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 "Реалізація алгоритмів з розгалуженням мовою С ++"

ХАІ.301. 6 ЛР

Виконав студент гр. _____318_____

__12.11.2023_____*Завада Тарас*_____
(підпис, дата) (П.І.Б.)

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

2023

МЕТА РОБОТИ

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

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання на алгоритми з розгалуженням. *If11 - Дано два числа. Якщо числа рівні - потрібно присвоїти обом числам значення 0. Якщо вони різні - присвоїти меншому числу значення більшого*.

Завдання 2. Дано координати точки на площині (x, y). Визначити, чи потрапляє точка в фігуру заданого кольору (або групу фігур) і вивести відповідне повідомлення.

Завдання 3. Обчислити площу і периметр плоскої фігури. *Дан радіус R.*

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі If

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

змінні типу int *firstnumb, secondnumb*

Вихідні дані (ім’я, опис, тип):

змінні типу int *firstnumb, secondnumb*

Алгоритм вирішення показано на рис. 4

Лістинг коду вирішення задачі розділ і номер задач(і) наведено в дод. А (стор. 4-5).

Екран роботи програми показаний на рис. Б.х.

Завдання 2.

Вирішення задачі

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

змінні типу double *x, y, r*

Вихідні дані (ім’я, опис, тип):

змінні типу bool *figure1, figure2, common*

Алгоритм вирішення показано на рис. 5

Лістинг коду вирішення задачі розділ і номер задач(і) наведено в дод. А (стор. 5-7).

Завдання 3.

Вирішення задачі

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

змінні типу double *r*

Вихідні дані (ім’я, опис, тип):

змінні типу double *figure_2surface, figure_1surface, figure_2perimeter, figure_1perimeter*

Алгоритм вирішення показано на рис. 6

Лістинг коду вирішення задачі розділ і номер задач(і) наведено в дод. А (стор. 7-8).

ВИСНОВКИ

Було вивчени логічні оператори || &&, були проблеми з другим завданням та знаходженням площі кола

ДОДАТОК А
Лістинг коду програми

```
   #include <iostream>
#define _USE_MATH_DEFINES // for C++
#include <math.h>
using namespace std;
//If1.Дано ціле число. Якщо воно є додатним, то відняти від нього 8;
// інакше не змінювати його. Вивести отримане число.
void task_if1(void); // завдання 1 оголошення функції
// Дано координати точки на площині (x, y).
// Визначити, чи потрапляє точка в фігуру заданого кольору (або групу фігур)
// і вивести відповідне повідомлення.
void task_geom2(void);// завдання 2 оголошення функції
// Дано радіус r.
// Визначти площину та периметр двох фігур
void task_geom3(void);// завдання 3 оголошення функції
//Функції 2 Завдання:
bool defaultFunction(double x, double y, double r); // Функція 1
bool reversedFunction(double x, double y, double r); // Функція 2
bool negativeFunction(double x, double y, double r); // Функція 3
bool circleFunction(double x, double y, double r); // Функція належності до кола
int main() {
      int menu;
      cout << "Task number:";
      cin >> menu;
      switch (menu)
      { // перемикання між завданнями
      case 1: task_if1(); break; // Завдання 1
      case 2: task_geom2(); break; // Завдання 2
      case 3: task_geom3(); break; // Завдання 3
      default: cout << "Wrong task! (Only 1,2,3)" << endl; //повідомлення про
помилку
      }
      system("pause");
      return 0;

}


void task_if1(void)// завдання 1 реалізація
{
      //Декларація змінних
      signed int firstnumber, secondnumber;
      cout << "************* If 11 *****************" << endl;
      //Введення даних
      cout << "Type First Number:";
```

```
	cin >> firstnumber;
	cout << "Type Second Number:";
	cin >> secondnumber;
	//Знаходження чисел за формулою
	if (firstnumber != secondnumber) //Якщо числа різні, то привласнюємо меншому числу значення більшого
	{
		if (firstnumber > secondnumber)
		{
			secondnumber = firstnumber;
		}
		else
		{
			firstnumber = secondnumber;
		}
	}
	else // Якщо числа рівні, то оба числа повинні дорівнювати нулю
	{
		firstnumber = 0;
		secondnumber = 0;
	}
	//Вивід даних
	cout << "First Number:" << firstnumber << endl; // Перше число
	cout << "Second Number:" << secondnumber << endl; // Друге число


}


void task_geom2(void)
{
	//Декларація змінних
	double X, Y, R;
	bool figure1 = false;
	bool figure2 = false;
	bool common = false;
	//Введення даних
	cout << "*************Task 2 *****************" << endl;
	cout << "Radius:" << endl;
	cin >> R;
	cout << "X:" << endl;
	cin >> X;
	cout << "Y:" << endl;
	cin >> Y;

	figure1 = circleFunction(X, Y, R) && defaultFunction(X, Y, R) && reversedFunction(X, Y, R); // Три вимоги знаходження точки на площі першої фігури
	figure2 = negativeFunction(X, Y, R) && (X > 0) && (Y < 0); // Дві вимоги знаходження точки на площі другої фігури
```

```
	common = figure1 || figure2; // Программа перевіряє, чи знаходиться на якійсь площі з цих двох фігур
	if (common)
	{
		cout << "Coordinates are on area!" << endl; // Точка знаходиться на площі
	}
	else
	{
		cout << "Coordinates are not on area!" << endl; // Точка не знаходиться на площі
	}
}

//Перелік Функцій
//Функція 1. y=x
bool defaultFunction(double x, double y, double r)
{
	double threshold;
	threshold = x;
	if (y > threshold) // Якщо координата вище прямої, то вимога виконується
	{
		return(true);
	}
	else return(false); // Якщо координата нижче прямої, то вимога не виконується
}
//Функція 2. y = -x + sqrt(2)*r
bool reversedFunction(double x, double y, double r)
{
	double threshold;
	threshold = -x + sqrt(2) * r;
	if (y > threshold) // Якщо координата вище прямої, то вимога виконується
	{
		return(true);
	}
	else return(false); // Якщо координата нижче прямої, то вимога не виконується
}
//Функція 3. y = x - sqrt(2)*r
bool negativeFunction(double x, double y, double r)
{
	double threshold;
	threshold = x - sqrt(2) * r; // Функція прямої
	if (y > threshold) // Якщо координата вище прямої, то вимога виконується
	{
		return(true);
	}
	else return(false); // Якщо координата нижче прямої, то вимога не виконується
```

```
}
//Функція 4. Коло.
bool circleFunction(double x, double y, double r)
{
	double distance, rcoord;
	rcoord = r / sqrt(2); // Координата радіусу
	distance = sqrt(pow(x - rcoord, 2) + pow(y - rcoord, 2)); // Формула дистанції даної точки до центра кола
	if (distance > r)
	{
		return(false); // Якщо дистанція від точки до центру кола менше радіусу, то точка знаходиться у колі.
	}
	else
	{
		return(true); // Якщо більше, то вона поза кола.
	}
}



void task_geom3(void)// завдання 3 реалізація
{
	//Декларація змінних
	double r, circleperimeter, figure_1perimeter, triangleperimeter, circlesurface, trianglesurface, figure_2surface, figure_1surface, triangleside, innerTriangleSurface, figure_2perimeter;
	double brownSegmentSurface;
	//Введення даних змінних
	cout << "*************Geometry 33*****************" << endl;
	cout << "Type Radius:" << endl;
	cin >> r;

	//Знаходження периметру кола та його площи
	circleperimeter = 2 * M_PI * r;
	circlesurface = M_PI * pow(r, 2);

	//Знаходження периметру та площи верхньої фігури
	figure_1surface = circlesurface / 4;
	figure_1perimeter = (circleperimeter / 4) + r + r;

	//Знаходження площи опуклості нижньої фігури
	innerTriangleSurface = pow(r,2)/2;
	brownSegmentSurface = figure_1surface - innerTriangleSurface;

	//Знаходження однієї зі сторін та площи трикутника знизу
	triangleside = sqrt(2)*r;
	trianglesurface = pow(triangleside, 2) / 2;
```

```
	//Знаходження периметру та площи нижньої фігури
	figure_2perimeter = triangleside + r + r + (circleperimeter / 4);
	figure_2surface = trianglesurface - brownSegmentSurface;

	//Вивод даних
	cout << "Figures Area:" << figure_2surface + figure_1surface << endl;

	cout << "Figures Perimeter:" << figure_2perimeter + figure_1perimeter << endl;

}
```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

рисунок 1

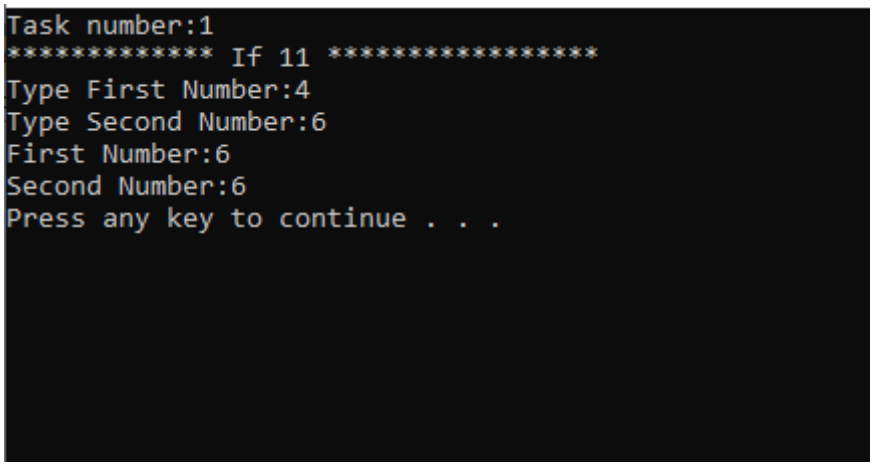

Рисунок 2

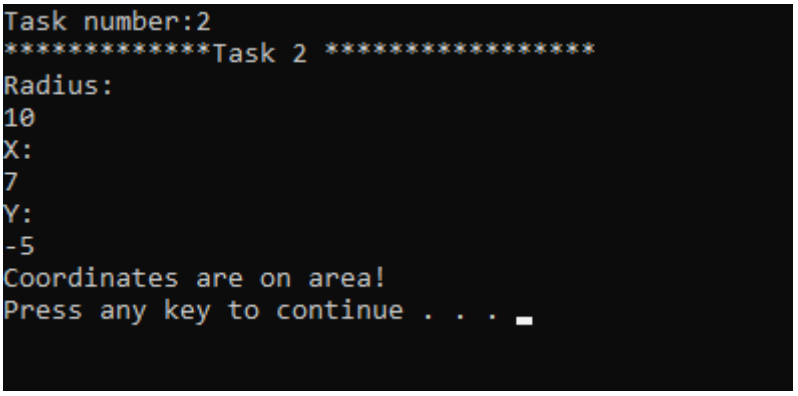

Рисунок 3

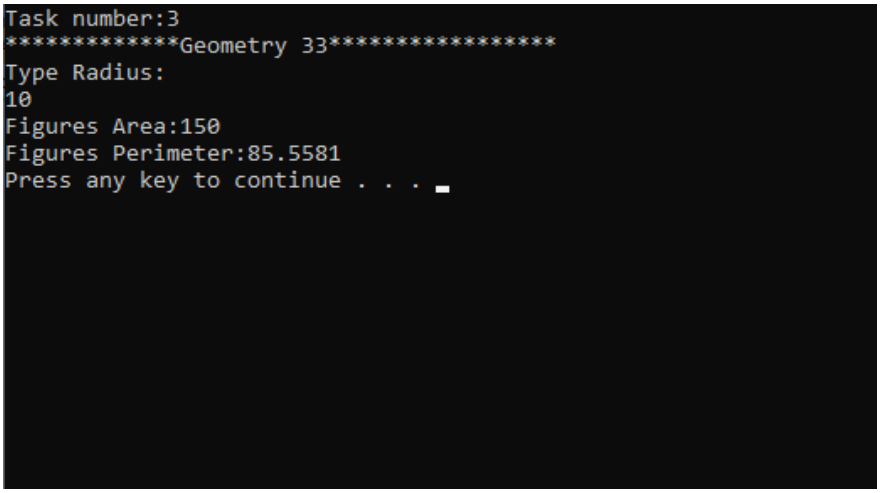

Рисунок 4

```
if (firstnumber != secondnumber) //Якщо числа різні, то привласнюємо меншому числу значення більшого
{
    if (firstnumber > secondnumber)
    {
        secondnumber = firstnumber;
    }
    else
    {
        firstnumber = secondnumber;
    }
}
else // Якщо числа рівні, то оба числа повинні дорівнювати нулю
{
    firstnumber = 0;
    secondnumber = 0;
}
```

Рисунок 5

```
cout << "*************Task 2 *****************" << endl;
cout << "Radius:" << endl;
cin >> R;
cout << "X:" << endl;
cin >> X;
cout << "Y:" << endl;
cin >> Y;

figure1 = circleFunction(X, Y, R) && defaultFunction(X, Y, R) && reversedFunction(X, Y, R); // Три вимоги знаходження точки на площі першої фігури
figure2 = negativeFunction(X, Y, R) && (X > 0) && (Y < 0); // Дві вимоги знаходження точки на площі другої фігури
common = figure1 || figure2; // Программа перевіряє, чи знаходиться на якійсь площі з цих двох фігур
if (common)
{
    cout << "Coordinates are on area!" << endl; // Точка знаходиться на площі
}
else
{
    cout << "Coordinates are not on area!" << endl; // Точка не знаходиться на площі
}
```

Рисунок 6

```
//Знаходження периметру кола та його площи
circleperimeter = 2 * M_PI * r;
circlesurface = M_PI * pow(r, 2);

//Знаходження периметру та площи верхньої фігури
figure_1surface = circlesurface / 4;
figure_1perimeter = (circleperimeter / 4) + r + r;

//Знаходження площи опуклості нижньої фігури
innerTriangleSurface = pow(r,2)/2;
brownSegmentSurface = figure_1surface - innerTriangleSurface;

//Знаходження однієї зі сторін та площи трикутника знизу
triangleside = sqrt(2)*r;
trianglesurface = pow(triangleside, 2) / 2;

//Знаходження периметру та площи нижньої фігури
figure_2perimeter = triangleside + r + r + (circleperimeter / 4);
figure_2surface = trianglesurface - brownSegmentSurface;
```