

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

Лабораторна робота №5
з дисципліни «Алгоритмізація та програмування»
Тема: " "Реалізація циклічних алгоритмів мовою C ++"
ХАІ.301.175.318.20 ЛР

Виконав студент гр. 318

_____ Аліна ХОБОТ
(підпис, дата) (П.І.Б.)

Перевірив

_____ к.т.н., доц. Олена ГАВРИЛЕНКО
(підпис, дата) (П.І.Б.)

МЕТА РОБОТИ

Вивчити теоретичний матеріал із синтаксису мовою C++ і поданням у вигляді UML діаграм циклічних алгоритмів і реалізувати алгоритми з використанням інструкцій циклу з передумовою, циклу з післяумовою і параметризованого циклу мовою C++ в середовищі VisualStudio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Дано дійсні числа (x_i, y_i) , $i = 1, 2, \dots, n$, – координати точок на площині. Визначити кількість точок, що потрапляють в фігуру заданого кольору (або групу фігур).

г - радіус кола
а - сторона великого квадрата

-  20 вар.
-  21 вар.
-  22 вар.
-  23 вар.
-  24 вар.
-  25 вар.
-  26 вар.

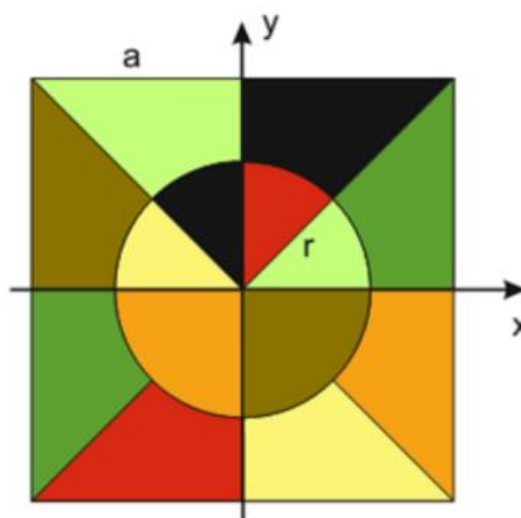


Рис. 1 – Умова завдання таблиця 1, варіант 25.

Завдання 2. Дано дійсне число x і натуральне число n . Необхідно:

- а) Обчислити значення виразу при заданих x і n для виразу з табл.2.
- б) Вивести: для парних варіантів – значення кожного третього елемента, для непарних – значення кожного четвертого елемента.

$$-\sum_{k=1}^n \frac{(-1)^k (-1 + \cos(x))^k}{k}, \quad |-1 + \cos(x)| < 1$$

Рис. 2 – Умова завдання таблиця 2, варіант 20.

Завдання 3. Дослідити ряд на збіжність. Умова закінчення циклу обчислення суми прийняти у вигляді: $|u_n| < \epsilon$ або $|u_n| > g$, де ϵ – мала величина для переривання циклу обчислення суми збіжного ряду ($\epsilon = 10^{-5} \dots 10^{-20}$); g – величина для переривання циклу обчислення суми розбіжного ряду ($g = 10^2 \dots 10^5$).

$$\sum_{n=1}^{\infty} \frac{x^n * n!}{n^{n/2}}$$

Рис. 3 – Умова завдання таблиця 3, варіант 30.

Завдання 4. Організувати меню в командному вікні для багаторазового виконання завдань *та для перевірки вхідних даних на коректність описати функції, що повертають логічне значення (true – в разі коректного значення переданих параметрів і false – в іншому випадку).

ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення завдання таблиця 1, варіант 25.

Вхідні дані (ім'я, опис, тип, обмеження):

int N – кількість точок, цілий тип, >0 .

float r – радіус кола, дійсний тип, >0 .

float b – сторона квадрата, дійсний тип, >0 .

float x, y – координати точки, дійсний тип.

Вихідні дані (ім'я, опис, тип):

bool in_circle – перевірка на попадання в чверть кола в 3-й чверті, логічний тип.

bool in_square_part – Перевірка на попадання в частину квадрата (4-та чверть), логічний тип.

Лістинг коду вирішення завдання таблиця 1, варіант 25 наведено в дод. А (стор. 6)

Екран роботи програми показаний на рис. Б.1 (дод. Б, стор. 12)

Завдання 2. Вирішення завдання таблиця 2, варіант 20.

Вхідні дані (ім'я, опис, тип, обмеження):

double x – число, дійсний тип із подвійною точністю, >0 .

Вихідні дані (ім'я, опис, тип):

Значення виразу – число, результат виразу.

Кожний третій елемент суми – число, елементи суми, що мають індекси, кратні трьом.

Лістинг коду вирішення завдання таблиця 2, варіант 20 наведено в дод. А (стор. 6)

Екран роботи програми показаний на рис. Б.2 (дод. Б, стор. 13)

Завдання 3. Вирішення завдання таблиця 3, варіант 30.

Вхідні дані (ім'я, опис, тип, обмеження):

double x – значення кута в градусах, дійсний тип із подвійною точністю, >0 .

Вихідні дані (ім'я, опис, тип):

double converging_sum – обчислена сума збіжного ряду для введенного кута в градусах, дійсний тип із подвійною точністю.

double diverging_sum – обчислена сума розбіжного ряду для введенного кута в градусах, дійсний тип із подвійною точністю.

Лістинг коду вирішення завдання таблиця 3, варіант 30 наведено в дод. А (стор. 6)

Екран роботи програми показаний на рис. Б.2 (дод. Б, стор. 14)

Завдання 4.

Організація меню.

Вхідні дані (ім'я, опис, тип, обмеження):

“Номер завдання:” – введення номеру завдання.

Вихідні дані (ім'я, опис, тип):

Choice 1 - якщо ввели цифру «1», виводиться вирішення таблиця 1, варіант 25.

Choice 2 - якщо ввели цифру «2», виводиться вирішення таблиця 2, варіант 20.

Choice 3 - якщо ввели цифру «3», виводиться вирішення таблиця 3, варіант 30.

Choice 4 – якщо ввели цифру «4», виходить з програми.

ВИСНОВКИ

Було вивчено теоретичний матеріал із синтаксису мовою C ++ і поданням у вигляді UML діаграм циклічних алгоритмів і реалізовано алгоритми з використанням інструкцій циклу з передумовою, циклу з післяумовою і параметризованого циклу мовою C ++ в середовищі VisualStudio.

ДОДАТОК А

Лістинг коду програми

```
#include <iostream>

#include <cmath>

#include <limits>

using namespace std;

// Функція для перевірки коректності введеного числа (число більше нуля)
bool isValidNumber(float value) {
    return value > 0;
}

// Функція для перевірки коректності введеного числа x для логарифму
bool isValidLogArgument(double cos_x) {
    double log_arg = -1 + cos_x;
    return fabs(log_arg) > 1;
}

// Функція для перевірки правильності введеного значення r і b
bool isValidCircleAndSquare(float r, float b) {
    return r > 0 && b > 0 && b >= r * sqrt(2);
}

// Функція для обчислення кількості точок, що потрапляють в одну з фігур
void task1() {
    int n;
    cout << "Введіть кількість точок (n): ";
    cin >> n;

    float r, b;
    cout << "Введіть радіус кола (r): ";
    cin >> r;
    cout << "Введіть сторону квадрата (b): ";
    cin >> b;
```

```

    if (!isValidCircleAndSquare(r, b)) {
        cout << "Помилка: Має бути числове значення, більше за нуль, не нульове  
та b >= r * sqrt(2)!" << endl;
        return;
    }

    int count = 0;

    cout << "Введіть координати точок (x, y):" << endl;
    for (int i = 0; i < n; ++i) {
        float x, y;
        cout << "Точка " << i + 1 << ": ";
        cin >> x >> y;

        // Перевірка на попадання в чверть кола в 3-й чверті
        bool in_circle = (x <= 0 && y <= 0 && (x * x + y * y <= r * r));

        // Перевірка на попадання в частину квадрата (4-та чверть)
        bool in_square_part = (x > 0 && y < 0 &&
                                x <= b && y >= -b &&
                                x * x + y * y >= r * r &&
                                y >= -x);

        if (in_circle || in_square_part) {
            count++;
        }
    }

    cout << "Кількість точок, що потрапляють в одну з фігур: " << count << endl;
}

// Функція для обчислення логарифму та суми за умовою
void task2() {
    double x;
    int num_elements; // Змінено ім'я змінної для кількості елементів суми

```

```

cout << "Введіть значення x (градуси): ";
cin >> x;

// Переведення градусів у радіани
double x_rad = x * M_PI / 180.0;

cout << "Введіть значення n (кількість елементів): ";
cin >> num_elements;

double cos_x = cos(x_rad);
if (!isValidLogArgument(cos_x)) {
    cout << "Помилка: Умова  $|-1 + \cos(x)| > 1$  не виконана." << endl;
    return;
}

double log_arg = -1 + cos_x;
double log_value = log(log_arg);

double sum = 0;
for (int k = 1; k <= num_elements; ++k) {
    sum += pow(-1, k) / (k * pow(log_arg, k));
}

cout << "Значення виразу: " << log_value - sum << endl;

cout << "Кожний третій елемент суми:" << endl;
for (int k = 1; k <= num_elements; ++k) {
    if (k % 3 == 0) {
        double term = pow(-1, k) / (k * pow(log_arg, k));
        cout << "Елемент " << k << ": " << term << endl;
    }
}
}

// Функція для обчислення збіжного ряду

```



```
double calculate_converging_series(double x, double epsilon = 1e-5) {
    double sum = 0.0;
    double term = x;
    int n = 1;

    while (fabs(term) > epsilon) {
        sum += term;
        n++;
        term = pow(x, n) / tgamma(n + 1);
    }

    return sum;
}
```

// Функція для обчислення розбіжного ряду

```
double calculate_diverging_series(double x, double g = 1e2) {
    double sum = 0.0;
    double term = x;
    int n = 1;

    while (fabs(term) < g) {
        sum += term;
        n++;
        term = pow(x, n) / tgamma(n + 1);
    }

    return sum;
}
```

// Функція для обчислення рядів

```
void task3() {
    double x;

    cout << "Введіть значення x (градуси): ";
    cin >> x;
```

```

// Переведення градусів у радіани
double x_rad = x * M_PI / 180.0;

double converging_sum = calculate_converging_series(x_rad);
cout << "Сума збіжного ряду: " << converging_sum << endl;

double diverging_sum = calculate_diverging_series(x_rad);
cout << "Сума розбіжного ряду: " << diverging_sum << endl;
}

// Головне меню
int main() {
    int choice;
    do {
        cout << "Меню:\n";
        cout << "1. Завдання 1: Кількість точок в фігурах\n";
        cout << "2. Завдання 2: Обчислення виразу\n";
        cout << "3. Завдання 3: Обчислення рядів\n";
        cout << "4. Вихід\n";
        cout << "Виберіть номер завдання: ";
        cin >> choice;

        switch (choice) {
            case 1:
                task1();
                break;
            case 2:
                task2();
                break;
            case 3:
                task3();
                break;
            case 4:
                cout << "Вихід з програми." << endl;
                break;
        }
    } while (choice != 4);
}

```

```
        default:
            cout << "Невірний вибір. Спробуйте ще раз." << endl;
        }

    } while (choice != 4);

    return 0;
}
```

ДОДАТОК Б
Скрін-шоти вікна виконання програми

```
Меню:
1. Завдання 1: Кількість точок в фігурах
2. Завдання 2: Обчислення виразу
3. Завдання 3: Обчислення рядів
4. Вихід
Виберіть номер завдання: 1
Введіть кількість точок (n): 5
Введіть радіус кола (r): 4
Введіть сторону квадрата (b): 10
Введіть координати точок (x, y):
Точка 1: 5 7
Точка 2: -5 6
Точка 3: -3 -2
Точка 4: -2 5
Точка 5: 6 -3
Кількість точок, що потрапляють в одну з фігур: 2
Меню:
1. Завдання 1: Кількість точок в фігурах
2. Завдання 2: Обчислення виразу
3. Завдання 3: Обчислення рядів
4. Вихід
Виберіть номер завдання: 4
Вихід з програми.

...Program finished with exit code 0
Press ENTER to exit console.
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання таблиця 1, варіант 25.

```
Меню:
1. Завдання 1: Кількість точок в фігурах
2. Завдання 2: Обчислення виразу
3. Завдання 3: Обчислення рядів
4. Вихід
Виберіть номер завдання: 2
Введіть значення x (градуси): 250
Введіть значення n (кількість елементів): 3
Значення виразу: -nan
Кожний третій елемент суми:
Елемент 3: 0.137912
Меню:
1. Завдання 1: Кількість точок в фігурах
2. Завдання 2: Обчислення виразу
3. Завдання 3: Обчислення рядів
4. Вихід
Виберіть номер завдання: 4
Вихід з програми.

...Program finished with exit code 0
Press ENTER to exit console.
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання таблиця 2, варіант 20.

```
Меню:
1. Завдання 1: Кількість точок в фігурах
2. Завдання 2: Обчислення виразу
3. Завдання 3: Обчислення рядів
4. Вихід
Виберіть номер завдання: 3
Введіть значення x (градуси): 250
Сума збіжного ряду: 77.5176
Сума розбіжного ряду: 77.5176
Меню:
1. Завдання 1: Кількість точок в фігурах
2. Завдання 2: Обчислення виразу
3. Завдання 3: Обчислення рядів
4. Вихід
Виберіть номер завдання: 4
Вихід з програми.

...Program finished with exit code 0
Press ENTER to exit console.
```

Рисунок Б.3 – Екран виконання програми для вирішення завдання таблиця 3, варіант 30.

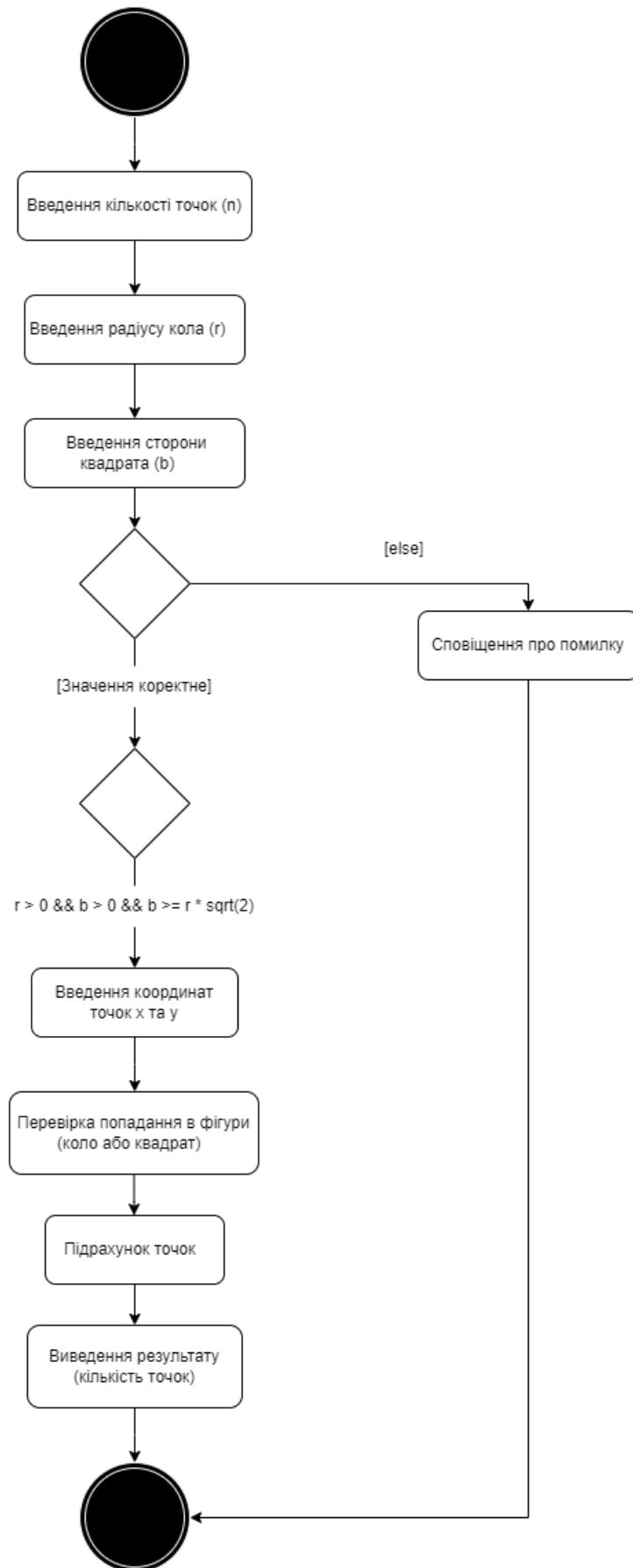


Рисунок Б.4 – Діаграма для завдання таблиця 1, варіант 25.

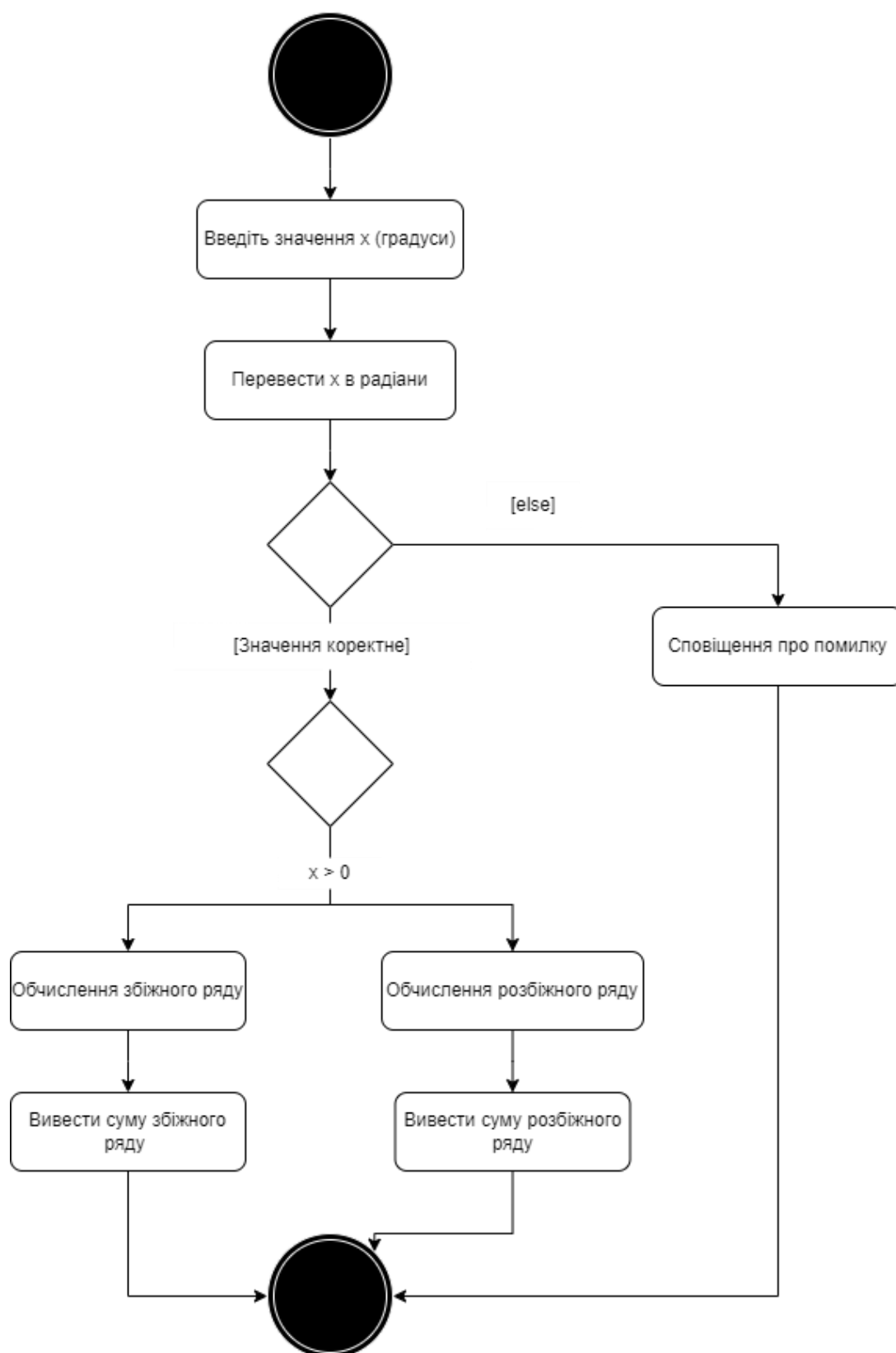


Рисунок Б.5 – Діаграма для завдання таблиця 3, варіант 30.

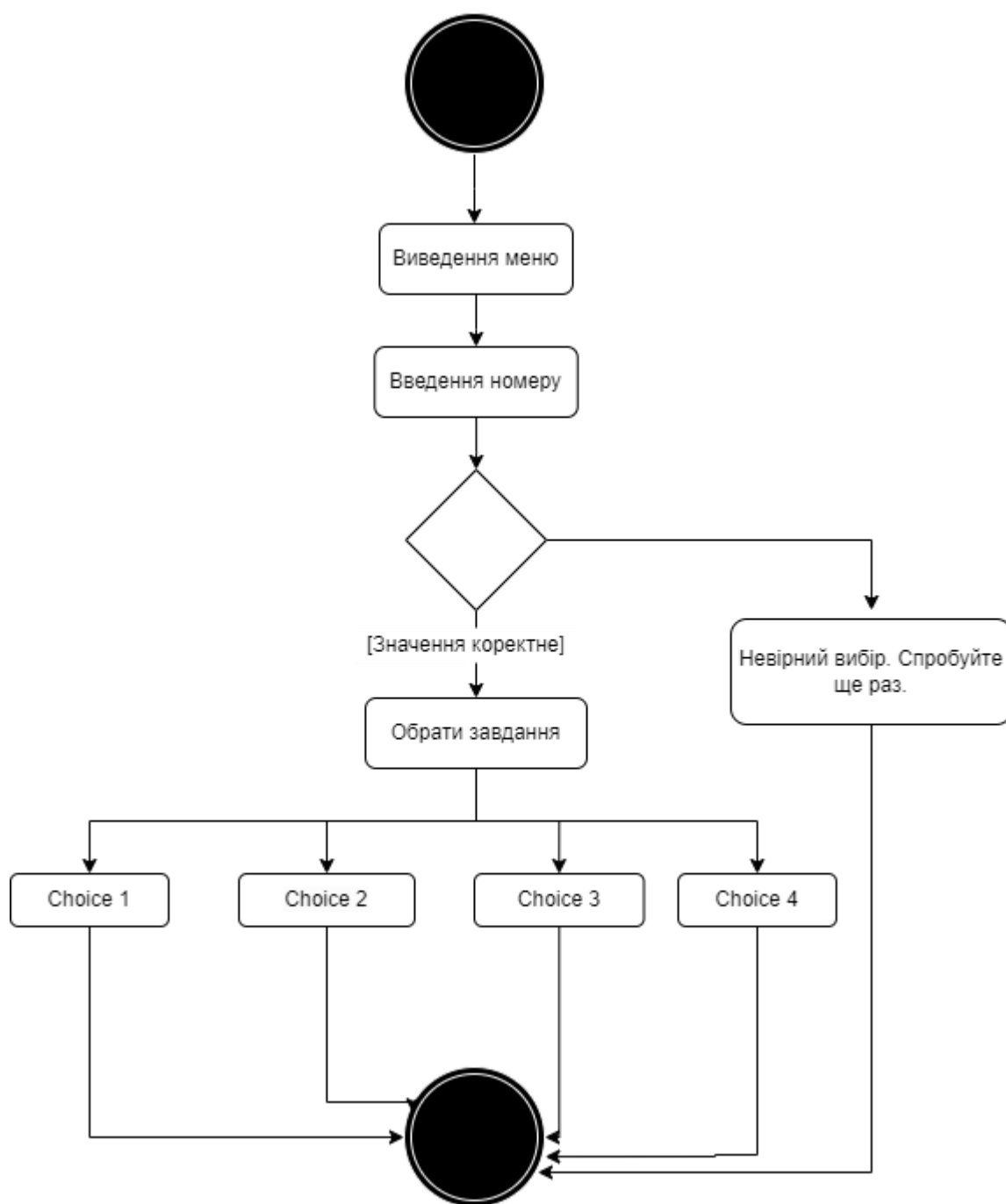


Рисунок Б.7 – Діаграма для завдання 4.