

Міністерство освіти та науки України
Національний технічний університет України «КПІ ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації та управління

ЗВІТ

З лабораторної роботи №2

На тему:

«Організація розгалужених процесів»

Виконав:

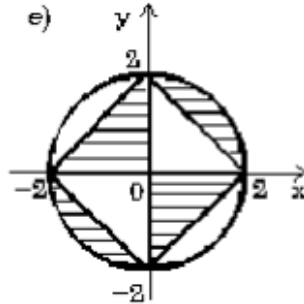
Гр. ІС-02, Плостак Ілля Михайлович

Прийняв:

Асистент Лебідь Сергій Олександрович

1. Умова

26. Задані дійсні числа x , y . Визначити, чи належить точка з координатами (x, y) заштрихованій частині площини:



2. Псевдокод

Функция ПринадлижатЛиКоординатыГрафику(Координата_X, Координата_Y) :

Проверка условие11

Проверка условие12

Проверка условие13

Проверка условие21

Проверка условие22

Проверка условие23

Проверка условие31

Проверка условие32

Проверка условие33

Если условия **соблюдаются**:

Вернуть Правда

Иначе:

Вернуть Ложь

Начало:

Вывод: Проверим, принадлежит ли точка А данной области

Дробное число Координата_X = ВводДробногоЧисла

Дробное число Координата_Y = ВводДробногоЧисла

Если ПринадлижатЛиКоординатыГрафику(Координата_X, Координата_Y) :

Вывод: Точка А лежит в данной области

Иначе:

Вывод: Точка А не лежит в данной области

Конец.

3. Виконання на мові C++

```
#include <stdio.h>
#include <iostream>
#include <limits>
#include <locale.h>
#include <math.h>

using namespace std;

double RoundTo(double Number, int DecimalPlace)
{
    double Rounded = round(Number * pow(10, DecimalPlace)) / pow(10, DecimalPlace);
    return Rounded;
}

float GetDouble(const char PromptMessage[], const char FailMessage[])
{
    float d_Number;

    while (true)
    {
        cout << PromptMessage << "\n";
        cin >> d_Number;

        if (cin.fail())
        {
            cout << FailMessage << "\n";
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }
        else
        {
            return d_Number;
        }
    }
}

int GetInt(const char PromptMessage[], const char FailMessage[])
{
    float f_Number;

    while (true)
    {
        cout << PromptMessage << "\n";
        cin >> f_Number;

        if (cin.fail())
        {
            cout << FailMessage << "\n";
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }
        else
        {
            return round(f_Number);
        }
    }
}
```

```

bool IsCoordsInFunction(double x, double y)
{
    x = RoundTo(x, 4);
    y = RoundTo(y, 4);

    bool c11 = x - y >= -2.0 ? true : false;
    bool c12 = y >= 0.0 ? true : false;
    bool c13 = x <= 0.0 ? true : false;

    bool c21 = x - y <= 2.0 ? true : false;
    bool c22 = y <= 0.0 ? true : false;
    bool c23 = x >= 0.0 ? true : false;

    bool c31 = pow(x, 2) + pow(y, 2) <= pow(2.0, 2) ? true : false;
    bool c32 = x + y >= 2.0 ? true : false;
    bool c33 = x + y <= -2.0 ? true : false;

    if ((c11 && c12 && c13) || (c21 && c22 && c23) || (c31 && (c32 || c33)))
        return true;
    else
        return false;
}

void CreateGraph(int Borders, double Scale)
{
    int Lenght = Borders / Scale;

    cout << "Вот график функции в масштабе " << Scale << ":\n";

    for (int i = -Lenght - 1; i <= Lenght + 1; i++)
    {
        if (i == 0)
        {
            printf("%d %s %d\n", -Borders, string(2 * Lenght + 1, '-').c_str(),
Borders);
        }
        else
        {
            printf(" ");
            for (int j = -Lenght; j <= Lenght + 1; j++)
            {
                double CoordX = double(j) * Scale;
                double CoordY = -double(i) * Scale;

                if (j == 0)
                {
                    if (i == -Lenght - 1)
                        printf("%d", Borders);
                    else if (i == Lenght + 1)
                        printf("%d", -Borders);
                    else
                        printf("|");
                }
                else if (IsCoordsInFunction(CoordX, CoordY))
                {
                    printf(".");
                }
                else
                {
                    printf(" ");
                }
            }
        }
    }
}

```

```

    }
    printf("\n");
}
}
}

int main()
{
    setlocale(LC_ALL, "Russian");

    /*int Borders = GetInt("Введите край графика:", "Введите край графика корректно!");

    while (Borders <= 0)
    {
        if (Borders <= 0)
        {
            cout << "Введите край графика корректно!" << endl;
        }
        Borders = GetInt("Введите край графика:", "Введите край графика корректно!");
    }*/

    cout << "Задача: Проверить, принадлежит ли точка А области, которая приведена
ниже.\n";

    int Borders = 2;

    double Scale = GetDouble("Введите масштаб отображения области десятичной дробью
(рекомендуемый масштаб: 0.1):", "Введите масштаб отображения области корректно!");

    while (Scale > 1 || Scale <= 0)
    {
        if (Scale > 1 || Scale <= 0)
        {
            cout << "Введите масштаб графика корректно!" << endl;
        }
        Scale = GetDouble("Введите масштаб отображения области десятичной дробью:",
"Введите масштаб отображения области корректно!");
    }

    CreateGraph(Borders, Scale);

    cout << "Проверим, принадлежит ли точка А данной области. ";

    double x = GetDouble("Введите значение координаты x:", "Значение задано
неправильно!");
    double y = GetDouble("Введите значение координаты y:", "Значение задано
неправильно!");

    if (IsCoordsInFunction(x, y))
        cout << "Точка А лежит в данной области." << endl;
    else
        cout << "Точка А не лежит в данной области." << endl;

    system("pause");
    return 0;
}

```

4. Результат выполнения програми

Задача: Проверить, принадлежит ли точка А области, которая приведена ниже.
Введите масштаб отображения области десятичной дробью (рекомендуемый масштаб: 0.1):
0.1

Вот график функции в масштабе 0.1:



Проверим, принадлежит ли точка А данной области. Введите значение координаты x:
0.1

Введите значение координаты y:

1.9

Точка А лежит в данной области.

Press any key to continue . . .