

РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ФОРМУВАННЯ ВІДРІЗКІВ ПРЯМИХ НА ГЕКСАГОНАЛЬНОМУ РАСТРІ

Вінницький національний технічний університет

Анотація

Розглянуто особливості формування відрізків прямих на гексагональному растрі. Для реалізації алгоритму використано метод оцінювальної функції. Розроблено програму для формування векторів. Наведено приклади формування відрізків прямих.

Ключові слова: гексагональний растр, формування вектора, формування відрізка, метод оцінювальної функції, злагоджена траєкторія.

Abstract

The peculiarities of the formation of line segments on a hexagonal raster are considered. To implement the algorithm, the method of the estimating function was used. A program for the formation of vectors has been developed. Examples of formation of line segments are given.

Keywords: hexagonal raster, vector formation, segment formation, estimation function method, coordinated trajectory.

Вступ

Сьогодні для підвищення реалістичності формування графічних зображень широко використовують гексагональний растр [1], який за забезпечує високу роздільну здатність.

Відрізки прямих є одними з найпоширенішими графічними примітивами. Тому доцільно розробити програмні засоби для реалізації лінійної інтерполяції на гексагональному растрі.

Результати дослідження

Серед методів формування відрізків прямих найбільшого поширення знайшли метод “прямий” метод, метод цифрового диференційного аналізатора, метод оцінювальної функції [2].

Згідно “прямого” методу координати точок траєкторії знаходять з рівняння прямої $Y = Y_n + \Delta Y * \frac{X}{\Delta X}$. При $\Delta X \geq \Delta Y$ у рівняння підставляють поточну координату X і знаходять координату Y, а при $\Delta X < \Delta Y$ – навпаки. Похибка інтерполяції при реалізації методу визначається обраним методом округлення.

Необхідність виконання “довгих” операцій, що визначає швидкодію формування векторів, суттєво обмежує застосування методу.

При використанні методу ЦДА відношення $P = \Delta Y / \Delta X$ знаходять в циклі підготовки, а операцію множення на X заміняють накопичуючим додаванням.

На практиці, в переважній більшості випадків, для лінійної інтерполяції використовують метод оцінювальної функції. Це пов'язано з високою точністю інтерполяції, простотою програмної та апаратної реалізації і відносно високою швидкістю, обумовленою часом виконання мікрооперації підсумовування.

Загальним прийомом для всіх інтерполяторів, що працюють за методом оцінювальної функції, є аналіз знаків оцінювальної функції, на основі якого роблять крок за тією чи іншою координатою з подальшим розрахунком її нового значення і корекцією відповідної координати поточної точки [3].

Якщо точка траєкторії знаходиться вище від прямої, то оцінювальна функція OF більша нуля і наступний крок необхідно виконувати по осі X; якщо точка знаходиться нижче від цієї прямої, то $OF < 0$

0 і наступний крок необхідно виконувати по осі Y. При використанні восьми векторної направленості крокових пристроїв в останньому випадку формують діагональний крок.

Згадані методи, які застосовуються для прямокутного растра, можуть бути модифіковані та адаптовані для застосування у гексагональному растрі. Оскільки метод оцінювальної функції отримав найбільше поширення, то доцільно модифікувати цей метод для формування відрізків прямої на гексагональному растрі.

При інтерполяції відрізка прямої, яка задана приростами Δ по осях координат X, Y, загальна формула буде мати такий вигляд:

$$OF_i = \frac{y_i}{x_i} - \frac{\Delta y}{\Delta x} = \frac{y_i + \Delta x - x_i + \Delta y}{x_i + \Delta x} [4].$$

Щоб визначити, по якій з осей OX чи OY робити наступний крок інтерполяції, знаходять знак OF_i . Покрокове формування відрізка буде відбуватися таким чином, як це зображено на рис 1.

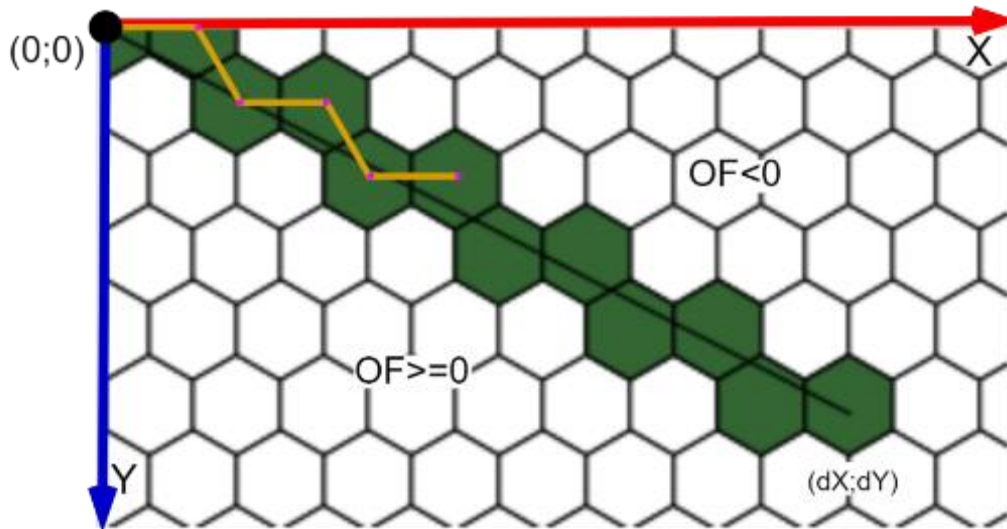


Рис. 1. Покрокове формування відрізка прямої

Якщо точка траєкторії знаходиться вище відрізка прямої b , то оцінювальна функція $OF_i \geq 0$, і наступний крок треба робити по осі OX. Якщо точка траєкторії при інтерполяції знаходиться нижче прямої b , то $OF_i \leq 0$ і наступний крок треба робити по обох осях, тобто виконується діагональний крок.

Оскільки в формулі $OF_i = \frac{y_i}{x_i} - \frac{\Delta y}{\Delta x} = \frac{y_i + \Delta x - x_i + \Delta y}{x_i + \Delta x}$ знаменник дробу $x_i * \Delta x$ не впливає на знак оцінювальної функції, то можливо ввести формулу $OF_i = y_i * \Delta x - x_i * \Delta y$ для пришвидшення обчислення.

Таким чином необхідно знайти нове значення оцінювальної функції при виконанні кроку по осі OX. При формування крокової траєкторії значення координат по OY не зміниться, а значення координат по OX зросте на 1, тобто, $x_{i+1} = x_i + 1$.

Для пришвидшення виконання даного обчислення можливо ввести таку формулу: $OF_{i+1} = y_i * \Delta x - (x_i + 1) * \Delta y = y_i * \Delta x - x_i * \Delta y - \Delta y = OF_i - \Delta y$.

Аналогічно обрахується нове значення оцінювальної функції при виконанні кроку по обох осях OX і OY. Значення координат по осі OX збільшиться на $\frac{1}{2}$, а по осі OY на $\frac{\sqrt{3}}{2}$.

Для розрахунку значення координати по осі OY слід виконати такі дії. Можливо визначити співвідношення x та y . $\sin 60^\circ = \frac{1}{2} * x$; $\frac{\sqrt{3}}{2} = \frac{1}{2} * x$; $x = \frac{1}{\sqrt{3}}$; $x = 2 * y$. Таким чином отримане співвідношення $x = 2 * y$. Оскільки $y = \frac{1}{2\sqrt{3}}$, то можна стверджувати що нове значення OY можна визначити як $3y = \frac{1}{2\sqrt{3}}$. Крок визначення y приведений на рис. 2.

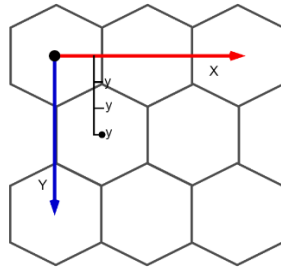


Рис. 2. Встановлення позиції центра гексагона відносно OY

Значення координат крокової траєкторії по осі OY розраховується виходячи з того, що на гексагональному растрі відстань між центрами сусідніх пікселів рівна одиниці. Таким чином, при виконання діагонального кроку координату по OX на $\frac{1}{2}$, а по осі OY на $\frac{3}{2\sqrt{3}}$.

З урахуванням останнього, нове значення оцінювальної функції знаходять за формулою $OF_{i+1} = (y_i + \frac{3}{2\sqrt{3}}) * \Delta x - (x_i + \frac{1}{2}) * \Delta y = y_i * \Delta x - x_i * \Delta y - \frac{\Delta y}{2} = OF_i + \frac{3 * \Delta x}{2\sqrt{3}} - \frac{\Delta y}{2}$.

Отримані формули та розрахунки справедливі для інтерполяції відрізків прямих з кутами нахилу від 0° до 60° по відношенню до осі OX.

При формуванні відрізків прямих з кутами нахилу від 60° до 90° елементарні кроки інтерполяції по осі OX не виконуються. В цьому випадку кожен крок виконується по обох осях, тобто діагонально, як це зображено на рис. 3.

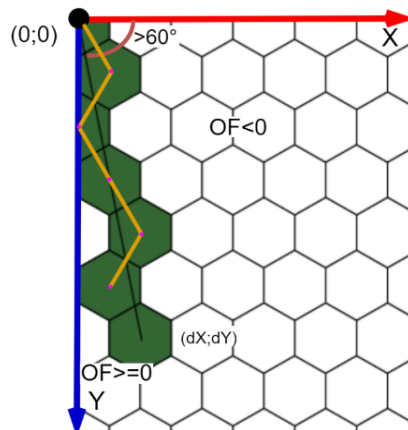


Рис. 3. Формування відрізків прямих з кутами нахилу від 60° до 90°

Покрокове формування відрізка прямої для кутів нахилу відрізків прямих в діапазоні від 60° до 90° буде обраховуватися за новими правилами. Таким чином для $OF_i \geq 0$ при виконанні діагонального кроку координата по OX буде змінюватись за формулою $x_{i+1} = x_i + \frac{1}{2}$, а по OY – $y_{i+1} = y_i + \frac{3}{2\sqrt{3}}$.

Нова оцінювальна функція: $OF_{i+1} = (y_i + \frac{3}{2\sqrt{3}}) * \Delta x - (x_i + \frac{1}{2}) * \Delta y = y_i * \Delta x + \frac{3\Delta x}{2\sqrt{3}} - x_i * \Delta y - \frac{\Delta y}{2} = OF_i + \frac{3 * \Delta x}{2\sqrt{3}} - \frac{\Delta y}{2}$.

Для $OF_i < 0$ при виконанні діагонального кроку координата по OX буде змінюватись за формулою $x_{i+1} = x_i - \frac{1}{2}$, а по OY – $y_{i+1} = y_i + \frac{3}{2\sqrt{3}}$.

Оцінювальна функція для даного випадку: $OF_{i+1} = (y_i + \frac{3}{2\sqrt{3}}) * \Delta x - (x_i - \frac{1}{2}) * \Delta y = y_i * \Delta x + \frac{3\Delta x}{2\sqrt{3}} - x_i * \Delta y + \frac{\Delta y}{2} = OF_i + \frac{3 * \Delta x}{2\sqrt{3}} + \frac{\Delta y}{2}$.

Програмний застосунок виконує такий алгоритм дій, що зображено на рис. 4. У результаті введення координати [7;4] у програму як кінцевої для формування відрізка користувач отримує результат, що зображено на рис. 5.

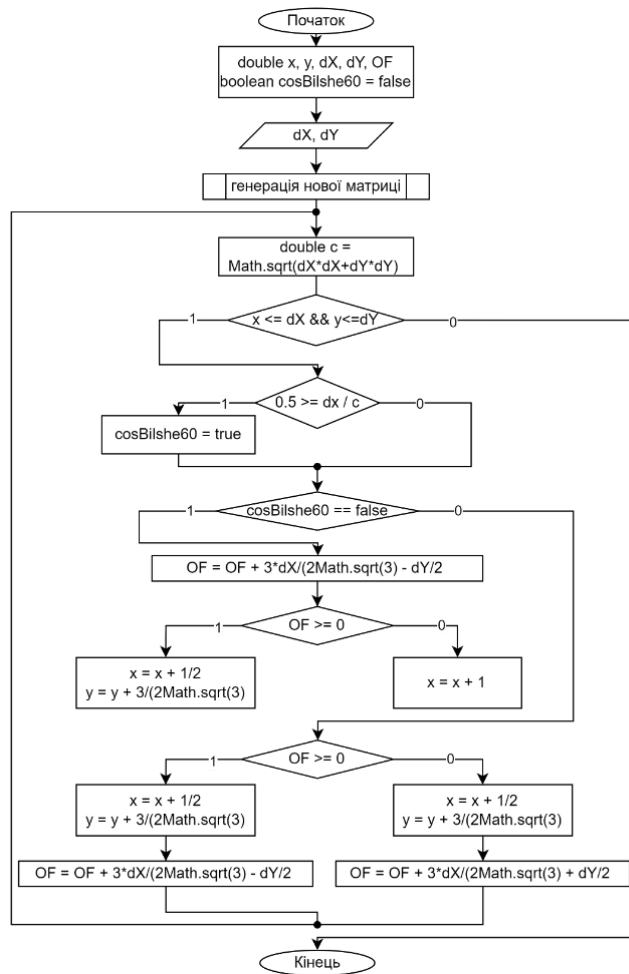


Рис. 4. Алгоритм формування відрізка

Для програмної реалізації даного методу актуальним є розробка алгоритму для формування лінії на імітованому екрані застосунку. Отриманий результат формування алгоритму за описаною процедурою формування відрізка на гексагональному екрані зображено на рис. 5.

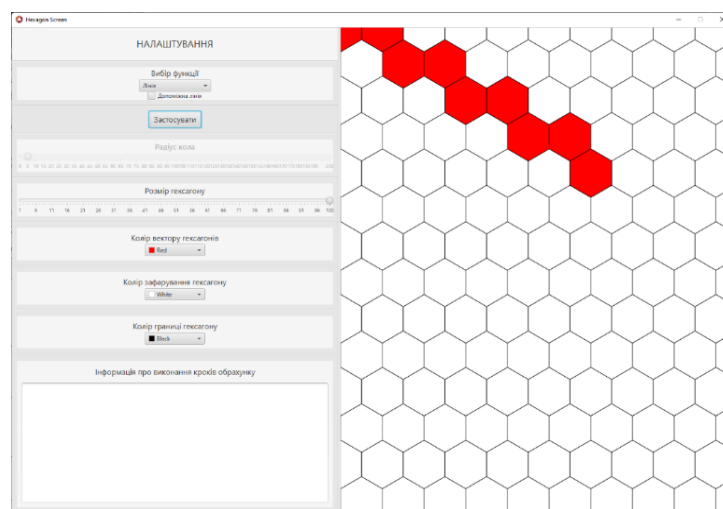


Рис. 5. Результат виконання програми

Програмна реалізація надає можливість формувати гексагон різного розміру. Таким чином для перевірки вірності виконання алгоритму пропонується використати таку координату, що буде легко

перевірити виконуючи власні розрахунки. Для зручності розгляду пікселів користувач може збільшити гексагон до 100 приведених пікселів на екрані користувача.

Інші налаштування залишаються у позиціях, що виставлені «по замовчуванню». Застосувавши такі налаштування можливо чітко відслідкувати роботу алгоритму. Користувач за потреби може встановити нове положення кінцевого пікселя.

Для перевірки максимально можливого результату імітації гексагонального пікселя на екрані користувач може встановити відповідний параметр налаштувань на мінімальне значення. Для того, щоб полегшити взаємодію із користувачем було вирішено зробити імітований екран інтерактивним. Схема алгоритму взаємодії із користувачем зображено на рис. 6.

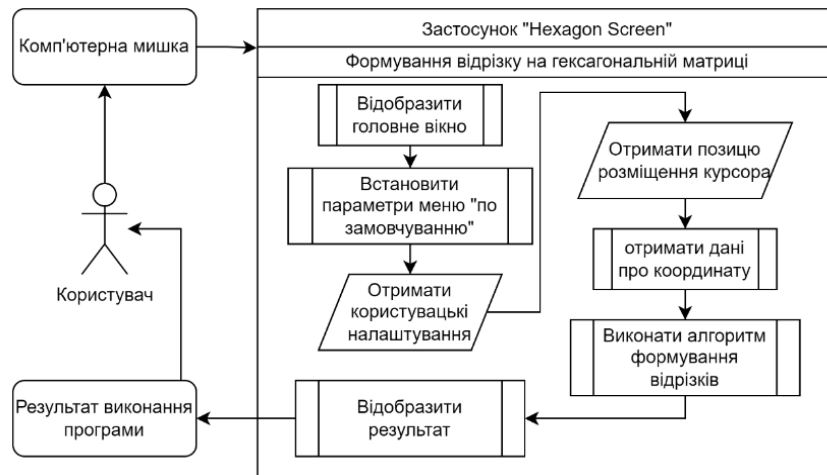


Рис. 6. Алгоритм взаємодії із користувачем

Таким чином користувач має змогу обирати кінцеву точку на екрані самостійно за допомогою натискання клавші мишки на гексагони робочого поля. Після виконання даної дії програма створює новий відрізок на основі отримання розташування кінцевої координати пікселя. У результаті програма відображає імітовану матрицю на екрані, як це зображено на рис. 7.

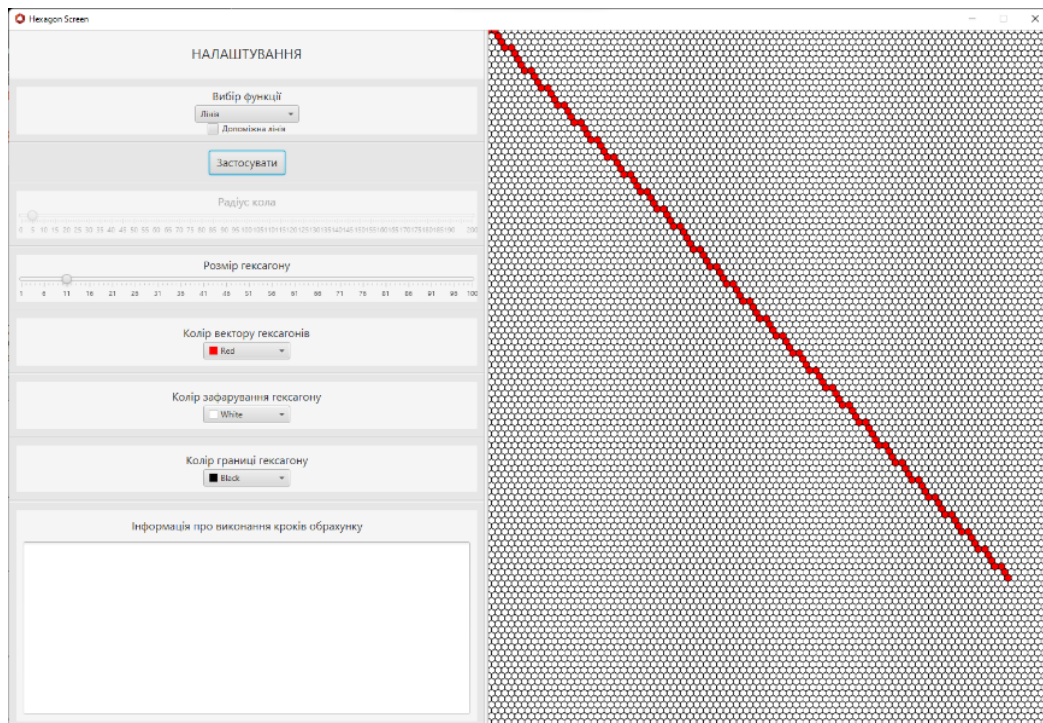


Рис. 7. Формування відрізка зі розміром гексагону в 11 пікселів

Для зручності можливо встановити такі характеристики відображення гексагонів, що задовольнять потреби користувача. Для прикладу можливо відобразити ситуацію, як зображено на рис. 8, коли потрібно отримати матрицю із розміром гексагону відносно екрану, яким користується користувач, у три піксела та білий екран із зафарбуванням відрізка у синій колір.

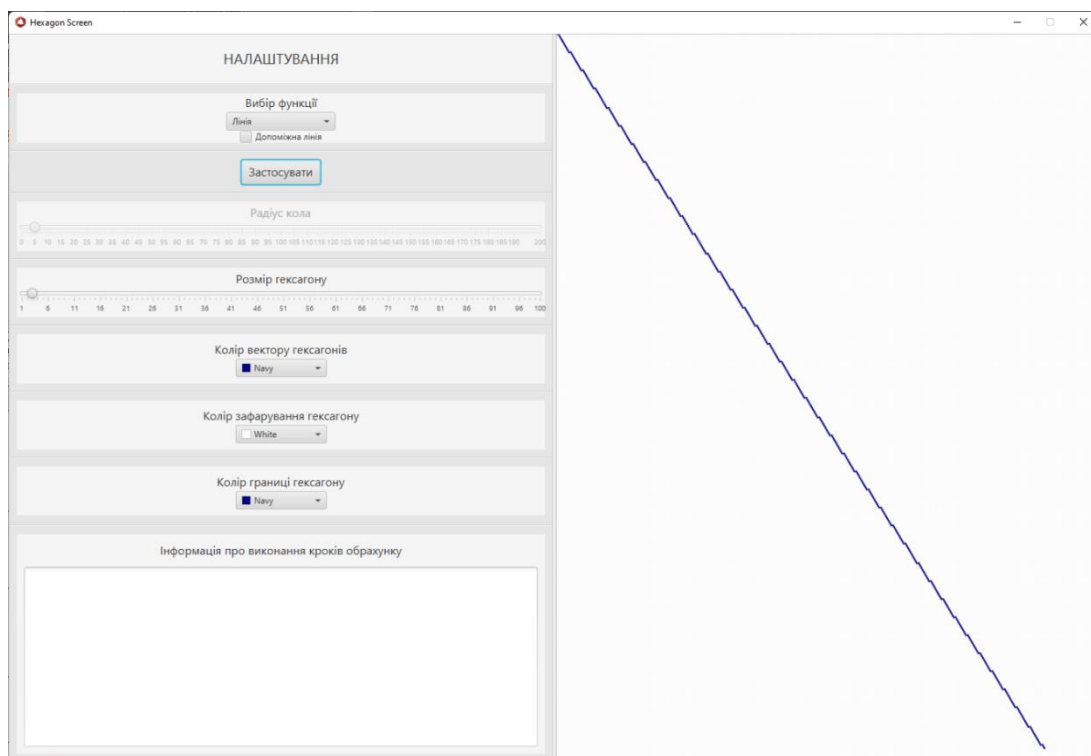


Рис. 8. Формування відрізка зі розміром гексагону в 3 піксела

Таким чином отримано програмний модуль застосунка, що надає можливість формувати відрізки на основі імітованої матриці гексагонального растру за використання звичної декартової системи відліку. Програмний додаток реалізовано за допомогою мови програмування Java у середовищі розробки IntelliJ IDEA. Виконано засобами збору проекту Apache Maven з використанням платформи JDK Java та вбудованого середовища розробки IntelliJ IDEA 2022. Для відображення інтерфейсу та результату виконання було взято бібліотека JavaFX.

Висновки

У роботі розроблено програмне забезпечення для імітування гексагонального растру на екрані із квадратним растром. Розроблено алгоритми реалізації формування відрізків на відповідному растрі. Проведено аналіз алгоритму побудови відрізків. Наведено результати виконання застосунка.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Романюк О. Н. Особливості гексагональної моделі піксела / О. В. Мельник, О. Н. Романюк // Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техніка в технологічних процесах», Хмельницький, ХНУ, 2014р. №1 (46) – 214 с. С. 91-95.2.
2. Петух А.М. Інтерполяція в задачах контурного формоутворення [Монографія] / Петух А.М., Обідник Д.Т., Романюк О.Н. – Вінниця: УНІВЕРСУМ-Вінниця, 2007. – 103 с.
3. Романюк О.Н, Використання методу оцінювальної функції для задач антилінійного /М. С. Курінний, О. Н. Романюк // Сборник научных трудов НГУ. – №19. – Том 2. – Дніпропетровськ: НГУ. – 2004. – С. 200-208.
4. Романюк О. Н. Формування відрізків прямих на гексагональному растрі / О. Н. Романюк, О. В. Мельник, О. В. Романюк // Наукові праці Донецького національного технічного університету. Се-

рія : Інформатика, кібернетика та обчислювальна техніка. - 2016. - № 2. - С. 69-72. - Режим доступу: http://nbuv.gov.ua/UJRN/Npdntu_inf_2016_2_10.

Шмалюх Владислав Анатолійович — студент третього курсу групи ЗПІ-19б, ФІТКІ, Вінницький національний технічний університет, Вінниця, e-mail: zskat02@gmail.com.

Романюк Олександр Никифорович — д.т.н., професор, завідувач кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: rom8591@gmail.com.

Мельник Олександр Васильович — пошукач кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: vinncei@gmail.com

Shmalyukh Vladyslav Anatoliyovych - thrid-year student of group ЗPI-19b, FITKI, Vinnytsia National Technical University, Vinnytsia, e-mail: zskat02@gmail.com.

Romanyuk Oleksandr Nykyforovych - Doctor of Technical Sciences, Professor, Head of the Software Department, Vinnytsia National Technical University, Vinnytsia, e-mail: rom8591@gmail.com.

Melnyk Oleksandr Vasyliovych - software seeker, Vinnytsia National Technical University, Vinnytsia, e-mail: vinncei@gmail.com