МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 1-40 01 01 «Программное обеспечение информационных технологий»

**РЕФЕРАТ**

на тему:

«Графики функций»

Выполнил студент Iкурса 4 группы специальности ПОИТ Точило О.В.

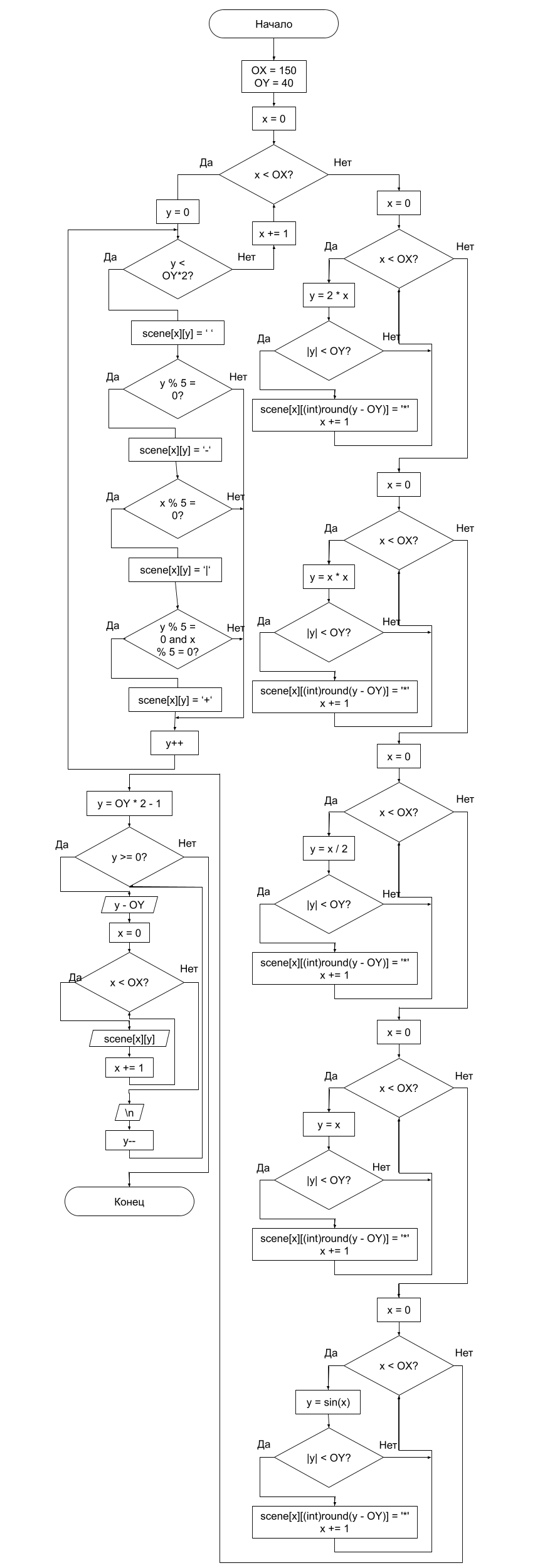
(Ф.И.О)

Руководитель *доц.* Белодед Н. И.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Минск 2021

Задача: создать программу, выводящую при помощи символов в консоль систему координат с сеткой и графиками функций.

Алгоритм:

Псевдокод:

НАЧАЛО

НАЧАЛО ЦИКЛА

ДЛЯ x от 0 до OX – 1 с шагом 1

ПОВТОРЯТЬ

НАЧАЛО ЦИКЛА

ДЛЯ y от 0 до 2 \* OY - 1 с шагом 1

ПОВТОРЯТЬ

ПРИСВОИТЬ scene[x][y] = ' '

ЕСЛИ y % 5 = 0

ПРИСВОИТЬ scene[x][y] = '-'

ЕСЛИ x % 5 = 0

ПРИСВОИТЬ scene[x][y] = '|'

ЕСЛИ y % 5 = 0 и x % 5 = 0

ПРИСВОИТЬ scene[x][y] = '+'

КОНЕЦ ЦИКЛА

КОНЕЦ ЦИКЛА

НАЧАЛО ЦИКЛА

ДЛЯ x от 0 до OX – 1 с шагом 1

ПОВТОРЯТЬ

ПРИСВОИТЬ y = 2 \* x

ЕСЛИ |y| < OY

ПРИСВОИТЬ scene[x][(int)round(y - OY)] = '\*'

КОНЕЦ ЦИКЛА

НАЧАЛО ЦИКЛА

ДЛЯ x от 0 до OX – 1 с шагом 1

ПОВТОРЯТЬ

ПРИСВОИТЬ y = x \* x

ЕСЛИ |y| < OY

ПРИСВОИТЬ scene[x][(int)round(y - OY)] = '\*'

КОНЕЦ ЦИКЛА

НАЧАЛО ЦИКЛА

ДЛЯ x от 0 до OX – 1 с шагом 1

ПОВТОРЯТЬ

ПРИСВОИТЬ y = x / 2

ЕСЛИ |y| < OY

ПРИСВОИТЬ scene[x][(int)round(y - OY)] = '\*'

КОНЕЦ ЦИКЛА

НАЧАЛО ЦИКЛА

ДЛЯ x от 0 до OX – 1 с шагом 1

ПОВТОРЯТЬ

ПРИСВОИТЬ y = x

ЕСЛИ |y| < OY

ПРИСВОИТЬ scene[x][(int)round(y - OY)] = '\*'

КОНЕЦ ЦИКЛА

НАЧАЛО ЦИКЛА

ДЛЯ x от 0 до OX – 1 с шагом 1

ПОВТОРЯТЬ

ПРИСВОИТЬ y = sin(x)

ЕСЛИ |y| < OY

ПРИСВОИТЬ scene[x][(int)round(y - OY)] = '\*'

КОНЕЦ ЦИКЛА

НАЧАЛО ЦИКЛА

ДЛЯ y от OY \* 2 – 1 до 0 с шагом -1

ПОВТОРЯТЬ

ВЫВОД y – OY

НАЧАЛО ЦИКЛА

ДЛЯ x от 0 до OX – 1 с шагом 1

ПОВТОРЯТЬ

ВЫВОД scene[x][y]

КОНЕЦ ЦИКЛА

ВЫВОД “\n”

КОНЕЦ ЦИКЛА

КОНЕЦ

Код:

#include <iostream>

#include <Windows.h>

#include <conio.h>

#define OX 200 //задание размеров поля

#define OY 50

int main()

{

char scene[OX][OY \* 2]; //определение двумерного массива

for (int x = 0; x < OX; x++) {

for (int y = 0; y < OY \* 2; y++)

{

scene[x][y] = ' ';

if (y % 5 == 0)

{

scene[x][y] = '-';

}

if (x % 5 == 0)

{

scene[x][y] = '|';

}

if (x % 5 == 0 and y % 5 == 0)

{

scene[x][y] = '+';

}

}

}

for (int x = 0; x < OX; x++) {

double y = 2 \* x;

if (abs(y) < OY) //проверка на нахождение точки внутри поля

scene[x][(int)round(y - OY)] = '\*';

}

for (int x = 0; x < OX; x++) {

double y = x \* x;

if (abs(y) < OY)

scene[x][(int)round(y - OY)] = '\*';

}

for (int x = 0; x < OX; x++) {

double y = x / 2;

if (abs(y) < OY)

scene[x][(int)round(y - OY)] = '\*';

}

for (int x = 0; x < OX; x++) {

double y = x;

if (abs(y) < OY)

scene[x][(int)round(y - OY)] = '\*';

}

for (int x = 0; x < OX; x++) {

double y = sin(x);

if (abs(y) < OY)

scene[x][(int)round(y - OY)] = '\*';

}

for (int y = OY \* 2 - 1; y >= 0; y--) {

printf("%03d ", y - OY); //вывод значений игрека слева от оси ординат

for (int x = 0; x < OX; x++) {

printf("%c", scene[x][y]);

}

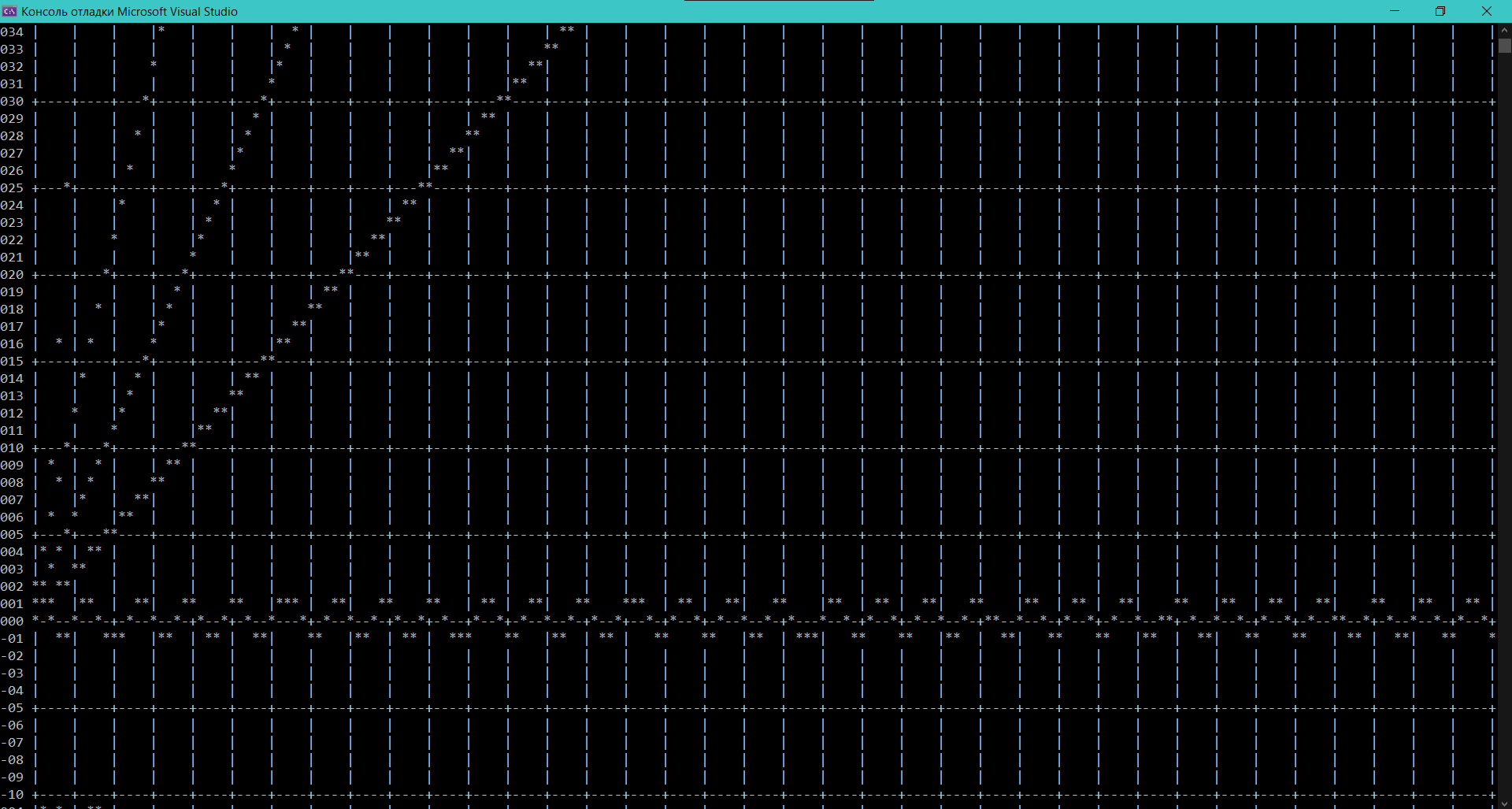
printf("\n");

}

return 0;

}

Скриншот результата:



Вывод: таким образом, реализация программы, выполняющей данную задачу, возможна с использованием двумерного массива.