім цього, компаніям, які займаються Інтернет-комерцією, підвищення якості рекомендацій дозволить збільшити продажі. Існує два основних типи алгоритмів формування рекомендацій [5]. Рекомендаційні алгоритми на основі вмісту. Даний підхід заснований на використанні даних з профілів користувачів і даних про об'єкти. Вміст профіля користувача може складатися не лише з історії його покупок чи

оцінок об'єктів, але й з великої кількості інших показників: вік, стать і т.д. Аналогічна ситуація з профілями об'єктів. Також існують колаборативні рекомендаційні алгоритми. Є множина користувачів  $u \in U$ , множина об'єктів  $i \in I$  (фільми, треки, товари і т.п.) і множина дій  $(r_{ui}, u, i, \dots) \in D$  (дії, які користувачі здійснюють з об'єктами). Кожна дія задається користувачем  $u$ , об'єктом  $i$ , своїм результатом  $r_{ui}$ . Результати формуються у вигляді матриці, наприклад, матриці рейтингів, які користувачі присвоїли об'єктам. Рекомендація формується на основі цієї матриці. Серед колаборативних методів фільтрації виділяють алгоритми на основі пошуку прихованих факторів користувачів і об'єктів. Суть алгоритму полягає в факторизації матриці рейтингів, тобто розбиття її на дві матриці, перемножуючи

стоці і рядки яких, можна передбачити значення рейтингів в початковій матриці. [3]

**Мета роботи.** Провести огляд існуючих підходів до формування списку рекомендацій. Запропонувати алгоритм колаборативної фільтрації на основі моделі прихованих факторів. Проаналізувати роботу розробленого алгоритму і провести його порівняння з іншими алгоритмами колаборативної фільтрації.

**Основна частина.**

**Модель прихованих факторів**

Далі розглядатимемо модель і алгоритм на основі даних з предметної області фільмів. Тобто, є множина користувачів  $u \in U$ , множина фільмів  $i \in I$ , оцінка, яку користувач поставив фільму  $r_{ui}$ . Усі оцінки зручно представляти як матрицю рейтингів.

Нижче описана спрощена модель представлення оцінок, на основі моделі — переможця конкурсу Netflix Prize. Для початкової моделі необхідна більша кількість структурованих даних, як наприклад, час виставлення рейтингу. Всі необхідні дані були надані учасникам в рамках конкурсу Netflix Prize. [4]

Проте, для роботи виключно з рейтингами фільмів, пропонується наведена нижче модель:

$$r_{ui} = \mu + b_i + b_u + \mathbf{q}_i * \mathbf{p}_u,$$

де  $r_{ui}$  — оцінка з матриці рейтингів  $R$ ;

$\mu$  — середній рейтинг по вибірці;

$b_i$  — базовий рейтинг користувача;

$b_u$  — базовий рейтинг фільму;

$\mathbf{q}_i$  — вектор факторів фільму;

$\mathbf{p}_u$  — вектор факторів користувача.

$b_u$  — характеризує базовий фактор користувача, як його оцінки відрізняються від середньої по вибірці. Цей фактор впливає на рейтинг незалежно від факторів фільму. Наприклад, якщо користувач ставить усім фільмам хороші оцінки, то базовий фактор буде високим, і — навпаки, якщо користувач жадібний на оцінки.

$b_i$  — характеризує базовий фактор фільму, його відносну якість. Цей фактор впливає на рейтинг незалежно від факторів користувача. Наприклад, якщо фільм краще знятий або більше розкручений, то й значення факторі буде вище, ніж у інших фільмів.

**Алгоритм формування рекомендацій на основі моделі прихованих факторів**

Якщо передбачити оцінки, які користувач поставив фільмам, то для вдалого формування рекомендацій необхідно у список рекомендацій додавати фільми з найвищими передбаченими оцінками. Щоб передбачити оцінки, треба знайти чисельні значення прихованих факторів, які формують оцінку.

Для цього по відомим значенням оцінок, які користувачі поставили фільмам (беруться з матриці рейтингів), знайдемо такі значення  $\mu, b_i, b_u, \mathbf{q}_i, \mathbf{p}_u \forall u, i$ , які найкраще наближають значення  $\mu + b_i + b_u + \mathbf{q}_i * \mathbf{p}_u$  до реальної оцінки, з точки зору квадрату похибки:

$$b_i, \mathbf{q}_i, \mathbf{p}_u = \arg \min_{b_i, \mathbf{q}_i, \mathbf{p}_u} \sum_{(i,u)} (r_{i,u} - \mu - b_i - b_u - \mathbf{q}_i^T \mathbf{p}_u)^2 + \lambda * (b_i^2 + b_u^2 + \|\mathbf{q}_i\|^2 + \|\mathbf{p}_i\|^2),$$

де  $\lambda$  — регуляризатор.

Регуляризація в машинному навчанні — метод додавання деякої додаткової інформації до умови з метою вирішити некоректно поставлене завдання або запобігти перенавчанню. Ця інформація часто має вигляд штрафу за складність моделі.

Для пошуку невідомих параметрів використано метод градієнтного спуску [2]. На кожному елементі вибірки невідомі параметри змінюються за даними формулами:

$$\begin{aligned} b_i &= b_i + \gamma * (e_{iu} - \lambda b_i), \\ b_u &= b_u + \gamma * (e_{iu} - \lambda b_u), \\ q_{ij} &= q_{ij} + \gamma * (e_{iu} p_{ij} - \lambda q_{ij}), \\ p_{ij} &= p_{ij} + \gamma * (e_{iu} p_{ij} - \lambda p_{ij}). \end{aligned}$$

Під час проходження усієї вибірки з даними, рахується помилка:

$$err = \mu - b_i - b_u - \mathbf{q}_i^T \mathbf{p}_i,$$

яка потім підноситься в квадрат і додається. Тобто після проходження вибірки підраховується RMSE. Якщо нове значення RMSE зменшується мало, то темп навчання збільшується вдвічі. Критерієм завершення навчання є зміна RMSE за цикл навчання менше, ніж на epsilon.

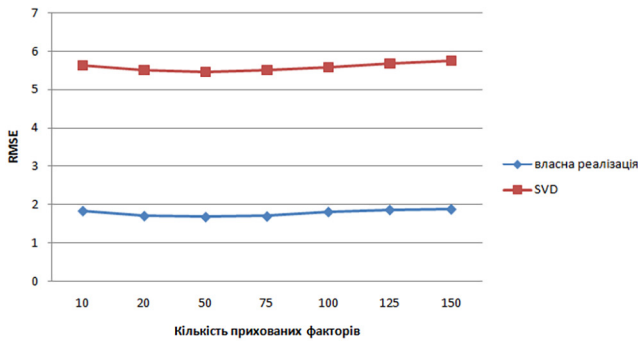
**Порівняння запропонованого алгоритму з іншим алгоритмом колаборативної фільтрації**

В ході експерименту, алгоритм навчався на початковій вибірці, в якій знаходилось 1.6 мільйонів оцінок користувачів, а перевірявся на тестовій вибірці розміром в 400 тисяч оцінок. Параметри алгоритму: регуляризатор — 2.5, epsilon — 0.000001.

Запропонований алгоритм було порівняно із алгоритмом колаборативної фільтрації з бібліотеки python-recsys [1], в основі якого лежить метод SVD. Якість роботи обох алгоритмів було оцінено за метриками RMSE і MAE.

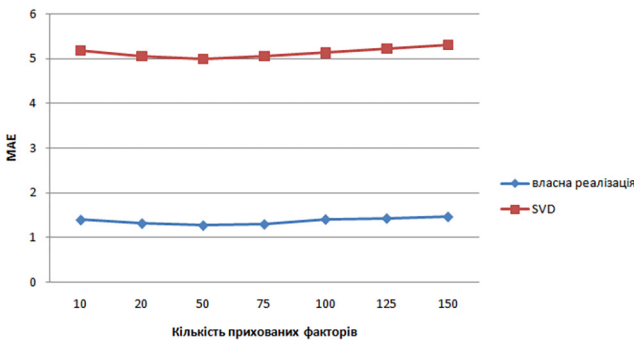
На графіку 1 зображено зміну оцінки похибки RMSE в залежності від кількості прихованих факторів для алгоритму SVD і власної реалізації:





Графік 1. Порівняння похибки за RMSE

На графіку 2 зображено зміну оцінки похибки MAE в залежності від кількості прихованих факторів для алгоритму SVD і власної реалізації:



Графік 2. Порівняння похибки за MAE

### Висновки

Стрімке зростання кількості даних різної природи, доступної користувачам, породжує проблему пошуку релевантної інформації. В даній роботі розв'язувалася задача побудови і аналізу алгоритму формування рекомендацій, який збільшує якість рекомендацій за обраними критеріями.

В роботі отримані наступні результати: проведено порівняння існуючих підходів до формування списку рекомендацій; запропоновано алгоритм колаборативної фільтрації на основі даних про рейтинги фільмів; проаналізовано роботу розробленого алгоритму. Власна реалізація алгоритму пошуку прихованих факторів показала кращі результати точності передбачення оцінок користувачів, ніж алгоритм SVD (з бібліотеки python-recsys) за критеріями RMSE і MAE.

### Література

1. Python-recsys on Github [Електронний ресурс]. — Режим доступу: <https://github.com/ocelma/python-recsys>
2. Вікіпедія — Метод стохастичного градієнта [Електронний ресурс]. — Режим доступу: [https://uk.wikipedia.org/wiki/Метод\\_стохастичного\\_градієнта](https://uk.wikipedia.org/wiki/Метод_стохастичного_градієнта)
3. Y. Koren, R. Bell, C. Volinsky, Matrix Factorization Techniques for Recommender Systems. IEE Computer Society, 2009.
4. R. Bell, Y. Koren and C. Volinsky. The BellKor Solution to the Netflix Prize. 2007.
5. Рекомендательные системы: Часть 1. Введение в подходы и алгоритмы [Електронний ресурс]. — Режим доступу: <https://www.ibm.com/developerworks/ru/library/os-recommender1/>

**Яременко Костянтин Миколайович**

*студент*

*Національний технічний університет України «Київський політехнічний інститут»*

**Яременко Константин Николаевич**

*студент*

*Национальный технический университет Украины «Киевский политехнический институт»*

**Yaremenko K. N.**

*Student*

*National Technical University of Ukraine «Kyiv Polytechnic Institute»*

## РОЗПІЗНАВАННЯ ЖЕСТИВ РУКИ

## РАСПОЗНАВАНИЕ ЖЕСТОВ РУКИ

## HAND GESTURE RECOGNITION

**Анотація.** Дана робота присвячена аналізу алгоритмів і методів розпізнавання об'єктів на зображенні в реальному часі, а також реалізації програми для жестів руки. Метою роботи є розробка програмного прототипу, який дозволяє розпізнавати окремі жести руки. Розроблений прототип розпізнає жести, що відповідають завантаженим шаблонам.

Програма реалізована за допомогою бібліотеки комп'ютерного зору OpenCV на мові програмування C++.

**Ключові слова:** комп'ютерний зір, OpenCV, контурний аналіз, колірний підхід, відстані Гаусдорфа.

**Анотация.** Даная работа посвящена анализу алгоритмов и методов распознавания объектов на изображении в реальном времени, а также реализации программы для распознавания жестов руки. Целью работы является разработка программного прототипа, который позволяет распознавать отдельные жесты руки.

Программа реализована с помощью библиотеки компьютерного зрения OpenCV на языке программирования C++.

**Ключевые слова:** компьютерное зрение, OpenCV, контурный анализ, цветовой подход, расстояния Хаусдорфа.

**Annotation.** Graduation work is devoted to analysis of algorithms and recognition methods in the image in real time, and program implementation for controlling a computer with hand gestures. The purpose is developing software that allows to recognize hand gesture.

The program is implemented through computer vision library OpenCV using the programming language C++.

**Keywords:** computer vision, OpenCV, contour analysis, color method, Hausdorff distance.

### Вступ

Метою роботи є розробка програмного прототипу для розпізнавання жестів руки. Для вирішення цієї задачі в роботі проводиться аналіз і обґрунтовується вибір відповідних методів та алгоритмів, які належать до оптичного підходу керування пристроями.

Для оптичної взаємодії користувача з комп'ютером використовується звичайна веб-камера без додаткових опціоналів.

Даний прототип стане основою для подальшої розробки програми керування персональним комп'ютером за допомогою жестів руки.

### 1. Огляд обраних методів для локалізації жесту та його розпізнавання

#### 1.1. Колірний простір HSV

Простір HSV (Hue – тон, Saturation – насиченість, Value – яскравість) являється інтуїтивною кольоровою моделлю, так як оперує тими ж характеристиками кольору, що і людина. Якщо уявити конкретний простір RGB у вигляді кубу, то вздовж головної діагоналі  $(0, 0, 0) - (1, 1, 1)$  розташовані сірі кольори. Координата V задає яскравість кольору і пропорційна відстані від початку координат і вздовж головної діагоналі. Її можна отримати за допомогою перпендикулярної проєкції точки в RGB на головну діагональ. Координата S визначає насиченість. В просторі RGB це виглядає як

відстань від головної діагоналі. Чим вона більша, тим ближче колір до чистих кольорів, розташованих у вершинах кубу. Координата  $H$  – це колірний тон. Якщо розглянути площину, що перпендикулярна до головної діагоналі кубу, то кут повороту вектору кольору в цій площині і визначає тональність. Перетворення із RGB простору в HSV, розраховується за формулами (1–3)

$$V \leftarrow \max(R, G, B) \quad (1)$$

$$S \leftarrow f(x) = \begin{cases} \frac{V - \min(R, G, B)}{V}, & V \neq 0 \\ 0, & V = 0 \end{cases} \quad (2)$$

$$V \leftarrow \begin{cases} \frac{60(G - B)}{S}, & V = R \\ 120 + \frac{60(B - R)}{S}, & V = G \\ 240 + \frac{60(G - B)}{S}, & V = B \end{cases} \quad (3)$$

### 1.2. Метод відстаней Гаусдорфа

В математиці відстань Гаусдорфа або метрика Гаусдорфа, яку також називають Помпейо-Гаусдорфою відстанню, міра яка визначає, як далеко одна від одної знаходяться дві підмножини метричного простору. Виходить множина непустих компактних підмножин метричного простору.

Нехай  $X$  і  $Y$  дві не пусті підмножини метричного простору  $(M, d)$ . Де їх відстань Гаусдорфа  $d_H(X, Y)$  розраховується за формулою (4), що еквівалентно формулі (5).

$$d_H(X, Y) = \max \left\{ \max_{x \in X} \min_{y \in Y} d(x, y), \max_{y \in Y} \min_{x \in X} d(x, y) \right\} \quad (4)$$

$$d_H(X, Y) = \min \{ d_H(X, Y); X \subseteq Y_\epsilon \cap Y \subseteq X_\epsilon \} \quad (5)$$

де  $X_\epsilon = \bigcup \{z \in M; d(z, X) \leq \epsilon\}$

Тобто  $\epsilon$  – множина всіх точок в межах  $\epsilon$  множини  $X$ .

## 2. Реалізація прототипу

### 2.1. Виділення шкіри

Щоб локалізувати руку спочатку необхідно виділити регіони кольору шкіри. Для цього була в ручну виділена шкіра людини на зображенні за умов природнього і штучного освітлення (рисунок 1–2).

Так як задача полягає в знаходженні шкіри при будь-якому освітленні, то для регіону, що нас цікавить можна звзвати множину значень кольорового простору HSV, а саме  $0 \leq H \leq 30, 15 \leq S \leq 200$ . Також було прийнято рішення обмежити компоненту  $V$  значеннями  $15 \leq V \leq 250$ , щоб виключити затемнення і засвічування області. Користувач може сам задати потрібні значення через відповідний пункт меню (рисунок 4).

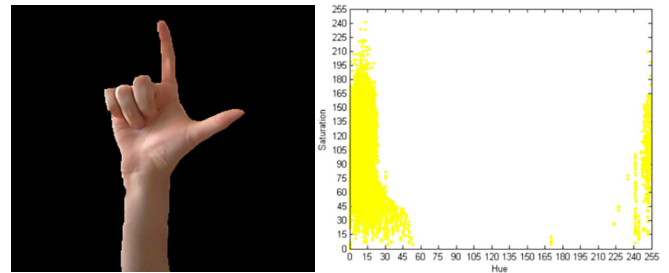


Рисунок 1. Зображення шкіри при природному освітленні та значення насиченості та яскравості в ненульових пікселях

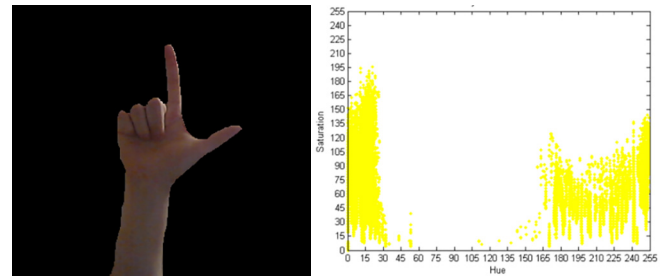


Рисунок 2. Зображення шкіри людини при штучному освітлення та значення насиченості та яскравості в ненульових пікселях

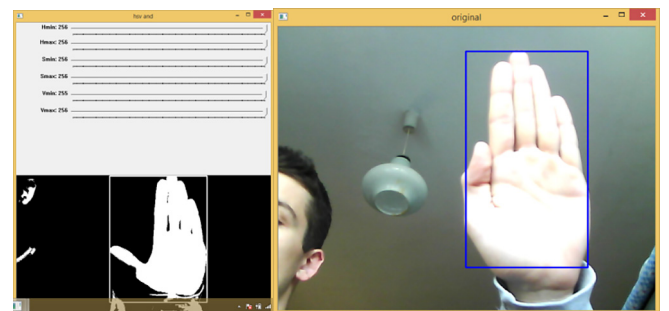


Рисунок 3. Вхідний кадр із локалізованою рукою і бінаризований регіон шкіри

### 2.2. Розпізнавання жесту руки

Після того, як рука була знайдена, із регіону зв'язної компоненти видаляються всі інші компоненти, колір помітки яких відрізняється від кольору помітки компоненти руки. Це робиться для того, щоб позбавитись від непотрібного сміття і в наслідку для покращення точності алгоритму.

Далі локалізований регіон масштабується в зображення формату  $100 \times 100$ , після чого на отриманому зображенні видаляються пустоти, робиться це за допомогою методів ерозії та дилатації.

Дилатація (морфологічне розширення) – згортка зображення або виділеної області зображення з деяким ядром. Ядро може мати довільну форму розмір. При цьому в ядрі виділяється єдина ведуча позиція, що суміщається з поточним пікселем при розрахунку згортки. В багатьох випадках в якості ядра обирається квадрат або круг з ведучою позицією в центрі. Ядро можна розглядати як шаблон або маску. Застосування

дилатації зводиться до проходу шаблону по всьому зображенню і застосуванню оператора пошуку локального максимуму до інтенсивностей пікселів зображення, що накриваються шаблоном. Така операція викликає ріст світлих областей на зображенні.

Ерозія (морфологічне звуження) — зворотна операція до дилатації. Дія ерозії подібна дилатації різниця полягає лиш в тому, що використовується оператор пошуку локального мінімуму.

Комбінування цих методів при застосування дозволяє позбавитись шумів і згладити контур зображення. Після ліквідації пустот, на зображенні виділяється контур за наступним алгоритмом:

1. У 8-зв'язній області помічаються точки, у яких є всі сусіди білого кольору ба помічені на попередніх кроках.

2. Помічені точки видаляються.

Коли було виділено контур жесту, відбувається порівняння його із набором жестів-шаблонів за допомогою методу відстаней Гаусдорфа. Алгоритм полягає у знаходженні критерію схожості двох контурів, що приймає значення найменшої відстані між шуканим об'єктом і шаблоном. Якщо цей критерій переважає задане допустиме значення — вхідне зображення ігнорується, у іншому випадку зображення вважаються схожими і програма відображає знайдений шаблон, після чого виконується наступна ітерація циклу. Результати роботи програми зображені на рисунку 4.

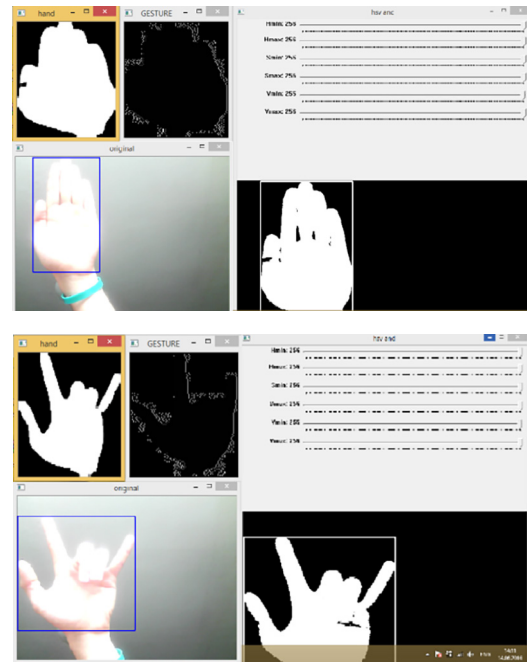


Рисунок 4. Результати роботи програми

### Висновки

В роботі коротко розглянуті обрані методи для локалізації та розпізнавання жестів руки. Описаний загальний алгоритм роботи програми. Наведено приклад роботи розробленої програми.

### Перелік посилань

1. R. Tyrrell Variational Analysis / R. Tyrrell // Springer-Verlag. — 2005. № 17. — P. 117.
2. Дж. Стокман Компьютерное зрение / Дж. Стокман, Л. Шапиро. — М.: БИНОМ. Лаборатория знаний, 2006. — 36 с.
3. Munkres Topology / Munkres, James // Prentice Hall. — 1999. — P. 280–281.

**Бондаренко Олексій Сергійович, Івченко Дмитро Анатолійович**

*студенти*

*Національний технічний університет України «Київський політехнічний інститут»*

**Бондаренко Алексей Сергеевич, Ивченко Дмитрий Анатольевич**

*студенты*

*Национальный технический университет Украины «Киевский политехнический институт»*

**Bondarenko O., Ivchenko D.**

*students*

*National Technical University of Ukraine «Kyiv Polytechnic Institute»*

## ТЕХНОЛОГІЇ АНАЛІЗУ РІВНЯ ГЛЮКОЗИ В ОРГАНІЗМІ ЛЮДИНИ

## ТЕХНОЛОГИИ АНАЛИЗА УРОВНЯ ГЛЮКОЗЫ В ОРГАНИЗМЕ ЧЕЛОВЕКА

## TECHNOLOGY OF ANALYZE OF GLUCOSE LEVEL IN THE HUMAN BODY

**Анотація.** *Огляд технологій аналізу рівня цукру в організмі людини.*

**Ключові слова:** *технології, аналіз рівня глюкози, діабет, медицина.*

**Аннотация.** *Обзор технологий анализа уровня глюкозы в организме человека.*

**Ключевые слова:** *технологии, анализуровняглюкозы, диабет, медицина.*

**Summary.** *Review of technologies of analyze of the glucose level in a human body.*

**Keywords:** *technologies, analyzeofglucoselevel, diabetes, medicine.*

### Вступ

Для того щоб контролювати рівень глюкози в крові, хворим на цукровий діабет, як правило, доводиться виконувати болючі і незручні аналізи крові, роблячи прокол на пальці, в деяких випадках, кілька разів на день. Альтернативою такому «механічному аналізу» може стати імплантований датчик моніторингу глюкози. Він встановлюється хірургічним шляхом, а згодом має видалятися для заміни.

Сучасні технології пропонують більш прості та зручні неінвазивні методи аналізу глюкози, такі як лазерний аналіз, спектральний аналіз, біосенсори та інші.

### Глюкозооксидазний метод

Глюкозооксидазний (фотометричний) метод аналізу базується на реакції окислення глюкози в присутності ферменту глюкозооксидази з утворенням перекисі водню, яка в свою чергу в присутності пероксидази окислює ортотолідін з утворенням забарвлених продуктів. О концентрації глюкози роблять висновок аналізуючи кількість забарвлених продуктів.

Найбільш важливою частиною глюкометру є тест-смужка. Вона отримує зразок крові та передає виміри на мікроконтролер через аналогово-цифровий перетворювач. Це 3 основних елементи глюкометру. Інші компоненти є опціональними або периферійними. На схемі відображена базова схема компонентів глюкометру.

Сенсор у тест-смужці використовує платиновий та срібний електроди для електролізу перекисі водню. В результаті реакції окислення глюкози на глюкозооксидазній мембрані виділяється пероксид водню. Визначити концентрацію глюкози можна вимірюючи силу струму, що проходить через результати реакції.

У портативних глюкометрах використовується постійна сила струму у 0.3V. Залежність сили струму від концентрації глюкози лінійна в діапазоні від 5 до 30 mmol/L. Швидкість виміру становить приблизно 20 секунд.

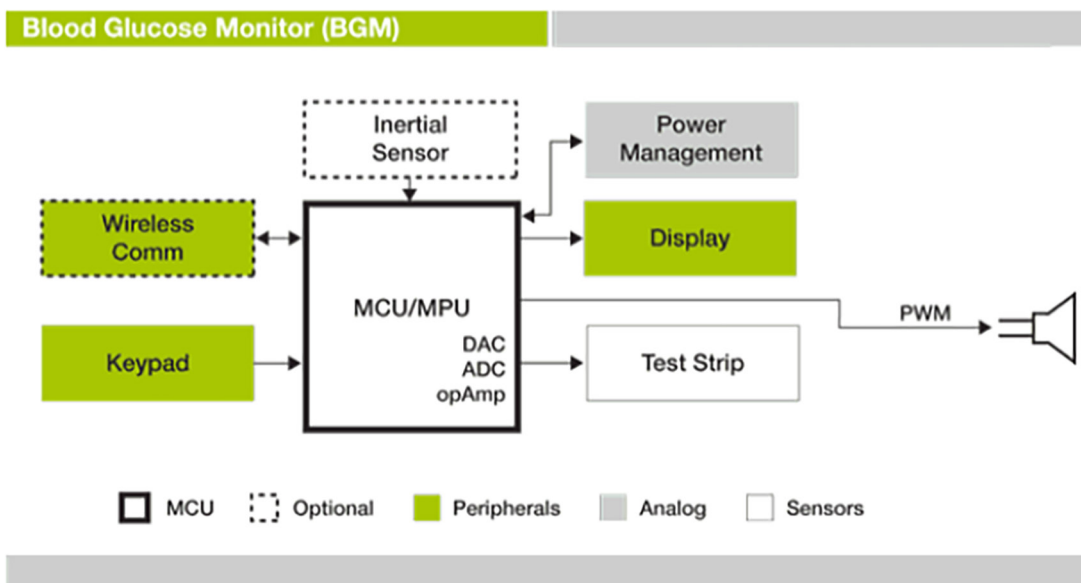


Рисунок 1. Схема компонентів глюкометра [6]

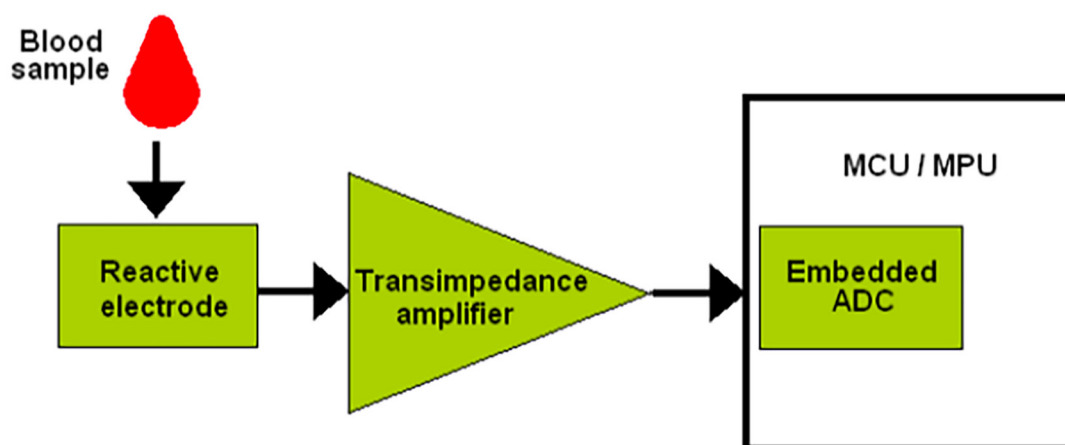


Рисунок 2. Схема компонентів тест-смужки [6]

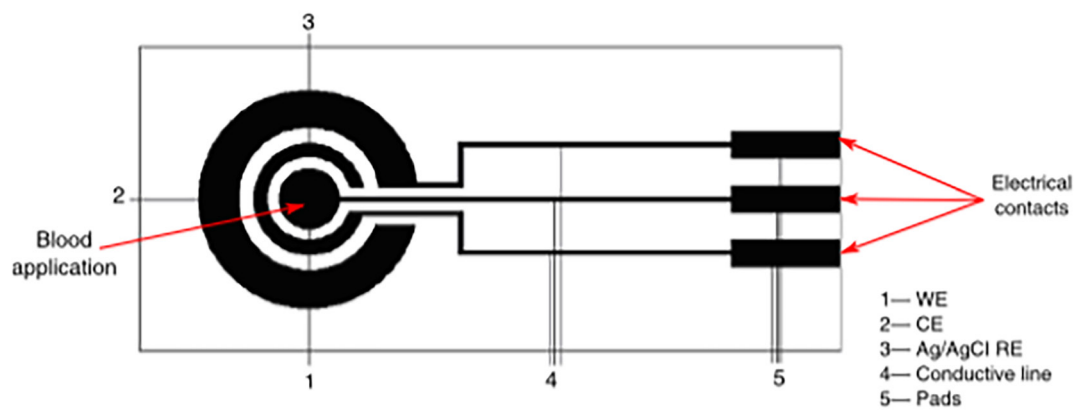


Рисунок 3. Схема побудови чіпу тест-смужки [6]

### Лазерний аналіз

Лазерний аналіз базується на здатності глюкози до поглинання інфрачервоного світла. Базовий прилад для аналізу рівня глюкози складається з лазера, детектору та процесору для обробки результатів. Одним з прикладів системи лазерного аналізу є GlucoSense.

Система GlucoSense була розроблена професором Гін Хосе та його командою в Університеті Лідса [1]. Пацієнту потрібно помістити подушечку пальця навпроти невеликого скляного віконця на пристрої. Малопотужний лазерний промінь ближнього інфрачервоного світла проектується через це вікно на палець. Частина світла поглинається глюкозою в крові, а інша відбивається назад на пристрій.

Відбите проміння, потрапляючи на скляну поверхню, починає світитися в інфрачервоному спектрі. Чим більше світла відіб'ється, тим довше буде світіння.

Шляхом вимірювання тривалості цієї флуоресценції, процесор в пристрої здатний визначити, скільки вихідного лазерного світла поглинається глюкозою, таким чином можна визначити кількість глюкози в крові. Весь процес займає менше 30 секунд.

Ця технологія вже пройшла період клінічних випробувань та процес комерціалізації. Розробники анонують 2 комерційні пристрої: настільний пристрій та портативний, що вимірює рівень глюкози безперервно.

Інший метод використання лазера розробляють в університеті Тохоку (Японія). Дослідники вважають, що використання ближнього інфрачервоного світла не повністю підходить для вимірювань, так як воно поглинається не тільки глюкозою, а ще й водою, білком та гемоглобіном. Тому це зменшує точність таких вимірювань.

Їм вдалося подолати обмеження близького інфрачервоного освітлення, зосередившись на дальньому інфрачервоному світлі. Хвилі довжиною близько 10 мікрон сильніше поглинаються глюкозою. З іншого



Рисунок 4. Лазерний детектор [14]

боку таке світло проникає тільки на кілька мікрон під шкіру. Так що для того, щоб правильно виявити вміст глюкози в крові за допомогою цього методу, команда була змушена придумати новий спосіб застосування і вимірювання його наслідків.

Їхній пристрій складається з гнучких оптичних волокон, які випромінюють інфрачервоне світло через невелику призму на кінці. Застосовуючи його до внутрішньої частини рота, де слизова оболонка вистилає внутрішні губи дослідники змогли виміряти рівень глюкози в крові з менш ніж 20% похибки.

В університеті Принстона розробляється пристрій, що використовує інфрачервоне світло в середньому діапазоні. Використання цього спектру світла дозволяє уникнути абсорбції променю водою та білками, але в той же час існують труднощі з генерацією такого променю. Потрібен досить потужний та стабільний лазер для того, щоб промінь зміг проникнути крізь шкіру. Дослідники знайшли вихід, побудувавши новий тип квантово-каскадного лазера [2].

Квантово-каскадний лазер може виробляти одну з безліч частот. Це можливо через проходження електронів крізь каскад напівпровідникових шарів. Невдавні розробки також забезпечують підвищену потужність та стабільність, що дозволяє дослідникам проводити дослідження з середнім інфрачервоним світлом.

При тестуванні дослідники вимірювали рівень цукру у 3 здорових людей до та після прийому солодкої їжі. Результати аналізу лазером порівнювалися з результатами аналізу крові. Ці виміри проводились регулярно протягом декількох тижнів. Точність приладу склала 84 відсотки.

### Раман-спектроскопія

Раман-спектроскопія — вид спектроскопії, в основі якої лежить здатність досліджуваних систем (молекул) до непружного (раманівського) розсіювання монохроматичного світла [3].

Суть методу полягає в тому, що через зразок досліджуваної речовини пропускають промінь із певною довжиною хвилі, який при контакті зі зразком розсіюється. Отримані промені за допомогою лінзи збираються в один пучок і пропускаються через світлофільтр, що відділяє слабкі (0,001% інтенсивності) раманівські промені від значно інтенсивніших (99,999%) релєївських. «Чисті» раманівські промені підсилюються і спрямовуються на детектор, який фіксує частоту їхнього коливання.

Спектроскопія дозволяє ідентифікувати хімічні сполуки. Цей метод для аналізу рівня глюкози протягом 15 років розробляють у лабораторії спектроскопії у Масачусетському технологічному інституті (MIT).

Метод дозволяє провести аналіз глюкози через просте сканування руки пацієнта або пальця ближнім інфрачервоним світлом.

Дослідники з МІТ розробили малий Раман-спектрограф, розміром з ноутбук, який може бути встановлений в кабінеті лікаря.

Одною з головних перешкод, з якою стикалися дослідники полягає в тому, що в ближньому інфрачервоному діапазоні світло проникає лише близько на півміліметра під шкіру, так що спектрограф вимірює кількість глюкози в рідині, яка омиває клітини шкіри (відома як міжклітинна рідина), а не кількість глюкози в крові. Щоб подолати це, команда придумали алгоритм, який пов'язує між собою дві концентрації, що дозволяє їм прогнозувати рівень глюкози в крові на базі концентрації глюкози в міжклітинній рідині.

Проте точність такого калібрування значно зменшується після прийому їжі, тому що рівень глюкози в крові збільшується швидко, на відміну від рівня в міжклітинній рідині — це може займати від 5 до 10 хвилин.

Аби поліпшити часову кореляцію дослідники розробили новий алгоритм динамічної концентраційної корекції (DCC), що включає швидкість, з якою глюкоза дифундує з крові в тканинну рідину. Завдяки цьому алгоритму вдалось підвищити точність вимірювань в середньому на 15 відсотків, а в деяких випадках на 30%.

### **Розумні контактні лінзи**

Потенціал розумних контактних лінз для лікування діабету, вперше став очевидним у 2009 році, коли професор Джин Чжан з університету Західного Онтаріо інтегрував електронні наноконпоненти в гідрогелеві лінзи. Ця система контролювала сльози на підвищення або зниження рівня цукру. В тому випадку, якщо рівень глюкози був небезпечний, система викликала хімічну реакцію, що змінювало колір лінзи.

У 2012 році цією технологією зацікавились у Microsoft. Вони розпочали спільний проект з Університетом штату Вашингтон в рамках якого намагались інтегрувати електроніку в контактні лінзи для бездротової передачі рівня глюкози. [4]

У 2014 року проект розумних контактних лінз анонсувала компанія Google. Команда проекту під керівництвом Брайана Отіс і Бабак Парвіз в лабораторії Google X, розробила бездротової чіп і мініатюрний датчик глюкози, що вставляються між двома шарами м'якого матеріалу для контактних лінз. Такі смарт-контактні лінзи, за даними Google, здатні отримувати рівень глюкози кожну секунду. Команда проекту працює над функцією, що могла б попереджати користувача, коли глюкоза знаходиться на небезпечних рівнях за допомогою вбудованого діоду. [5]

Google заявляє, що вони завершили численні клінічні наукові дослідження і в даний час ведуть переговори з управлінням з якості харчових продуктів та медикаментів США (FDA). Вони також шукають партнерів для розробки додатків аби підвищити зручність роботи з пристроєм для лікарів і пацієнтів.

### **Біосенсори**

Замість аналізу проб крові, біосенсор здатен безперервно вимірювати рівень глюкози через аналіз таких рідин, як пот чи сльози. Один із таких сенсорів розробляється в інституті Фраунгофера (FraunhoferInstitut eforMicroelectronicCircuitsandSystems — IMS).

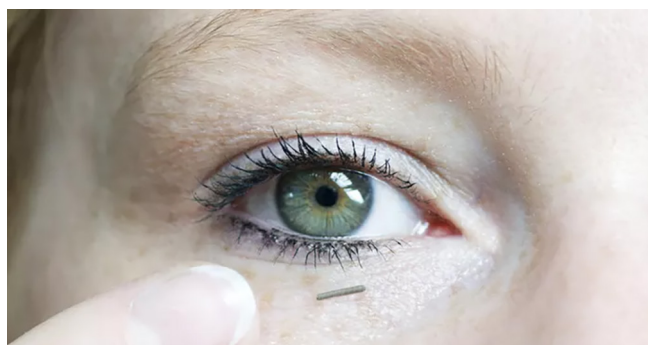


Рисунок 5. Біосенсор біля ока людини [11]

В той час як сама ідея біосенсорів не нова, попередні реалізації подібних пристроїв були надто великими, неточними та споживали забагато енергії. Новий біосенсор складається з чіпа, розмір якого всього 0.5 x 2 mm та яких споживає менше 100 мікроампер на п'ять вольт.

Чіп пристрою інтегрує нано-потенціостату, що вимірює концентрації перекиси водню (H<sub>2</sub>O<sub>2</sub>) та інших хімічних речовин, які виникають в наслідок електрохімічної реакції. Реакція виникає за допомогу ферменту — глюкозооксидази (GOx). Пристрій використовує концентрації цих хімічних речовин для розрахунку рівня глюкози користувача. Також в пристрій інтегровані аналого-цифровий перетворювач, що перетворює електрохімічні сигнали в цифрові дані, та передавач, які може передавати дані по бездротовій мережі на мобільних пристрій.

Інший біосенсор був розроблений датською компанією NovioSense. Він встановлюється між вікою та глазом та має технологію вимірювання схожу з вищеописаною.

NovioSense BV складається з декількох проводів, що поєднуються у формі пружини та утворюють мікро-електрохімічний осередок. Спіральна форма надає пристрою гнучкості та може відповідати контуру навколишнього середовища. Порожній сердечник катушки забезпечує корпус для нано-потенціостату



і нано-чіпи передавачі, які дозволяють передавати вимірювання на мобільний пристрій. Захисний шар пристрою складається з гідрогелю, що утворює гладку поверхню. Гнучкість і м'який гель покриття означає, що датчик може бути використаний в якості неінвазивного пристрою моніторингу без дискомфорту.

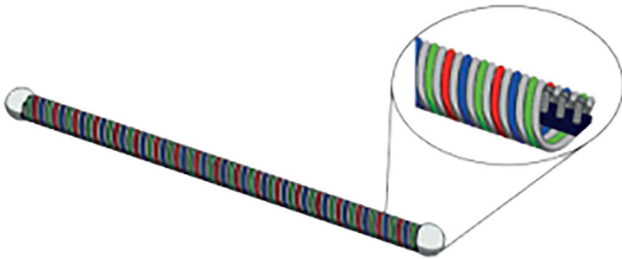


Рисунок 6. Схема пристрою NovioSense BV [12]

### Тату-монітори

В Каліфорнійському університету, Сан-Дієго аспірантом AmapBandodkar, розробляється прототип гнучкого пристрою, що складається з електродів, надрукованих на папері для тимчасового татуювання.



Рисунок 7. Тату монітор [14]

У лабораторних тестах, тату-монітори були застосовані до та після прийому високо вуглеводної їжі. Татуювання виявило сплеск рівня глюкози так само точно, як і традиційний тест.

На даний час тату-монітори не забезпечують числовий аналіз рівня глюкози. Для цієї мети розробляється окремий пристрій. Команда також працює над створенням більш витривалих татуювань, так як вони на даний час здатні витримувати близько доби після нанесення на шкіру. На щастя, вони дуже недорогі [6].

Дослідники також розробили чорнило та ручку, за допомогою яких можна намалювати сенсор безпосередньо на поверхні тіла. Чорнило безпечне для застосування в організмі людини, і може зберігатися до застосування протягом тривалого періоду часу. Було підраховано, що чорнила в одній ручці було б достатньо для близько 500 індивідуальних тестів [7].

### Аналіз слини

У 2012 році вчені з Університету Брауна Род-Айленд розробили прототип біочіпу для виявлення дуже низьких концентрацій глюкози в слині.

Біочіп має квадратну форму розміром в один дюйм, який покритий шаром срібла. В це срібло інтегровані тисячі нано-інтерферометрів. Кожен з них складається з однієї щілини, та двох каналів; щілини шириною 100 нанометрів, яка проходить прямо через срібло та каналів шириною 200 нм, що йдуть паралельно щілини з обох боків.

Коли світло проходить крізь рідину, деякі фотони захоплюються щілинами, в той час як інші фотони розсіюються канавками та взаємодіють з вільними електронами на поверхні срібла. Ці взаємодії призводять до коливань, відомих як поверхневі плазмон [8].

Коли фотони, що проходять по каналам, зустрічаються з фотонами, замкненими в щілині, виникає інтерференція. Величина інтерференції залежить від концентрації глюкози в рідині, а також впливає на кількість світла, яке здатне пройти крізь щілину. Вимірюючи інтенсивність світла, що проходить крізь кожен щілину, датчик може визначити концентрацію глюкози в рідині.

### Висновки

В межах даної статті були розглянуті технології аналізу рівня глюкози в організмі людини. Серед них були описані глюкооксидазний метод, лазерний метод, метод рама-спектроскопії, біосенсори та інші. Глюкооксидазний метод, що використовує для аналізу рівня глюкози в крові хімічну реакцію окислення оксидази та вимірює силу струму у продуктах реакції. Лазерний метод базується на принципі поглинання глюкозою світла в інфрачервоному діапазоні. Розумні контактні лінзи здатні аналізувати рівень цукру в сльозі кожної секунди та змінювати колір при підвищенні показників встановленої норми. Біосенсори, найменші з розглянутих технологій, аналізують піт та сльози. Тату монітори аналізують піт та змінюють колір при змінах в показниках.

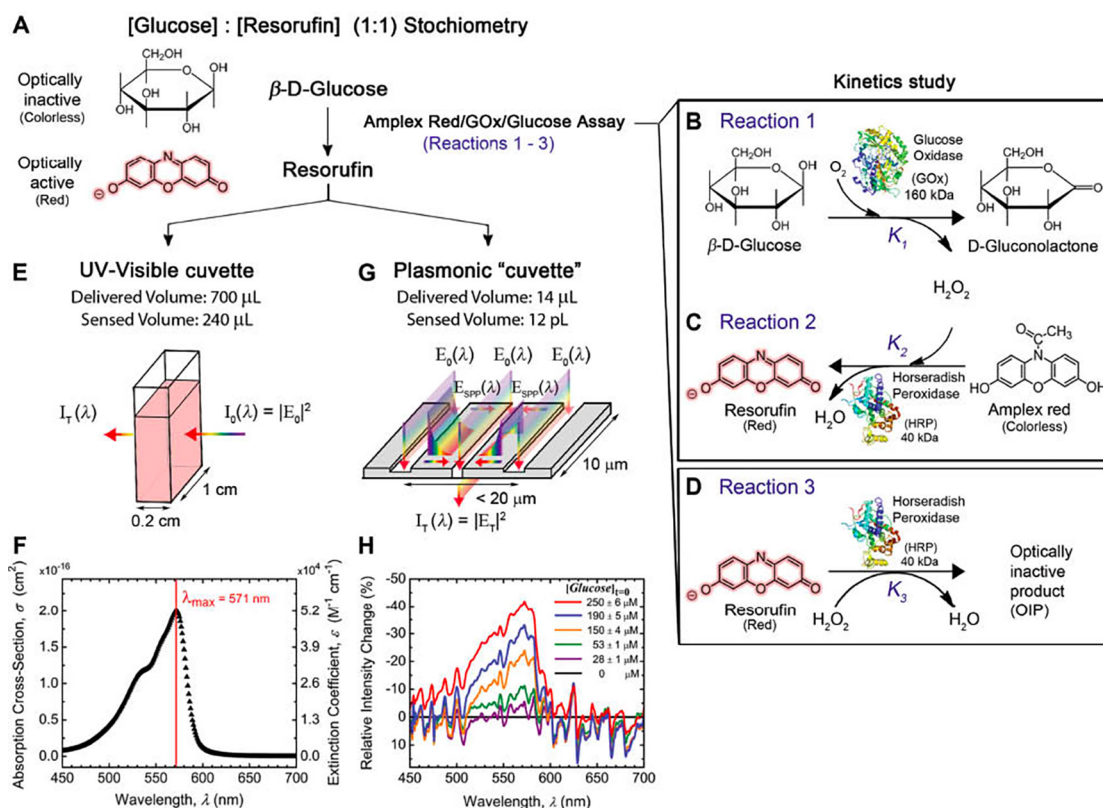


Рисунок 8. Схема побудови біочіпу для аналізу слини [13]

### Література

1. Chaffey D. MobileMarketingStatisticscompilation. — Режим доступа: <http://www.smartinsights.com/> — Дата доступа: 12.03.2016.
2. Govette J. 30 AmazingMobileHealthTechnologyStatisticsforToday'sPhysician. — Режим доступа: <https://getreferralmd.com/>. — Дата доступа: 25.04.2016.
3. Elea C., Kristeen C. Healthline. — Режим доступа: <http://www.healthline.com/> — Дата доступа: 21.02.2016.
4. Cushman W. C. Effectsofintensiveblood-pressurecontrolintype 2 diabetesmellitus. / Cushman W. C., Evans G. W., Byington R. P., Goff D. C., Grimm R. H. // N Engl J Med, квітень 2010. — С. 1575–1585.
5. Funnell M. M. Standardsofcarefordiabetes: what'snew? / Funnell M. M., Brown T. L., Childs B. P. // Nursing. — 2010. — С. 54–56.
6. Chomutare T. FeaturesofMobileDiabetesApplications: ReviewoftheLiteratureandAnalysisofCurrentApplicationsComparedAgainstEvidence-BasedGuidelines/ Chomutare T., Fernandez-Luque L., rsand E., Hartvigsen G. // J MedInternetRes. — 2013 — С. 65.
7. DIABETESNET.COM. DiabetesSoftware. — Режим доступа: <http://www.diabetesnet.com>. — Дата доступа: 24.03.2016.
8. Wicklund E. ResearchKit — ThePerfectPlatformforPopulationHealth? — Режим доступа: <http://mhealthintelligence.com/> — Дата доступа: 3.05.2016.
9. Wikipedia. GoogleContactLens. — Режим доступа: <https://en.wikipedia.org/> — Дата доступа: 12.05.2016.
10. Quick D. Laserdevicedetectsbloodglucoselevelswithoutthefinger-prick. — Режим доступа: <http://www.gizmag.com/> — Дата доступа: 10.04.2016.
11. Quick D. Non-invasivewaytomonitorbloodglucoselevelswithlight. — Режим доступа: <http://www.gizmag.com/> — Дата доступа: 23.03.2016.
12. Wood C. Finalpieceofthediabetespuzzleopensthe doortobetterscreening. — Режим доступа: <http://www.gizmag.com/> — Дата доступа: 12.05.2016.
13. Holloway J. Microsoft developingelectroniccontactlens tomonitorbloodsugar. — Режим доступ: <http://www.gizmag.com/>. — Дата доступа: 20.03.2016.
14. Coxworth B. Temporarytattoocouldletdiabeticsmonitorglucoselevelswithoutjabbingthemselves. — Режим доступа: <http://www.gizmag.com/> — Дата доступа: 12.02.2016.
15. Bandothkar A. J. BiocompatibleEnzymaticRollerPensforDirectWritingofBiocatalyticMaterials: «Do-it-Yourself» ElectrochemicalBiosensors/ Bandothkar A. J., Jia W., Ramirez J., Wang J. // AdvancedHealthcareMaterials. — 2015. — С. 1215–1224.
16. Wikipedia. Поверхневий плазмон. — Режим доступа: <https://uk.wikipedia.org>. — Дата доступа: 23.04.2016.

**Бондаренко Олексій Сергійович**

*студент*

*Національний технічний університет України «Київський політехнічний інститут»*

**Бондаренко Алексей Сергеевич**

*студент*

*Национальный технический университет Украины «Киевский политехнический институт»*

**Bondarenko O.**

*student*

*National Technical University of Ukraine «Kyiv Polytechnic Institute»*

## МОБІЛЬНІ ДОДАТКИ ДЛЯ МЕДИЦИНИ

## МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ ДЛЯ МЕДИЦИНЫ

## MOBILE APPLICATIONS FOR MEDICINE

**Анотація.** Аналіз мобільних додатків мобільної медицини та додатків для пацієнтів з цукровим діабетом.

**Ключові слова:** мобільний додаток, медицина, діабет, аналіз рівня цукру в крові.

**Аннотация.** Анализ мобильных приложения мобильной медицины и приложений для пациентов с сахарным диабетом.

**Ключевые слова:** мобильное приложение, медицина, диабет, анализ уровня сахара в крови.

**Summary.** Analyse of mobile applications of mHealth and applications for patients with diabetes.

**Key words:** mobile application, medicine, diabetes, the analysis of blood sugar levels.

### Вступ

Згідно даних Kleiner Perkins Caufield Byers у 2014 році, кількість мобільних пристроїв перевищила кількість стаціонарних комп'ютерів [1]. Таке зростання спровокувало стрімке збільшення кількості мобільних додатків у всіх сферах.

52% користувачів збирають інформацію про стан свого здоров'я на мобільних пристроях. Це включає в себе інформацію про конкретну медичну проблему, процедури, дієти, харчування або фітнес.

З розповсюдженням мобільних технологій отримала розвиток мобільна медицина. Мобільна медицина — це концепція, що має на меті об'єднати пацієнта і лікаря за допомогою мобільних пристроїв, з метою більш точного та повного обміну даними про стан пацієнта. Це зумовлює поліпшення якості надавання медичних послуг.

З точки зору діабету мобільна медицина може запропонувати декілька можливих варіантів застосування мобільних пристроїв. По-перше, пацієнт може заповнювати електронний щоденник пацієнта, що відображає інформацію про його стан та параметри харчування та фізичної активності. По-друге, за допо-

могою спеціальних пристроїв стає можливим контролювати рівень цукру в крові.

У сфері мобільної медицини смартфони стають джерелом великої кількості даних. Дані можуть збиратися з вбудованих сенсорів та з зовнішніх пристроїв — носимих пристроїв. Багато даних можуть збиратися в автоматичному режимі, що зменшує навантаженість на пацієнта.

За допомогою мобільних пристроїв пацієнти можуть збирати спеціалізовані дані з деяких захворювань протягом тривалого проміжку часу, які раніше було важко отримати. Лікарі також можуть аналізувати поведінку пацієнтів, характер фізичних навантажень. Наприклад, браслети Fitbit та Jawbone's Up24, які є популярними серед користувачів, можуть з легкістю відстежувати фізичні вправи, харчування та навіть цикли сну.

Мобільний пристрій може допомогти контролювати рівень цукру в крові. Це стало можливим завдяки глюкометрам, з вбудованими Bluetooth і audio-jack інтерфейсам. Такі пристрої можуть бути як самостійними, що мають вбудовану пам'ять та джерело живлення, так і спеціальним гаджетом, що працює лише в парі з мобільним телефоном.

Спеціальне мобільний додаток може збирати дані з глюкометрів і проводити аналіз стану користувача, нагадувати про необхідність уколу інсуліну. Своєчасний аналіз може попередити критичний стан пацієнта.

За останні кілька років, спостерігається стрімке зростання кількості мобільних пристроїв на ринку — від планшетів до носимих пристроїв. Медичні співробітники стають мобільними. Згідно зі звітом Research2Guidance, 80% лікарів використовують смартфони і медичних програми.

Носимі пристрої здатні забезпечити безперервний моніторинг стану людини. Вони можуть збирати дані самостійно, без потреби з боку користувача ініціалізації вимірювання. Це відкриває можливості для глибокого аналізу медичних даних і виявлення причин зміни стану людини.

Метою дипломної роботи був аналіз можливостей розробки програмного продукту для пацієнтів з діабетом та розробка програмного забезпечення під мобільні платформи для контролю рівня цукру у крові.

### **Арсенал mHealth: огляд існуючих мобільних додатків та можливості для їх поліпшення**

Всього для мобільних платформ існує близько 100 000 додатків у сфері мобільної медицини. Їх можна поділити на 2 основні категорії: для пацієнтів та для лікарів. З 85% — це додатки для пацієнтів, а 15% — для лікарів. Більшість додатків безкоштовна.

В 2014 році Research2Guidance отримало данні про більше ніж 4 мільйони завантажених безкоштовних мобільних додатків з mHealth кожен день. Ця цифра продовжує збільшуватися.

### **Статистика використання мобільних систем**

Сьогодні на ринку мобільних операційних систем домінують 3 платформи: Android, iOS та Windows Phone. Розподіл ринку між цими мобільними системами 61%, 32% та 2.54% відповідно.

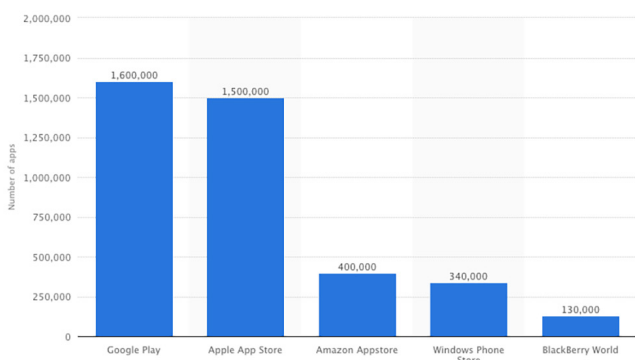


Рисунок 1. Кількість мобільних додатків в магазинах мобільних додатків [1]

Для кожної з платформ існує магазин додатків, а для Android є декілька таких магазинів. Статистика за кількістю додатків трошки відрізняється від статистики розподілу операційних систем: в ній нема такого істотного розриву між iOS та Android.

### **Статистика mHealth**

З моменту запуску iPhone у 2007 році, мобільні технології у сфері охорони здоров'я стали постійно зростаючим трендом серед людей по всьому світу.

52% користувачів збирають інформацію про стан свого здоров'я на мобільних пристроях. Це включає в себе інформацію про конкретну медичну проблему, процедури, дієти, харчування або фітнес. Також популярними є такі теми:

- рецепти або ліки, що продаються без рецепту;
- альтернативні методи лікування;
- медична страховка;
- депресія, неспокій стан або стрес;
- і конкретний лікар або лікарня.

Проте, пошук інформації про стан здоров'я — не єдиний приклад використання мобільних пристроїв. Медичні фахівці використовують мобільні пристрої для надання професійних медичних послуг. Цей ринок за прогнозами, у 2017 році буде коштувати близько \$26 млрд (згідно зі звітом про ринок мобільної охорони здоров'я по Research2Guidance за 2013–2017 рр). [2]

### **Тенденції mHealth**

За останні кілька років, спостерігається стрімке зростання кількості мобільних пристроїв на ринку — від планшетів до носимих пристроїв. Медичні співробітники стають мобільними. Згідно з тим же звітом Research2Guidance, 80% лікарів використовують смартфони і медичних програми.

Якщо раніше для галузі охорони здоров'я було проблемою адоптуватись для використання мобільних пристроїв через проблеми з дотриманням безпеки IT, то тепер все більше організації долають ці труднощі. Оскільки все більше додатків для здоров'я розробляються з дотриманням вимог HIPAA та приватної практики, лікарням та іншим закладам медичної галузі стає простіше інтегрувати мобільні пристрої у всій організації. Постачальники і співробітники тепер використовують мобільні для спілкування, безперервної освіти, передачі записів про пацієнтів.

Найбільший ризик від мобільної медицини — витік інформації. Тому важливо реалізувати правильну архітектуру IT-системи, аби мінімізувати можливі збитки від більшості ризиків. Наприклад, втрата або крадіжка мобільного пристрою.

За статистикою 91% дорослого населення мають при собі мобільні пристрої в режимі 24/7. Це 24

години на добу, сім днів на тиждень, а це означає, що мобільні пристрої знаходяться поруч з користувачами, навіть коли вони сплять. Таке використання не є рідкістю в медичній галузі, оскільки характер роботи часто вимагає, щоб лікарі були доступні в будь-який час дня і ночі. У той же час, лікарі повинні мати доступ до джерел інформації в різний час.

### **Розвиток ринку мобільної медицини**

Сьогодні 61% користувачів мобільних пристроїв хоча б раз встановлювали додаток з mHealth. Це приголомшлива цифра. Люди хочуть бути в курсі про стан свого здоров'я, а медичні заклади хочуть мати зв'язок з пацієнтом, коли їм це потрібно.

Згідно з результатами дослідження, загальна вартість додатків mHealth в 2010 році склала лише \$85 млн. З тих пір вона виросла до приблизно \$489 млн у 2015 році. Сама по собі цифра у \$489 млн не є приголомшливою для ринку мобільної медицини в цілому, але вона показує тенденцію к стрімкому зростанню, зважаючи на ненасиченість ринку.

Крім того, це зростання відбувається швидше, ніж у багатьох інших категорій додатків. Згідно з результатами дослідження, загальне зростання додатків склало 38,1% та посилюється щорічно протягом останніх п'яти років, в той час як для медичних додатків — 41,9%.

### **Приклад впровадження mHealth**

Госпіталь Nahnemann у Філадельфії представило тестовий мобільний додаток, що реалізував функції служби підтримки та текстового чату з 350 пацієнтами з застійною серцевою недостатністю. В ході цього експерименту пацієнти отримували по електронній пошті та через текстові повідомлення нагадування про необхідність наступного візиту до лікарні.

В результаті лікарні вдалось скоротити 30-ти денний термін повторної госпіталізації на 10%. В кінці 10 місяців пілотного проекту було зроблено висновки, що мобільний додаток може не тільки допомогти пацієнтам дотримуватися графіку лікування, а ще й зекономити кошти лікарні.

### **Як лікарі використовують мобільні пристрої**

Ось деякі статистичні дані про джерела інформації, які лікарі шукають за допомогою мобільних пристроїв:

- 72% лікарів отримують інформацію про ліки з смартфонів;
- 63% лікарів отримати доступ до медичних досліджень з планшетів;
- 44% лікарів спілкуватися з медсестрами та іншими співробітниками через смартфони.

Окрім швидкого пошуку інформації мобільні пристрої використовуються для комунікації лікар-пацієнт та збору медичних даних. Зазвичай спілкування відбувається через спеціальний додаток — медичний портал. За даними дослідження 2015 HIMSS Mobile Technology Survey з 200 опитаних респондентів 90% використовують мобільні технології для залучення пацієнтів до процесу надання медичної допомоги.

За допомогою мобільних пристроїв пацієнти можуть збирати спеціалізовані дані з деяких захворювань протягом тривалого проміжку часу, які раніше було важко отримати. Лікарі також можуть аналізувати поведінку пацієнтів, характер фізичних навантажень. Наприклад, браслети Fitbit та Jawbone's Up24, які є популярними серед користувачів, можуть з легкістю відстежувати фізичні вправи, харчування та навіть цикли сну. Інші пристрої, такі як глюкометри від компанії iHealth, дозволяють пацієнтам вимірювати кількості цукру у крові та ділитися цими даними зі своїм лікарем.

### **Огляд мобільних додатків**

За останні десять років життя людей з діабетом значно поліпшилось. Збільшилась кількість інформації про хворобу та методи ранньої діагностики [3]. Можливості для домашнього моніторингу хвороби започаткували нове покоління пацієнтів.

Не дивлячись на наявність великої кількості мобільних додатків, розроблених для Apple's iPhone, Google's Android, BlackBerry і Nokia Symbian і доступних сьогодні людям з цукровим діабетом (таких додатків близько 200 найменувань!), існують певні невідповідності між науково обґрунтованими клінічними рекомендаціями до основними функціями і параметрами таких додатків [4] і реальними функціональними можливостями наявних мобільних додатків, які можна знайти в інтернет-магазинах [5].

Фахівці з лікування діабету рекомендують використовувати мобільні засоби самоконтролю пацієнта за наступними важливими показниками:

- «персоналізація» медичної інформації, що поставляється пацієнтові, та зворотним зв'язком з лікарем;
- моніторинг стану пацієнтів і негайна допомога в екстремальних випадках;
- моніторинг уживаного інсуліну і ходу медикаментозного лікування;
- соціальна та психологічна допомога;
- контроль застосовується дієти;
- контроль ваги пацієнта;
- контроль фізичної активності;
- інші терапевтичні заходи лікування (догляд за ногами, догляд за очима);

- імунізація;
- попередження ускладнень хвороби [6].

На сьогодні в світі немає мобільних додатків, що підтримують всі перераховані вище рекомендації медиків [7]. Наявні програми можна умовно розділити на три класи в залежності від їх функціональності [8].

Слід зазначити, що розробка мобільних додатків для діабетиків активно триває. Наприклад, Національний науковий фонд США виділив 1,2 млн доларів в 2012 році на розробку нового мобільного додатка, призначеного для пацієнтів з діабетом. Створенням додатка займуться Політехнічний інститут в Вустере (штат Массачусетс) і Медична школа, діюча в складі Університету Массачусетсу. Новий додаток намічено створити за чотири роки. Два перших з них підуть на розробку

прототипу, а решту часу — на тестування додатка в Медичній школі при Університеті Массачусетса.

### **Висновки**

Було проведено аналіз основних тенденцій у mHealth та розглянуто мобільні додатки для пацієнтів з діабетом. Для цього були використані дані статистики використання додатків лікарями та пацієнтами. Також були розглянуті приклади впровадження mHealth в лікарнях світу.

У сфері мобільної медицини смартфони стають джерелом великої кількості даних. Дані можуть збиратися з вбудованих сенсорів та з зовнішніх пристроїв. Багато даних можуть збиратися в автоматичному режимі, що зменшує навантаження на пацієнта.

### **Література**

1. Chaffey D. Mobile Marketing Statistics compilation. — Режим доступу: <http://www.smartinsights.com/> — Дата доступу: 12.03.2016.
2. Govette J. 30 Amazing Mobile Health Technology Statistics for Today's Physician. — Режим доступу: <https://getreferralmd.com/>. — Дата доступу: 25.04.2016.
3. Elea C., Kristeen C. Healthline. — Режим доступу: <http://www.healthline.com/> — Дата доступу: 21.02.2016.
4. Cushman W. C. Effects of intensive blood-pressure control in type 2 diabetes mellitus. / Cushman W. C., Evans G. W., Byington R. P., Goff D. C., Grimm R. H. // *N Engl J Med*, квітень 2010 — С. 1575–1585.
5. Funnell M. M. Standards of care for diabetes: what's new? / Funnell M. M., Brown T. L., Childs B. P. // *Nursing*. — 2010 — С. 54–56.
6. Chomutare T. Features of Mobile Diabetes Applications: Review of the Literature and Analysis of Current Applications Compared Against Evidence-Based Guidelines / Chomutare T., Fernandez-Luque L., Årsand E., Hartvigsen G. // *J Med Internet Res*. — 2013 — С. 65.
7. DIABETESNET.COM. Diabetes Software. — Режим доступу: <http://www.diabetesnet.com>. — Дата доступу: 24.03.2016.
8. Wicklund E. ResearchKit — The Perfect Platform for Population Health? — Режим доступу: <http://mhealthintelligence.com/> — Дата доступу: 3.05.2016.

**Сергеев Егор Игоревич**

*студент,*

*Національний технічний університет України «Київський політехнічний інститут»*

**Serheiev Y.**

*student,*

*National Technical University of Ukraine «Kyiv Polytechnic Institute»*

**Serheiev Y.**

*student*

*National Technical University of Ukraine «Kyiv Polytechnic Institute»*

**РОЗРОБКА КРОС-ПЛАТФОРМНИХ ДОДАТКІВ  
З ВИКОРИСТАННЯМ ФРЕЙМВОРКА XAMARIN  
РАЗРАБОТКА КРОССПЛАТФОРМНЫХ ПРИЛОЖЕНИЙ  
С ИСПОЛЬЗОВАНИЕМ ФРЕЙМВОРКА XAMARIN  
DEVELOPING CROSS-PLATFORM SOFTWARE  
USING XAMARIN FRAMEWORK**

**Анотація.** Досліджено особливості та підходи розробки крос-платформних додатків з використанням фреймворка Xamarin.

**Ключові слова:** крос-платформність, мобільний, додаток, iOS, Android.

**Аннотация.** Исследованы особенности и подходы разработки кроссплатформных приложений с использованием фреймворка Xamarin.

**Ключевые слова:** кроссплатформность, мобильный, приложение, iOS, Android.

**Summary.** Were investigated features and approaches of developing cross-platform mobile applications.

**Keywords:** cross-platform, mobile, application, iOS, Android.

**Вступ**

Коли постає питання, яким чином будувати iOS чи Android додатки, багато людей, в першу чергу, згадують такі мови програмування як Swift, Java, або Objective-C. Проте в цій роботі буде розглянуто зовсім іншу технологію.

Xamarin – це унікальний фреймворк який дозволяє будувати такі додатки з використанням однієї мови програмування – C#. З використанням данного фреймворка ми компілюємо (не інтерпретуємо, як, наприклад, при використанні аналогічних JavaScript фреймворків) так званий «нативний» код для цих операційних систем (ОС).

Кожна з цих ОС має свій власний набір можливостей і кожна різниться в здатності писати нативні додатки котрі компілюються в машинний код. Наприклад, ОС Android дозволяє писати код на Java, WindowsPhone (WP) – з використанням JavaScript та C#, iOS – Objective-C та Swift. Звичайно є й інші

фреймворки, які дозволяють писати додатки на не нативних для платформи мовах програмування, наприклад CoronaSDK, PhoneGap та інші, проте при використанні таких фреймворків швидкість роботи додатку значно знижується, ніж в тих додатках, що використовують нативні засоби розробки.

**Основні підходи розробки  
з використанням Xamarin**

З найпершого релізу фреймворка Xamarin в ньому було доступно такі два блоки – Xamarin.iOS та Xamarin.Android – ці частини фреймворка дозволяють розробляти додатки під відповідні платформи з використанням єдиної мови програмування, що, звичайно сильно полегшує процес розробки програмного продукту, та дозволяє використовувати спільну бізнес логіку додатку та, наприклад, логіку доступу до SQLite бази даних, що встановлено на мобільному пристрої чи планшеті. Також використання єдиної

логіки додатку веде за собою скорочення часу на розробку, що є гарною перевагою в порівнянні з використанням різних мов програмування, проте є й недоліки, основним з яких є складність для розробника. Розробнику потрібно гарно знати платформи Android та iOS щоб писати дійсно складні та функціональні додатки, тобто потрібно витратити дуже багато часу на те, щоб почати створювати конкурентоспроможні програмні продукти. Xamarin.Android та Xamarin.iOS є частиною підходу розробки, що називається XamarinNative, тобто написання чистого нативного додатку з використанням C#.

Однак, з плином часу та вдосконаленням інфраструктури XamarinNative розробниками з компанії Xamarin було вирішено створити ще одну частину свого фреймворка — Xamarin.Forms. Даний підхід є набагато простішим для розробника, оскільки він полегшує роботу з різними операційними системами. Звісно, розробнику і досі потрібно писати певний платформо-залежний код, проте з використанням даного підходу даного коду стане набагато менше, оскільки в Xamarin.Forms використовується XAML (Extensible application markup language) для побудови графічного інтерфейсу, що дозволяє винести увесь код пов'язаний з графічним дизайном у спільну бібліотеку, тобто розробник тепер не повинен знати специфіку інтерфейсу для кожної платформи, достатньо лише знати XAML та пов'язані з ним класи, даний підхід, тобто отримуємо зв'язку C# та XAML, що схожа на JavaScript та CSS.

Xamarin.Forms повністю «покриває» усі три платформи та дозволяє розробнику зосередитися на бізнес логіці додатку та його структурі і економити час розробки. Однак, не усе так гарно як могло б здаватися, звичайно, такий додаток буде крос-платформним, однак даний додаток матиме багато обмежень з точки зору функціональної частини додатку, оскільки неможливо об'єднати усі можливості платформ і використовувати як одну спільну, адже вони реалізовані по-різному та й деякі з них не наявні у інших платформах. Оскільки тема Xamarin.Forms досить обширна, розглянемо основні недоліки та переваги цієї технології:

1. Дуже схожа на стандартні Windows технології;
2. Спільна логіка для всіх додатків;
3. Не потрібно вглиблюватись в можливості конкретної платформи;
4. Нижча швидкодія, порівняно з Xamarin.iOS, Xamarin.Android;
5. Неповна реалізація усіх можливостей платформ;
6. Компромісні рішення для реалізації функціоналу, що відрізняється на різних платформах;
7. Можлива різна поведінка додатку для платформ.

Тобто, точних рекомендацій для розробника, який з даних підходів використовувати — XamarinNative чи Xamarin.Forms немає, усе залежить від бажання та можливостей розробника, масштабів проекту. Є лише декілька критеріїв селекції одного з цих підходів — час розробки та можливості додатку. Якщо програмний продукт повинен мати широкий функціонал та використовувати усі можливості програмного інтерфейсу (api) платформи, то гарним вибором, звичайно буде Native підхід, якщо ж основним критерієм є час розробки, то єдиним логічним вибором є Xamarin.Forms.

### **Технологія Xamarin.Android**

Xamarin.Android додатки виконуються в CBMono. Mono працює пліч-о-пліч з віртуальною машиною AndroidRuntime (ART). Обидва СВ працюють над ядром Linux і надають різні інтерфейси API для коду програміста, що дозволяє розробникам отримати доступ до даної системи.

Розробникам надано простори імен System, System.IO, System.Net та інші бібліотеки класів .NET, щоб отримати доступ до основних можливостей операційної системи Linux [2].

На Android, більшість системних можливостей, таких як аудіо, графіки, OpenGL і телефонії не доступні безпосередньо до нативних додатків, вони доступні тільки через AndroidJavaRuntimeAPI, що були реалізовані в просторі імен Java.\* або Android.\* [2].

В самому центрі роботи Xamarin.Android розташоване ядро Linux, над яким знаходяться два середовища виконання — MonoART, що взаємодіють між собою за допомогою AndroidCallableWrappers (ACW). Вже над даними середовищами виконання розташовані .NETAPI та спеціальні прив'язки до Android.\* та Java.\* можливостей, що взаємодіють з використанням ManagedCallableWrappers (MCW) [2].

AndroidCallableWrappers — спеціальний Java-Native-Interface (JNI) «міст», що використовується ОС Android для виконання коду.

ManagerCallableWrappers — спеціальні обгортки типу JNI для того, щоб виконувати Android код, що був створений з використанням ACW. Кожна така обгортка зберігає у собі глобальне Java посилання (reference) на об'єкт. Кількість таких об'єктів є лімітованою. Загалом доступно 52.000 таких посилань під час роботи додатку.

Так як Xamarin додатки компілюються в нативні додатки, то програмістам доступні усі контроли (controls), що використовуються при написанні додатків з використанням Java — PopupMenus, Views, DatePickers і т.д.

Також такі додатки можуть використовувати усі можливості APIAndroid—AndroidBeam, камера,



аутентифікація з використанням відбитку пальця, карти, локація, Android.Speech та інші.

Кожен додаток Android працює з використанням спеціальних об'єктів, названих activity. Наприклад, кожна сторінка додатку на Android є activity, а кожне таке activity має свій цикл існування (lifecycle) [1]. Xamarin.Android також підтримує цю концепцію, дозволяючи описувати Activity з використанням C#. Усі activity в Xamarin.Android обрамляються спеціальними атрибутами, що дозволяють додавати дані activity до спеціального Android манифеста, де описуються та додаються основні компоненти додатку. Також Xamarin.Android додатки надають такі можливості:

1. Створення спеціальних провайдерів контенту та резолверів контенту, що надають можливість отримувати та надавати дані, а також шукати дані з для інших додатків у системі;

2. Створення сервісів, що мають свій життєвий цикл та дозволяють виконання деяких складних задач в іншому потоці, наприклад, скачати статтю з певного сайту. Дані сервіси бувають трьох типів, а також можуть використовуватися іншими додатками у системі, створювати повідомлення для користувача та інше;

3. Використання фрагментів замість декількох activity для створення більш швидкого додатку, оскільки не потрібно підгружати різні activity у відповідь користувачу;

4. Xamarin.Android підтримує усі API рівні ОС Android, отже розробникам завжди доступні усі можливості даної платформи;

5. Як і в звичайному додатку під Android, Xamarin.Android на дає змогу створювати ресурси – зображення, елементи інтерфейсу, локалізація, аудіо та відео, файлова система та інше.

Отже, Xamarin.Android наділена багатим функціоналом для розробки додатків під ОС Android, що не гірші від Java аналогів, дозволяє використовувати усе, що надає платформа, а додатки по швидкодії не програють аналогам.

### Технологія Xamarin.iOS

Технологія Xamarin.iOS відрізняється від Xamarin.Android, в першу чергу це пов'язано з кардинальними відмінностями даних платформ. На відміну від Android, iOS не дозволяє використовувати середовища виконання, тому використовувати Mono не вийде. Xamarin.iOS не використовує Mono для додатку. Для iOS використовуються спеціальні прив'язки, для нативних компонентів та можливостей даної ОС. Однак, неможливість використання середовища виконання – не проблема, оскільки на відміну від використання JIT(Just-In-Time) компіляції, додаток, що написаний на C# під iOS компілюється одразу в байт

код і для даного додатку уже не потрібно використання будь-якого середовища виконання.

Як і для Xamarin.Android, Xamarin.iOS дозволяє використання усіх нативних контролів та можливостей платформи та пристрою, на якому виконується програма.

Основними робочими блоками в Xamarin.iOS є event, delegate та protocol. Event викликається під час взаємодії користувача з пристроєм, такі event можна зв'язати із кнопкою чи іншим елементом управління (UIKit) для створення певної поведінки додатку та реагування на дії користувача, ви можете прив'язувати до елементів багато подій (events) та присвоювати їм різну поведінку. Protocol – щось на зразок інтерфейсу (interface) в C#, що дозволяє «дізнатися» ОС який метод викликати, не знаючи його поведінку, а delegate використовуються як відповідь (callback) на якусь дію.

### Технологія Xamarin.Forms

Xamarin.Forms – це зовсім несхожа технологія на дві інші, що були описані раніше. Дана технологія більше схожа на JavaScript фреймворки, які було описано в першому розділі. Різниця в тому, що Xamarin.Forms дозволяє писати нативні додатки з більшою кількістю спільного коду. Наприклад, інтерфейс програми буде спільний для усіх додатків(по структурі та по написанню, проте матиме дещо різний вигляд на платформах) оскільки для побудови інтерфейсу користувача використовується спеціальна мова розмітки – XAML. Тобто розробнику не потрібно знати особливості кожної платформи, розробники із Xamarin все зробили за вас [3].

Xamarin.Forms надає розробнику абстракції графічного інтерфейсу, що використовуються для кожної платформи. Дана технологія використовує нативні компоненти під час роботи додатку, тобто код графічного інтерфейсу на різних платформах має різне підґрунтя, а саме його буде перетворено на нативний код окремої конкретної платформи. Код Xamarin.Forms має можливість взаємодіяти з операційною системою, використовувати її програмний інтерфейс.

Основними типами компонентами, що використовуються в розробці з використанням Xamarin.Forms:

1. Представлення

Представлення – це основний блок графічного інтерфейсу, бо саме з цих елементів його побудовано. Прикладом представлення є кнопка, поле вводу і т.д.

2. Макет

Макети визначають, як представлення будуть розташовані на екрані, їх розмір при різних розмірах екрану, поведінку представлень в певних ситуаціях;

3. Сторінка

Сторінка має дві основні функції. Перша — сторінка представляє собою контейнер для представлень та макетів, тобто сторінка і є екраном телефону, також сторінка має і іншу функцію — навігаційну. Саме сторінка дозволяє перехід від однієї сторінки до іншої.

Основними блоками, що дозволяють писати логіку додатку є клас `Device`, центр обміну повідомленнями (`messagecenter`) та сервіс залежностей (`dependencyservice`). Клас `Device` надає методи для використання платформи-специфічних можливостей додатку та побудови поведінки додатку з точки зору ОС. Центр обміну повідомленнями надає можливість використовувати методи `publish/subscribe` — тобто підписуватися на якусь подію, що сталася в ОС, чи, наприклад, зіставити метод, що буде спрацьовувати при натисканні кнопки [3].

### Висновки

Було досліджено можливості фреймворка Xamarin, розглянуто основні підходи розробки додатків з використанням Xamarin. Можна зробити висновок, що Xamarin — перспективний фреймворк, використання якого дає багато переваг в циклі створення мобільних додатків, скорочуючи розробку та зменшуючи кількість коду за рахунок спільної бізнес логіки та методів доступу до даних, а з використанням `Xamarin.Forms` дає й можливість розробляти спільний інтерфейс користувача для додатків. Хоча підхід, що використовується в Xamarin має й свої недоліки, основним з яких є те, що програмісту потрібно мати гарні знання одразу трьох різних ОС, що мають мало спільного, тобто розробник повинен мати високу кваліфікацію, хоча цей недолік зникає з роками досвіду.

### Література

1. Jonathan Peppers. Xamarin Cross-platform Application Development Second Edition / Peppers J. — Birmingham: Packt, 2015. — 459 с.
2. Nilanchala Panigrahy. Xamarin Mobile Application Development for Android Second Edition / Panigrahy N. — Birmingham: Packt, 2015. — 296 с.
3. Charles Petzold. Creating Mobile Apps with Xamarin.Forms First Edition / Petzold C. — Redmond: Microsoft Press, 2016. — 1187 с.

**Піпич Артем Андрійович**

*бакалавр комп'ютерних наук,*

*Національний технічний університет України «Київський політехнічний інститут»*

**Пипич Артём Андреевич**

*бакалавр компьютерных наук,*

*Национальный технический университет Украины «Киевский политехнический институт»*

**Pipich A.**

*Bachelor of computer science*

*The National Technical University of Ukraine «Kyiv Polytechnic Institute»*

## КОНФЛИКТИ В КОНЦЕПТУАЛЬНИХ ПРОШАРКАХ КОНТЕКСТНО-ЗАЛЕЖНИХ СИСТЕМ КОНФЛИКТЫ В КОНЦЕПТУАЛЬНЫХ СЛОЯХ КОНТЕКСТНО-ЗАВИСИМЫХ СИСТЕМ CONFLICTS IN CONTEXT-AWARE SYSTEMS CONCEPTUAL LAYERS

**Анотація.** Розглянуто класифікацію конфліктів в контекстно-залежній системі на основі її концептуальних меж прошарків та особливості їх усунення.

**Ключові слова:** контекстно-залежні системи, конфлікти, класифікація конфліктів, концептуальні межі прошарків, усунення конфліктів.

**Аннотация.** Рассмотрена классификация конфликтов в контекстно-зависимой системе на основе ее концептуальных границ слоев и особенности их устранения.

**Ключевые слова:** контекстно-зависимые системы, конфликты, классификация конфликтов, концептуальные границы слоев, устранение конфликтов.

**Abstract.** The classification of conflicts in the context-aware system based on its conceptual boundaries layers and features of its resolving.

**Keywords:** context-aware systems, conflict, conflict classification, conceptual boundaries layers, conflict resolution.

Контекстно-залежні системи в повсюдних середовищах стикаються з конфліктними ситуаціями під час збору даних датчиків, їх обробки для вилучення послідовної та узгодженої інформації про контекст високого рівня, і поширення цієї інформації контексту для надання допомоги в прийнятті рішень з адаптації до постійно мінливих ситуацій, не відволікаючи людську увагу. Ці конфліктні ситуації створюють серйозні проблеми для проектування і розробки контекстно-залежних систем. Для їх усунення може бути застосовано підхід Quality of Context. Надалі розглянуто конфліктні ситуації, з якими контекстно-залежні системи можуть зіткнутися на різних рівнях концептуального проектування і політики розв'язання конфліктів, що базуються на підході Quality of Context, а також показано, як ці політики можуть бути використані в різних

конфліктних ситуаціях для підвищення продуктивності і ефективності контекстно-залежних систем.

Повсюдні середовища характеризується великою кількістю обчислювально-комунікаційних сенсорних пристроїв, вбудованих в об'єкти повсякденного користування, в зв'язку з чим в різних типах зв'язків між пристроями виникають різноманітні конфліктні ситуації, які сильно впливають на здатність контекстно-залежних систем адаптуватися до ситуації в повсюдних середовищах.

В результаті аналізу загальних конфліктних ситуацій, які можуть виникнути на різних прошарках контекстно-залежних систем можуть бути отримані політики розв'язання конфліктів, засновані на якості параметрів контексту. Також такий аналіз надає інформацію про те, яким чином ці політики можуть бути використані; опис прототипу реалізації систе-

ми, до якої застосовано політику. Отримані політики можуть бути досліджені експериментально. Зазвичай, ті політики, які використовують комбінацію різних параметрів QoS, є більш ефективними відповідно до перспективи використання контекстної інформації.

Конфлікти можуть мати місце на різних прошарках CMS, і в залежності від цього по різному впливати на продуктивність контекстно-залежних додатків.

### **Здобуття контексту**

У повсюдних середовищах обсяг даних, отриманих за допомогою датчиків робить аналіз контексту неможливим для людини [6, с. 105–108]. Дані датчиків також можуть відрізнятися один від одного, з огляду на частоту оновлення контексту, здатність датчика збирати контекст певної сутності, точність методу, що використовується датчиками, формату представлення, а також цінність контекстної інформації [5, с. 929–945; 6, с. 621–634]. Наприклад, інформація про місцезнаходження користувача мобільного пристрою може бути зібрана за допомогою GPS і GSM методів. Постійна оцінка датчика для збору контексту певної сутності не є можливою. Таким чином, існує необхідність в стратегії, яка може динамічно вирішити, який датчик є більш надійним, щоб зібрати контекст певної сутності в певний момент часу. Параметри QoS, які були динамічно оцінені з інформації про джерело контексту можуть бути використані для вирішення конфліктів в такій ситуації.

### **Обробка**

На стадії обробки, контекст високого рівня витягується з даних датчика низького рівня. Дані датчиків не можуть бути представлені безпосередньо в додатку. Вони повинні бути відфільтровані, зібрані в цілісну структуру, корельовані, перекладені для вилучення даних контексту вищого рівня і виявлення зареєстрованих подій [3, с. 75–79].

Ці показники використовуються, щоб прийняти рішення щодо добування інформації про контекст високого рівня. Так, заявлена достовірність джерела

контексту може бути використана для прийняття рішення про витягання певної блоку інформації шляхом комбінування даних з різних джерел.

QoS параметри, які надають інформацію про своєчасність, надійність, повноту і значущість, можуть замінити ці метрики і зробити прийняття рішень з обробки даних більш змістовними та реалістичними для усунення конфліктів.

### **Розподіл контексту**

Рухомість датчиків, ненадійні бездротові з'єднання, а також характер завдань в повсюдних середовищах призводить до накопичення великих обсягів надлишкового і суперечливого контексту, що не тільки призводить до витрат обмежених ресурсів, а й може призвести до небажаного поведіння контекстно-залежних додатків. Прості політики усунення конфліктів (drop first, drop all) можуть привести до втрати цінної інформації. У критичній ситуації, наприклад, функціонування контекстно-залежного додатку медицини на дому [2, с. 576–581] і телемедицини [1, с. 69], втрата інформації може призвести до серйозних наслідків. Рішення про відміну чи збереження сутності контексту доцільніше приймати відповідно до політик, визначених QoS параметрами.

### **Додатки**

Контекстно-залежні додатки використовують контекстну інформацію, щоб адаптувати свою поведінку до потреб користувачів і змін в середовищі. Якщо конфлікти не будуть вирішені в контекстній інформації на більш ранніх стадіях, то вони виникнуть і в додатках, які приймають рішення відповідно до контекстної інформації. Контекстно-залежні додатки можуть також бути втягнуті в конфлікт відповідно до різних пріоритетів, встановлених користувачами. Для врегулювання роботи таких додатків використовуються різні стратегії [4, с. 359–364]. Інформація про своєчасність, надійність, повноту і значущість контекстної інформації дозволяє легко вирішувати конфлікти і приймати рішення на її основі.

### **Література**

1. Nadjia Kara Reasoning with contextual data in telehealth applications / Nadjia Kara, O. Andrei Dragoi // In Proceedings of the Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications. — 2007. — № 3. — С. 69.
2. Younghee Kim A quality measurement method of context information in ubiquitous environments / Younghee Kim, Keum-suk Lee // Proceedings of the 2006 International Conference on Hybrid Information Technology. — 2006. — № 6. — С. 576–581
3. Alex Wun A system for semantic data fusion in sensor networks / Alex Wun, Milenko Petrovi, Hans-Arno Jacobsen // Proceedings of the 2007 inaugural international conference on Distributed event-based systems. — 2007. — № 7. — С. 75–79f.
4. Insuk Park A dynamic context-conflict management scheme for group-aware ubiquitous computing environments / Insuk Park, Dongman Lee, Soon J. Hyun // Proceedings of the 29th Annual International Computer Software and Applications Conference. — 2005. — № 1. — С. 359–364.
5. Licia Capra Carisma: Context-aware reflective middleware system for mobile applications / Licia Capra, Wolfgang Emmerich, Cecilia Mascolo // IEEE Transactions on Software Engineering. — 2003. — № 29. — С. 929–945.
6. Diane J. Cook Making sense of sensor data / Diane J. Cook // IEEE Pervasive Computing. — 2007. — № 6. — С. 105–108.

**МІЖНАРОДНИЙ НАУКОВИЙ ЖУРНАЛ**  
**INTERNATIONAL SCIENTIFIC JOURNAL**  
**МЕЖДУНАРОДНЫЙ НАУЧНЫЙ ЖУРНАЛ**

Сборник научных статей

Выпуск 6

2 том

Глава редакционной коллегии — д.э.н., профессор *Тарасенко И.А.*

Киев 2016

**Издано в авторской редакции**

---

Адрес: Украина, г. Киев, ул. Павловская, 22, оф. 12  
Контактный телефон: +38(067) 401-8435  
E-mail: editor@inter-nauka.com  
www.inter-nauka.com

Подписано в печать 21.07.2016. Формат 60×84/8  
Бумага офсетная. Гарнитура PetersburgC.  
Условно-печатных листов 8,84. Тираж 100. Заказ № 397.  
Цена договорная. Напечатано с готового оригинал-макета.

Напечатано в ООО «Спринт-Сервис»  
г. Киев, ул. Почайнинская, 28б  
Свидетельство: Серия ДК №4365 от 17.07.2012  
Контактный телефон: +38(050) 647-1543