Пензенский государственный технологический университет

Факультет автоматизированных информационных технологий

Кафедра «Прикладная информатика»

ЕН.Ф.02

**КОНСТРУКТОРЫ, НАСЛЕДОВАНИЕ**

Отчет о лабораторной работе № 4

по дисциплине «Информатика и программирование»

Выполнил: ст-т гр.17СН1с

Тюняев А.В., Богданов М.О.

Проверил: ст. преп. каф. ПИ

Юранов В.С.

2018

**1 Формулировка задачи**

1. В приложение из 1 лабораторной работы необходимо добавить классы фигуры и соответствующие классы наследники – горизонтальная линия и вертикальная линия.
2. В основной программе создать объекты данных классов, которые формируют на экран прямоугольник.
3. Сделать коммит изменений. Отправить изменения в удаленный репозиторий

**2 Требования к программе**

Программа должна выводить символы на экран, которые будут образовывать прямоугольник.



**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.

Конструктор Point ….

Метод Draw служит для вывода символов на экран.

*Класс Figure:*

Создается список pList для записи координат точек

Метод Draw служит для вывода символов на экран используя список pList.

*Класс VerticalLine:*

Этот класс является наследником класса Figure.

Метод VerticalLine принимает значения смещения по оси y, координату по оси x и символ.

Создается экземпляр класса List<Point> (списка).

В цикл передаются значения начальной и конечной точки по оси y.

В цикле создается экземпляр класса Point и в список добавляются координаты точек по осям x и y, и необходимый символ.

*Класс HorizontalLine:*

Этот класс является наследником класса Figure.

Метод HorizontalLine принимает значения смещения по оси x, координату по оси y и символ.

Создается экземпляр класса List<Point> (списка).

В цикл передаются значения начальной и конечной точки по оси x.

В цикле создается экземпляр класса Point и в список добавляются координаты точек по осям x и y, и необходимый символ.

*Класс Program:*

Метод Main (основной метод) в нем создается экземпляры классов VerticalLine и HorizontalLine, передаются данные в эти классы, вызывается метод Draw, для вывода рисунка.

**3 Описание применения**

После запуска программы выводятся символы в заданной последовательности образуя прямоугольник.

**Выводы**

Познакомились с созданием конструкторовдля классов. Изучили механизмы наследования и полиморфизма. Познакомились с управляющими операторами языка C#.

**Контрольные вопросы**

1. Что такое конструктор?
2. Зачем нужен конструктор?
3. Что такое наследование?
4. Что такое методы и поля класса?
5. Что такое модификаторы доступа и для чего они нужны?

ТЕКСТ ПРОГРАММЫ

Приложение А

(обязательное)

namespace Lab4

{

class Point

{

int x;

int y;

char sym;

public Point(int x, int y, char sym)

{

this.x = x;

this.y = y;

this.sym = sym;

}

public void Draw()

{

Console.SetCursorPosition(x, y);

Console.Write(sym);

}

}

class Figure

{

protected List<Point> pList;

public void Draw()

{

foreach (Point p in pList)

{

p.Draw();

}

}

}

class VerticalLine : Figure

{

public VerticalLine(int yUp, int yDown, int x, char sym)

{

pList = new List<Point>();

for (int y = yUp; y <= yDown; y++)

{

Point p = new Point(x, y, sym);

pList.Add(p);

}

}

}

class HorizontalLine : Figure

{

public HorizontalLine(int xLeft, int xRight, int y, char sym)

{

pList = new List<Point>();

for (int x = xLeft; x <= xRight; x++)

{

Point p = new Point(x, y, sym);

pList.Add(p);

}

}

}

class Program

{

static void Main(string[] args)

{

VerticalLine v1 = new VerticalLine(3, 4, 1, '\*');

v1.Draw();

HorizontalLine h1 = new HorizontalLine(1, 10, 2, '\*');

h1.Draw();

//VerticalLine v1 = new VerticalLine(3, 4, 1, '\*');

//v1.Draw();

VerticalLine v2 = new VerticalLine(3, 4, 10, '\*');

v2.Draw();

HorizontalLine h2 = new HorizontalLine(1, 10, 5, '\*');

h2.Draw();

Console.ReadKey();

}

}

}

РЕЗУЛЬТАТЫ ИСПЫТАНИЙ

Приложение Б

(обязательное)

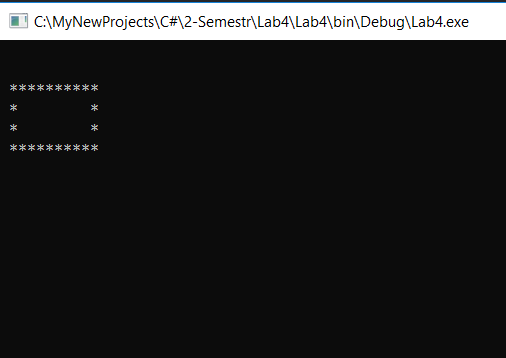


Рисунок 1 – Результат выполнения программы