Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский государственный технический университет имени Н.Э. Баумана»

(национальный исследовательский университет)

Московский техникум космического приборостроения

09.02.07 Информационные системы и программирование

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №4**

по МДК 04.01. Внедрение и поддержка компьютерных систем

Тема: «Табличный процессор»

|  |  |
| --- | --- |
| Выполнил студент | П.Р. Симонян |
| Курс \_\_\_\_\_\_3\_\_\_\_\_\_Группа\_\_\_\_\_ТИП-51\_\_\_\_ |  |
| Подпись студента\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
|  |  |
| Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | С.В. Гончаренко |

Москва 2021

ПРАКТИЧЕСКАЯ РАБОТА 4

Измерение и анализ эксплуатационных характеристик качества программного обеспечения

Время выполнения: 2 ч.

Материально-техническое обеспечение: персональный компьютер (системный блок, монитор, клавиатура, мышь)

Программное обеспечение: ОС Windows, интегрированная среда разработки, офисные программы Microsoft Office.

Учебно-методическое обеспечение: конспект лекций, описание работы, встроенная справочная система ОС Windows и сред разработки.

Порядок выполнения практической работы: изучить теоретический материал, выполнить задания, составить отчёт с выводами о проделанной работе, ответить на контрольные вопросы, сдать выполненную работу и отчёт.

Требования к отчёту: отчёт должен быть оформлен в соответствии с «ГОСТ 19.XXX ЕСПД».

Цель: приобретение навыков обнаружения вируса и устранения последствий его влияния.

Содержание

[1. Программа 4](#_Toc88518786)

[2. Анализ кода 6](#_Toc88518787)

[Количество строк кода (операций) 6](#_Toc88518788)

[Метрики Холстеда 6](#_Toc88518789)

# Программа

Была написана консольная программа для отрисовки цветных линий с помощью алгоритма Брезенхэма. Программа использует написанные самостоятельно на языке C# модули Geometry и Canvas, а также файл самой программы Program.cs.

Листинг 1.1 файла Program.cs:

using System;

using VKS.Geometry;

namespace VKS

{

class Program

{

static void Main()

{

Console.CursorVisible = false;

Point a = new(51, 25);

Point b = new(40, 10);

Canvas.DrawLine(a, b, ConsoleColor.Red);

Console.ReadKey();

}

}

}

Листинг 1.2 файла Geometry.cs:

namespace VKS.Geometry

{

class Point

{

public float X, Y;

public Point(float x, float y)

{

X = x;

Y = y;

}

}

}

Листинг 1.3 файла Canvas.cs:

using System;

using VKS.Geometry;

using static System.Console;

static class Canvas

{

static Point dimensions = new Point(WindowWidth, WindowHeight); //Размер окна вывода консоли

static public void DrawLine(Point start, Point end, ConsoleColor color) //Алгоритм Брезенхэма

{

float x0 = start.X, x1 = end.X, y0 = start.Y, y1 = end.Y;

float dx = Math.Abs(x1 - x0), sx = x0 < x1 ? 1 : -1;

float dy = Math.Abs(y1 - y0), sy = y0 < y1 ? 1 : -1;

float err = (dx > dy ? dx : -dy) / 2, e2;

Point p;

for(; ; )

{

p = new(x0, y0);

DrawPoint(p, color);

if(x0 == x1 && y0 == y1) break;

e2 = err;

if(e2 > -dx) { err -= dy \* 0.5f; x0 += sx; }

if(e2 < dy) { err += dx \* 0.5f; y0 += sy; }

}

}

static public void DrawPoint(Point p, ConsoleColor color)

{

if(p.X > dimensions.X || p.Y > dimensions.Y || p.X < 0 || p.Y < 0)

throw new Exception($"Точка отрисовки выходит за пределы экрана консоли ({WindowWidth};{WindowHeight}).");

SetConsoleCursor(p);

BackgroundColor = color;

Write(' ');

}

static public void SetConsoleCursor(Point point)

{

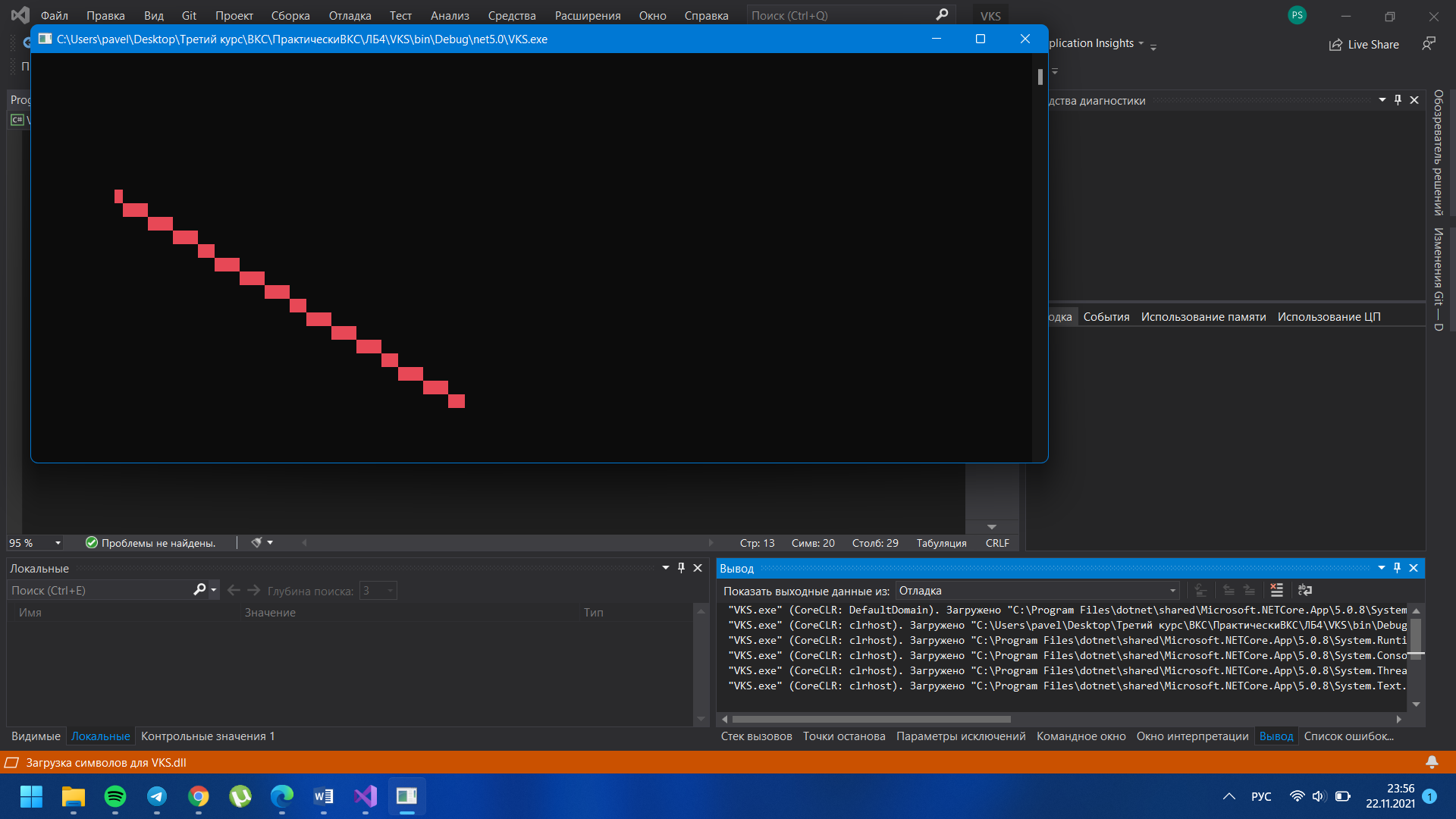
CursorLeft = (int)(point.X);

CursorTop = (int)point.Y;

}

}

Пример работы программы на рисунке 1.1:



# Анализ кода

Написанная программа была проанализирована по нескольким метрикам.

## Количество строк кода (операций)

29

## Метрики Холстеда

Объем:

Функция VKS.Main: 73

Конструктор VKS.Point: 19

Функция VKS.Canvas.SetConsoleCursor: 61

Функция DrawPoint: 235

Функция DrawLine: 585

Словарь = 11

Длина = примерно 182

## Цикломатическое число Мак-Кейба

Вся программа: 17

Функция VKS.Main: 1

Конструктор VKS.Point: 1

Функция VKS.Canvas.SetConsoleCursor: 1

Функция DrawPoint: 5

Функция DrawLine: 9

## Метрика Джилба

Условных операторов во всем проекте: 4

## Метрика обращения к глобальным переменным

Модуль Canvas читает размеры окна для определения его границ.

P = 2 \* 1 = 2;

A = 2;

R = A / P = 2 / 2 = 1;

## Спен-метрика

Функция VKS.Main: 2

Конструктор VKS.Point: 1

Функция VKS.Canvas.SetConsoleCursor: 1

Функция DrawPoint: 3

Функция DrawLine: 5 (без учета цикла, длина которого не определена явно)

Итого для всей программы примерно: (2 + 1 + 1 + 3 + 5) / 5 = 2.4

## Метрика Чепина

P: {a.x, b.x, a.y, b.y, color} = 5 (длина)

M: {x0, x1, y0, y1, dx, dy, err, e2, sx, sy, px, py} = 12

C: {WindowWidth, WindowHeight } - 2

T: {} – 0

Q = P + 2 \* M + 3 \* C + 0.5 \* T = 5 + 24 + 6 + 0 = 35

## Метрика уровня комментированности программы

F = 2/29 = 0.069;

## Метрика Кафура

L = length \* (fan\_in \* fan\_out) = (Берем метрику объема Холстеда) 73 \* 1 \* 1 = 73

Функция VKS.Main: 219

Конструктор VKS.Point: 38

Функция VKS.Canvas.SetConsoleCursor: 122

Функция DrawPoint: 470

Функция DrawLine: 585

## Метрика корректности программы по Холстеду

Примерно 27.65

## Метрика «Интеллектуальное содержание конкретного алгоритма, инвариантное по отношению к используемым языкам реализации»

L^·V = 489

## Метрика необходимых интеллектуальных усилий по написанию программы

E` = N \* log2(n/L) = 144

## Метрика флуктуации длин программной документации

Для программы не составлялась документация. Вычислить метрику флуктуации длин программной документации невозможно.

# Расчет теоретической длины программы

Используя формулу Ñ = (n1 × log2 n1) + (n2×log2 × n2), можно рассчитать теоретическую длину программы.

(9 \* log2(9)) + (4 \* log2(4)) = 36.53