Міністерство освіти України

Національний технічний університет "ХПІ"

Кафедра "Системний аналіз та інформаційно-аналітичні технології"

**Звіт**

**Лабораторна робота 3**

з дисципліни "Об’єктно орієнтоване програмування"

Виконав: студент групи КН-1223Б

Фільчак М.В.

Перевірив:

Івашко А.В.

Харків 2023

Зміст

[Мета роботи 3](#_Toc148186193)

[Завдання 3](#_Toc148186194)

[Алгоритм завдання 3](#_Toc148186195)

[Базові функції 3](#_Toc148186196)

[Розробка програми 4](#_Toc148186197)

[Висновок 6](#_Toc148186198)

**Лабораторна робота №3**  
**Базові типи даних та введення — виводу**

Варіант 16

## Мета роботи

Метою лабораторної роботи є отримання практичних навичок в роботі з умовними операторами while() та if() і розгалуженими алгоритмами в мові C++.

Завдання

|  |
| --- |
|  |

Створити програму, що буде зчитувати координати точки та перевіряти чи потрапляє вона на заштриховану область на малюнку, що відповідає варіанту 16.

Алгоритм завдання

У варіанті 16 на графіку є дві функції. Перша – квадрат зі сторонами 1, що знаходиться тільки у другій чверті графіку. Друга – коло, що перебуває тільки у четрертій чверті графіку.

Програма розпочинається виводом меню в якому необхідно вибрати завдання, що буде виконуватися. Далі програма повідомить оператору дані, що їх треба надати.

Останньою частиною є перевірка введених даних на відповідність умові й виведення на екран повідомлення, що буде підтверджувати чи заперечувати виконання умови.

## Базові функції

Оператор if() дозволяє проаналізувати вказану умову і здійснити виконання деякого фрагмента програми тільки при істинному результаті аналізу. Оператор if має дві форми:

if (вираз) оператор1;

if (вираз) оператор1; else оператор2;

Якщо в результаті обчислення виразу набуте значення true, то в обох формах оператора if виконується оператор1. Якщо в результаті обчислення виразу набуте значення false, тоді виконання оператора if, представленого в першій формі, закінчується, а в операторі, що має другу форму, виконується оператор2.

Оператор switch дозволяє залежно від значення деякого виразу вибрати один з багатьох варіантів продовження програми. Оператор має наступний формат:

switch(вираз){

case(константний вираз1): оператор1; break;

case(константний вираз2): оператор2; break;

case(константний виразN): операторN; break;

default: операторN+1;

}

Виконання оператора switch починається з обчислення виразу, записаному в дужках. Цей результат послідовно порівнюється з результатами обчислення константних виразів при операторах case, і, якщо буде виявлена рівність результатів, то виконується оператор відповідного case. Якщо збіг результатів не виявлений, виконується оператор default. Якщо оператор default відсутній, то оператор switch пропускається.

## Розробка програми

Для початку необхідно підключити необхідні бібліотеки.

Ця бібліотека містить функції для операцій вводу-виводу.

#include <iostream>

Ця бібліотека декларує всі функції Windows API.

#include <Windows.h>

Далі у функцію main() додаємо підтримку кирилиці.

setlocale(LC\_ALL, "");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Через те, що у цій лабораторній роботі присутні додаткові завдання варто використовувати оператор switch() для створення початкового меню. Його можна зробити за допомогою змінної типу int, яку оператор switch() буде перевіряти на відповідність умовам операторів case: .

Далі програма за допомогою операторів cin та cout пропонує оператору ввести значення змінних x та y, що записуються у форматі double.

Потім, за допомогою операторів if() та else(), програма перевіряє дані введені оператором на відповідність умовам. У разі підтрердження дійсності однієї з умов на екран виводиться повідомлення: “The point falls into the specified area”, якщо координати введеної точки не відповідають умовам – виводиться наступне повідомлення: “ The point does not fall into the specified area”.

Далі наведено повний код програми:

#include <iostream>

#include <locale>

#include <Windows.h>

using namespace std;

#define RESET "\033[0m"

#define RED "\033[31m"

#define GREEN "\033[32m"

#define YELLOW "\033[33m"

#define BLUE "\033[34m"

void do\_main\_task() {

double x, y;

cout << "\n Input x value: \n";

cin >> x;

cout << "Input y value: \n";

cin >> y;

cout << GREEN << x << RESET << ";" << GREEN << y << endl;

if (((x \* x) + (y \* y) <= 1) && (x >= 0) && (y <= 0)) {

cout << BLUE << "\nThe point falls into the specified area\n" << RESET;

}

else if ((y >= x) && (0 <= y) && (y <= 1) && (-1 <= x) && (x <= 0)) {

cout << BLUE << "\nThe point falls into the specified area\n" << RESET;

}

else {

cout << RED <<"\nThe point does not fall into the specified area" << RESET;

}

}

int main()

{

setlocale(LC\_ALL, "");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int i;

while (true) {

cout << "\n\n1. Main task" << endl;

cout << "2. Task A" << endl;

cout << "3. Task B" << endl;

cout << "4. Exit" << endl;

cin >> i;

switch (i) {

case 1:

do\_main\_task();

break;

case 2:

do\_sqr\_root();

break;

case 3:

do\_mushrooms();

break;

case 4:

exit(0);

break;

default:

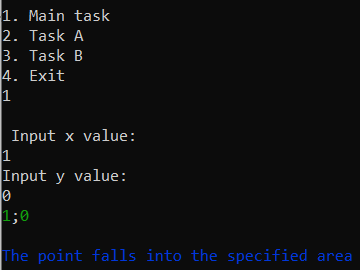
cout <<RED << "\nPlease, input numbers from 1 to 4." << RESET;

break;

}

}

}

Результат роботи надано скріншотом: 

## Висновок

Під час виконання цієї лабораторної роботи ми навчилися створювати контекстне меню для програм з декількох компонентів за допомогою оператора switch(), що перевіряє умову на відповідність операторам case: . Ці знання будуть корисними під час виконання лабораторних робіт з декількома завданнями.

Важливим також є отримання навичок роботи з умовними операторами if() та else, адже їх функціонал є досить потужним, вони будуть необхідними для виконання завдань. Для їх правильної роботи слід уважно перевірити коректність умови, а саме використання відповідних умовних позначень.

Іншим компонентом лабораторної роботи, що варто розглянути є зміна кольорів консолі та тексту в ній. Це можна робити різними способами, але найбільш компактним та зручним мені здалося використання функції #define для визначення кольорів, з допомогою цього можна спростити синтаксис у програмі. Це виглядає наступним чином:

#define RESET "\033[0m"

#define GREEN "\033[32m"

cout << GREEN << x << RESET << ";" << GREEN << y << endl;