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-ты гр.17СН1с

Тюняев А.В. Богданов М.О.

Проверил: ст. преп. каф. ПИ

Юранов В.С.

2018

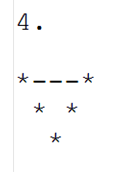
**1 Формулировка задачи**

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 поля, 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 объекта 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.

Выводы

В ходе выполнения лабораторной работы было сформулировано и утверждено техническое задание, разработан алгоритм решения задачи по выводу символов на экран.

**Контрольные вопросы**

1. **Что такое класс?**

Класс - множество объектов, связанных общностью структуры и поведением.

В объектно-ориентированном программировании (ООП) – **класс** это основной элемент, в рамках которого осуществляется конструирование программ. Класс содержит в себе данные и код, который управляет этими данными.

Класс зачастую описывает объект реального мира. Как и реальный объект, класс содержит свой набор параметров и характеристик. Каждый такой параметр называется **поле класса**. Также класс способен манипулировать своими характеристиками (полями) с помощью **методов класса**.

1. **Что такое объект?**

**Объе́кт** — некоторая сущность в компьютерном пространстве, обладающая определённым состоянием и поведением, имеющая заданные значения свойств ([атрибутов](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B5_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0)) и операций над ними ([методов](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)))[[1]](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)#cite_note-def-1). Как правило, при рассмотрении объектов выделяется то, что объекты принадлежат одному или нескольким [классам](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), которые определяют поведение (являются моделью) объекта. Термины «**экземпляр класса**» и «объект» взаимозаменяемы.

1. **Как связаны между собой классы и объекты в программе?**
2. **Что такое инкапсуляция?**

Суть же понятия "инкапсуляция" заключается в манипулировании модификаторами доступа. Это означает, что разработчик сам решает, какие свойства, методы и классы будут открыты классу-клиенту, а какие - скрыты.

1. **За счет чего реализуется защита от несанкционированного доступа к данным?**

За счет инкапсуляции, скрытия части данных, что бы их нельзя было изменить

1. **Чем отличаются поля от переменных?**

Поле является переменной любого типа, которая объявлена непосредственно в классе или структуре.

Поле - эта та же переменная, но являющаяся частью структуры / класса.

Переменная может существовать без класса, поле – нет.

Как правило, следует использовать поля только для переменных, являющихся закрытыми или защищенными

1. **Что такое свойство?**

Свойство — способ доступа к внутреннему состоянию объекта. Обращение к свойству объекта выглядит так же, как и обращение к структурному полю (в структурном программировании), но, в действительности, реализовано через вызов функции.

При попытке задать значение данного свойства вызывается один метод, а при попытке получить значение данного свойства — другой.

1. **Какие методы есть у свойства?**

Существует два таких метода: get (для получения данных) и set (для записи).

1. **Что делают эти методы?**

get (для получения данных) и set (для записи).

1. **Что такое точка входа?**

Место с которого начинает выполняться программа

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)

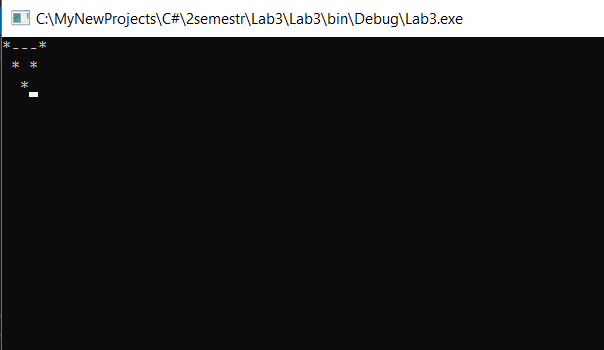


Рисунок 1 – Результат выполнения программы