

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 № 3

з дисципліни «Алгоритмізація та програмування»  
на тему "Введення-виведення даних в C ++"

XAI.301. 141. 319a. 19 ЛР

Виконав студент гр. 319a

Овчинніков Станіслав

(підпис, 09.11.2024)

(П.І.Б.)

Перевірив

                     к.т.н., доц. Олена

ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

2024

## МЕТА РОБОТИ

Вивчити теоретичний матеріал щодо синтаксису у мові C++ і подання у вигляді UML діаграм активності алгоритмів з розгалуженням та реалізувати алгоритми з використанням інструкцій умовного переходу і вибору мовою C++ в середовищі Visual Studio. Також опанувати та відпрацювати навички структурування програми з функціями.

## ПОСТАНОВКА ЗАДАЧІ

Завдання 1 If 16 . Вирішити завдання на алгоритми з розгалуженням.

Варіанти представлено в табл.1.

Завдання 2 геом.обл 34 . Дано координати точки на площині (x, y).

Визначити, чи потрапляє точка в фігуру заданого кольору (або групу фігур) і вивести відповідне повідомлення. Варіанти фігур представлено в табл.2.

Завдання 3. Обчислити площу і периметр плоскої фігури. Варіанти представлено в табл.2.

Завдання 4. Для вибору користувачем одного з трьох зазначених вище завдань розробити алгоритм організації меню в командному вікні з використанням інструкції вибору.

## ВИКОНАННЯ РОБОТИ

Завдання 1 If 16

Дано три змінні дійсного типу: A, B, C. Якщо їх значення впорядковані за зростанням, то подвоїти їх; в іншому випадку замінити значення кожної змінної на протилежне. Вивести нові значення змінних A, B, C.

Вихідні дані (ім'я, опис, тип): A, B, C — три вещественных числа.

Вихідні дані (ім'я, опис, тип): Если числа упорядочены по возрастанию ( $A < B < C$ ), их значения удваиваются, и выводятся новые значения. Если числа не упорядочены по возрастанию, они заменяются на противоположные значения, и выводятся новые значения.

### Алгоритм вирішення

Запросить у пользователя значения переменных A, B и C.

Проверить, упорядочены ли значения по возрастанию ( $A < B < C$ ).

Если да, удвоить значения каждой переменной:

$A = A * 2$

$B = B * 2$

$C = C * 2$

Если нет, заменить значения каждой переменной на противоположные:

$A = -A$

$B = -B$

$C = -C$

Вывести новые значения A, B и C.

Завершить выполнение задачи.

```

// Завдання if16: Удвоєння або зміна знаків змінних A, B, C
void task_if16() {
    float A, B, C; // Змінні для введення даних

    cout << "***** Завдання if16 *****" << endl;

    // Введення значень A, B, C з перевіркою
    cout << "Введіть значення A, B і C: ";
    while (!(cin >> A >> B >> C)) { // Перевірка на числове значення
        cout << "Помилка! Введіть три числові значення: ";
        cin.clear();
        cin.ignore(10000, '\n'); // Очищення буфера
    }

    // Перевірка умови зростання значень
    if (A < B && B < C) {
        // Якщо значення впорядковані за зростанням, подвоюємо їх
        A *= 2;
        B *= 2;
        C *= 2;
        cout << "Значення подвоєні: A = " << A << ", B = " << B << ", C = " << C << endl;
    } else {
        // Якщо не впорядковані, змінюємо знаки
        A = -A;
        B = -B;
        C = -C;
        cout << "Значення замінені на протилежні: A = " << A << ", B = " << B << ", C = " << C << endl;
    }
}
}

```

Рисунок 1 – If 16

Лістинг коду вирішення задачі :

```

// Завдання if16: Удвоєння або зміна знаків змінних A, B, C
void task_if16() {
    float A, B, C; // Змінні для введення даних

    cout << "***** Завдання if16 *****" << endl;

    // Введення значень A, B, C з перевіркою
    cout << "Введіть значення A, B і C: ";
    while (!(cin >> A >> B >> C)) { // Перевірка на числове значення
        cout << "Помилка! Введіть три числові значення: ";
        cin.clear();
        cin.ignore(10000, '\n'); // Очищення буфера
    }

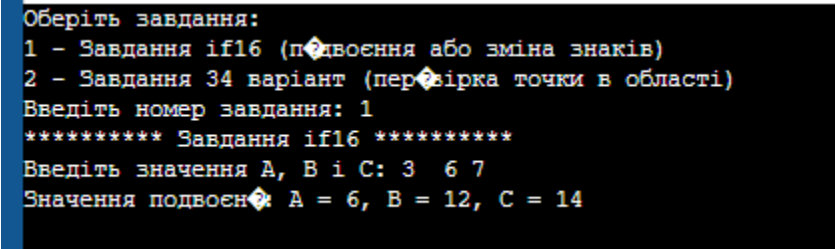
    // Перевірка умови зростання значень
    if (A < B && B < C) {
        // Якщо значення впорядковані за зростанням, подвоюємо їх

```

```

A *= 2;
B *= 2;
C *= 2;
cout << "Значення подвоєні: A = " << A << ", B = " << B << ", C = " << C << endl;
} else {
    // Якщо не впорядковані, змінюємо знаки
    A = -A;
    B = -B;
    C = -C;
    cout << "Значення замінені на протилежні: A = " << A << ", B = " << B << ", C = " << C << endl;
}
}

```



Оберіть завдання:  
 1 - Завдання if16 (подвоєння або зміна знаків)  
 2 - Завдання 34 варіант (перевірка точки в області)  
 Введіть номер завдання: 1  
 \*\*\*\*\* Завдання if16 \*\*\*\*\*  
 Введіть значення A, B і C: 3 6 7  
 Значення подвоєні: A = 6, B = 12, C = 14

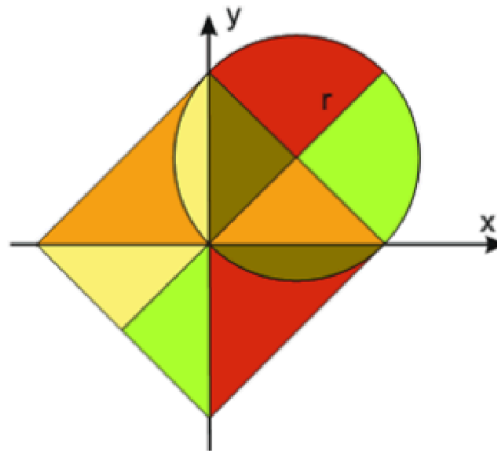
Екран роботи програми показаний на рис 2.

Завдання 2 ( Геом обл 34 )

32  
-36

**г - радіус кола**

- 32 вар.**
- 33 вар.**
- 34 вар.**
- 35 вар.**
- 36 вар.**



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

г — радіус окружности (положительное число).

х — координата точки по оси х.

у — координата точки по оси у.

Вихідні дані:

in\_area — логический результат, указывающий, находится ли точка в заданной области:

true, если точка в окружности или в одном из четырёх треугольников.

false, если точка не принадлежит ни одной из фигур.

"Точка находится в заданной области!" — если точка принадлежит области.

"Точка находится вне заданной области!" — если точка вне области.

Алгоритм вирішення:

Запросить у пользователя значение радиуса  $r$ .

Если  $r \leq 0$ , вывести сообщение об ошибке: "Радиус должен быть положительным числом!" и завершить выполнение задачи.

Запросить у пользователя координаты точки  $x$  и  $y$ .

Проверить, принадлежит ли точка заданной области:

Проверить, принадлежит ли точка окружности:  $(x * x + y * y \leq r * r) \&\& (x \geq 0) \&\& (y \geq 0)$ .

Проверить, принадлежит ли точка треугольнику 1 (правый верхний):  $(x \geq 0 \&\& y \geq 0 \&\& y \leq -x + r)$ .

Проверить, принадлежит ли точка треугольнику 2 (левый верхний):  $(x \leq 0 \&\& y \geq 0 \&\& y \leq x + r)$ .

Проверить, принадлежит ли точка треугольнику 3 (левый нижний):  $(x \leq 0 \&\& y \leq 0 \&\& y \geq x - r)$ .

Проверить, принадлежит ли точка треугольнику 4 (правый нижний):  $(x \geq 0 \&\& y \leq 0 \&\& y \geq -x - r)$ .

Если точка удовлетворяет хотя бы одному из этих условий, вывести сообщение: "Точка находится в заданной области!".

В противном случае вывести сообщение: "Точка находится вне заданной области!".

Завершить выполнение задачи.



```

// Завдання 34 варіант: Перевірка попадання точки в задану область
void task_34_variant() {
    float x, y, r; // Координати та радіус

    cout << "***** Завдання 34 варіант *****" << endl;

    // Введення радіуса з перевіркою
    cout << "Введіть радіус r: ";
    while (!(cin >> r) || r <= 0) { // Перевірка позитивного числа
        cout << "Помилка! Радіус має бути додатним числом: ";
        cin.clear();
        cin.ignore(10000, '\n');
    }

    // Введення координат точки
    cout << "Введіть координати точки (x, y): ";
    while (!(cin >> x >> y)) { // Перевірка правильного введення координат
        cout << "Помилка! Введіть числові координати: ";
        cin.clear();
        cin.ignore(10000, '\n');
    }

    // Логіка перевірки знаходження в фігурі
    bool in_circle = (x * x + y * y <= r * r) && (x >= 0) && (y >= 0); // Коло в правому верхньому квадранті
    bool in_triangle1 = (x >= 0 && y >= 0 && y <= -x + r); // Правий верхній трикутник
    bool in_triangle2 = (x <= 0 && y >= 0 && y <= x + r); // Лівий верхній трикутник
    bool in_triangle3 = (x <= 0 && y <= 0 && y >= x - r); // Лівий нижній трикутник
    bool in_triangle4 = (x >= 0 && y <= 0 && y >= -x - r); // Правий нижній трикутник

    // Результати перевірки
    if (in_circle || in_triangle1 || in_triangle2 || in_triangle3 || in_triangle4) {
        cout << "Точка знаходиться в заданій області!" << endl;
    } else {
        cout << "Точка знаходиться поза заданою областю!" << endl;
    }
}

```

Рисунок 3- код Геом обл 5.

Лістинг коду :

```

// Завдання 34 варіант: Перевірка попадання точки в задану область
void task_34_variant() {
    float x, y, r; // Координати та радіус

    cout << "***** Завдання 34 варіант *****" << endl;

    // Введення радіуса з перевіркою
    cout << "Введіть радіус r: ";
    while (!(cin >> r) || r <= 0) { // Перевірка позитивного числа
        cout << "Помилка! Радіус має бути додатним числом: ";
        cin.clear();
        cin.ignore(10000, '\n');
    }
}

```

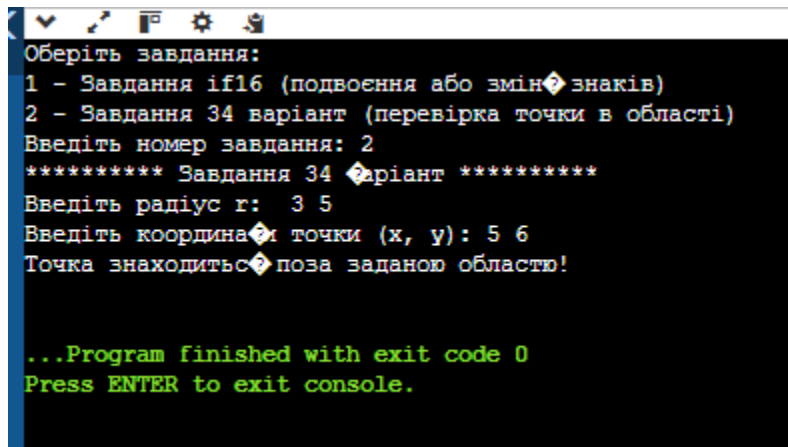
```

// Введення координат точки
cout << "Введіть координати точки (x, y): ";
while (!(cin >> x >> y)) { // Перевірка правильного введення координат
    cout << "Помилка! Введіть числові координати: ";
    cin.clear();
    cin.ignore(10000, '\n');
}

// Логіка перевірки знаходження в фігурі
bool in_circle = (x * x + y * y <= r * r) && (x >= 0) && (y >= 0); // Коло в правому верхньому квадранті
bool in_triangle1 = (x >= 0 && y >= 0 && y <= -x + r); // Правий верхній трикутник
bool in_triangle2 = (x <= 0 && y >= 0 && y <= x + r); // Лівий верхній трикутник
bool in_triangle3 = (x <= 0 && y <= 0 && y >= x - r); // Лівий нижній трикутник
bool in_triangle4 = (x >= 0 && y <= 0 && y >= -x - r); // Правий нижній трикутник

// Результати перевірки
if (in_circle || in_triangle1 || in_triangle2 || in_triangle3 || in_triangle4) {
    cout << "Точка знаходиться в заданій області!" << endl;
} else {
    cout << "Точка знаходиться поза заданою областю!" << endl;
}
}

```



```

Оберіть завдання:
1 - Завдання if16 (подвоєння або змін знаків)
2 - Завдання 34 варіант (перевірка точки в області)
Введіть номер завдання: 2
***** Завдання 34 варіант *****
Введіть радіус r: 3 5
Введіть координати точки (x, y): 5 6
Точка знаходиться поза заданою областю!

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

```

Екран роботи програми показаний на рис 4.

## Завдання 4

### Вирішення задачі

Завдання 4. Для вибору користувачем одного з трьох зазначених вище завдань розробити алгоритм організації меню в командному вікні з використанням інструкції вибору.

Вхідні дані (ім'я, опис, тип, обмеження): menu — целое число, введённое пользователем для выбора задачи:

1 — выбрать задачу if16.

2 — выбрать задачу 34 (вариант).

В зависимости от выбора задачи:

Если выбрана задача if16, потребуются значения для переменных A, B и C.

Если выбрана задача 34 (вариант), потребуются значения для радиуса r и координат точки x, y.

Вихідні дані : Сообщение о выбранной задаче или ошибке:

"Значения удвоены:  $A = \dots$  ,  $B = \dots$  ,  $C = \dots$ " или "Значения заменены на противоположные:  $A = \dots$  ,  $B = \dots$  ,  $C = \dots$ " — для задачи if16.

"Точка находится в заданной области!" или "Точка находится вне заданной области!" — для задачи 34 (вариант).

"Неверный номер задачи! Введите 1 или 2." — если введено некорректное значение menu.

## Алгоритм вирішення:

Начало.

Вивести меню с вариантами задач:

Задача 1: Проверка чисел А, В, С и их обработка (задача if16).

Задача 2: Проверка попадания точки в сложную область (задача 34 (вариант)).

Запросить у пользователя ввод номера задачи (menu).

Проверить значение menu:

Если menu равно 1, выполнить алгоритм задачи if16.

Если menu равно 2, выполнить алгоритм задачи 34 (вариант).

Если menu не равно ни 1, ни 2, вывести сообщение об ошибке: "Неверный номер задачи! Введите 1 или 2." и вернуться к шагу 3.

Завершить выполнение программы после выполнения выбранной задачи.

Экран работы програми показаний на рис 5.

## Лістинг коду :

```
// Головна функція програми
int main() {
    int menu; // Змінна для вибору завдання

    // Меню вибору завдання
    cout << "Оберіть завдання:" << endl;
    cout << "1 - Завдання if16 (подвоєння або зміна знаків)" << endl;
    cout << "2 - Завдання 34 варіант (перевірка точки в області)" << endl;

    // Введення вибору з перевіркою
    cout << "Введіть номер завдання: ";
    while (!(cin >> menu) || (menu != 1 && menu != 2)) {
        cout << "Помилка! Введіть 1 або 2: ";
        cin.clear();
        cin.ignore(10000, '\n');
    }

    // Виконання обраного завдання
    switch (menu) {
        case 1:
```

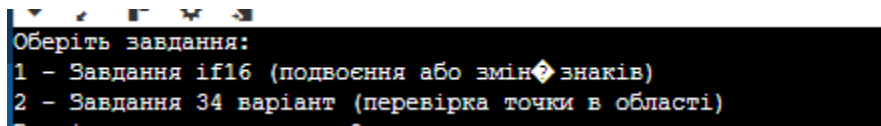
```

        task_if16();
        break;
    case 2:
        task_34_variant();
        break;
    default:
        cout << "Невірний номер завдання!" << endl;
    }

    return 0; // Завершення програми
}

```

Алгоритм вирішення:



Екран роботи програми показаний на рис 6.

## ВИСНОВКИ

Програма успішно реалізує меню для вибору между двумя задачами, что позволяет пользователю интуитивно выбрать и выполнить нужную операцию. Проверка корректности ввода и модульная структура кода делают программу надежной и легко расширяемой.