Пензенский государственный технологический университет

Факультет автоматизированных информационных технологий

Кафедра «Прикладная информатика»

ЕН.Ф.02

**КЛАССЫ И ОБЪЕКТЫ. ИНКАПСУЛЯЦИЯ**

Отчет о лабораторной работе № 3

по дисциплине «Информатика и программирование»

Выполнил: ст-т гр.17СН1с

Тюняев А.В.

Проверил: ст. преп. каф. ПИ

Юранов В.С.

2018

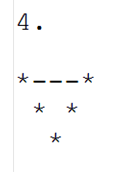
**1 Формулировка задачи**

1. Создать удаленный репозиторий для нового проекта.
2. Создать локальный репозиторий с проектом консольного приложения на языке C# в среде Visual Studio
3. В рамках консольного приложения необходимо создать класс, описывающий точку на плоскости.
4. Поля класса сделать закрытыми.
5. Сделать обращение к полям через функцию.
6. Поля инициализировать после создания объекта. Конструктор оставить по умолчанию (пустой).
7. Добавить в класс функцию для вывода точки на экран консоли(Draw())
8. Добавить в функцию Draw() параметры координат, т.е. Draw(int x, int y), описать в функции Draw передачу значения параметров соответствующим полям класса.
9. В теле главного метода Main создать объект класса точки, пользуясь методом Draw вывести на экран символьный рисунок согласно варианту.
10. Проиндексировать файлы проекта системой git. Сделать коммит изменений и отправить изменения в удаленный репозиторий.

**2 Требования к программе**

Программа должна выводить символы на экран, которые будут образовывать фигуру, необходимую фигура указана в варианты задания.

Поля класса должны быть закрытыми.



Согласно варианту задания, необходимо вывести данную фигуру.

**3 Описание программы**

**3.1 Общие сведения**

Программа разработана в среде Visual Studio 2017

**3.2 Функциональное назначение**

Программа предназначена для вывода символов на экран.

* 1. **Описание логической структуры**

*Класс Point:*

В классе объявляются три переменные, значение оси x, значение оси y, символ который будет выводится.

Методы SetX, SetY, SetSym используются для задания значений x,y,sym с проверкой (если значение меньше 0, присвоиться 0, т.к. на оси для вывода в консоль нет отрицательных значений)

Метод Draw служит для вывода символов на экран.

*Класс Program:*

Метод Main (основной метод) в нем происходит объявление переменной класса Point, передаются значения с координатами точек и символов, которые необходимо вывести в методы SetX, SetY, SetSym класса Point, вызывается метод p.Draw() класса Point для вывода символов в консоль.

**3 Описание применения**

После запуска программы выводятся символы в заданной последовательности образуя фигуру.

int x;

int y;

char sym;

SetX(int x)

(x >= 0)

this.x = x;

this.x = 0;

SetY(int y)

this.y = 0;

this.y = y;

(y >= 0)

SetSym(char symbol)

sym = symbol;

boo1 = true

text[i] == ',' && text[i+1] == '-'

Draw()

txt(str1);

txt(str1);

a == true

str2 = Такой последовательности знаком нет

str1 = Такая последовательность знаков есть

a = check(text)

text

text = sr.ReadToEnd()

Console.SetCursorPosition(x, y);

Console.Write(sym);

Point p = new Point();

p.SetX(0);

p.SetY(0);

p.SetSym('\*');

p.Draw();

p.SetY(0);

p.Draw();

p.SetSym('\*');

p.SetX(1);

p.SetX(2);

p.Draw();

p.SetY(0);

p.SetSym('\*');

p.Draw();

p.SetSym('\*');

p.SetY(0);

p.SetX(3);

p.Draw();

p.SetSym('\*');

p.SetY(0);

p.SetX(4);

p.SetY(1);

p.SetSym('\*');

p.Draw();

p.SetX(1);

p.SetX(2);

p.SetY(1);

p.SetSym('\*');

p.Draw();

p.SetX(3);

p.SetY(1);

p.SetSym('\*');

p.Draw();

p.SetX(2);

p.SetY(2);

p.SetSym('\*');

p.Draw();

ТЕКСТ ПРОГРАММЫ

Приложение А

(обязательное)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

class Point

{

int x;

int y;

char sym;

public void SetX(int x)

{

if (x >= 0)

this.x = x;

else

this.x = 0;

}

public void SetY(int y)

{

if (y >= 0)

this.y = y;

else

this.y = 0;

}

public void SetSym(char symbol)

{

sym = symbol;

}

public void Draw()

{

Console.SetCursorPosition(x, y);

Console.Write(sym);

}

}

class Program

{

static void Main(string[] args)

{

Point p = new Point();

p.SetX(0);

p.SetY(0);

p.SetSym('\*');

p.Draw();

p.SetX(1);

p.SetY(0);

p.SetSym('-');

p.Draw();

p.SetX(2);

p.SetY(0);

p.SetSym('-');

p.Draw();

p.SetX(3);

p.SetY(0);

p.SetSym('-');

p.Draw();

p.SetX(4);

p.SetY(0);

p.SetSym('\*');

p.Draw();

p.SetX(1);

p.SetY(1);

p.SetSym('\*');

p.Draw();

p.SetX(2);

p.SetY(1);

p.SetSym(' ');

p.Draw();

p.SetX(3);

p.SetY(1);

p.SetSym('\*');

p.Draw();

p.SetX(2);

p.SetY(2);

p.SetSym('\*');

p.Draw();

Console.ReadKey();

}

}

}

РЕЗУЛЬТАТЫ ИСПЫТАНИЙ

Приложение Б

(обязательное)

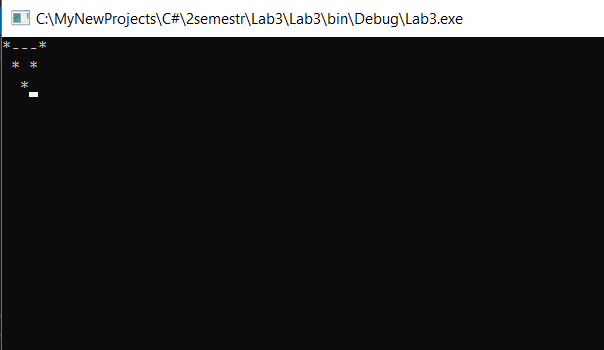


Рисунок 1 – Результат выполнения программы