#### Московский государственный технический университет имени Н.Э. Баумана

## Лабораторная работа по дисциплине БКИТ №3

*Выполнил: Ханмурзин Тагир, ИУ5-34, 24.10.2017*

**1. Описание задания**

Разработать программу, реализующую работу с коллекциями.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Создать объекты классов «Прямоугольник», «Квадрат», «Круг».
3. Для реализации возможности сортировки геометрических фигур для класса «Геометрическая фигура» добавить реализацию интерфейса IComparable. Сортировка производится по площади фигуры.
4. Создать коллекцию класса ArrayList. Сохранить объекты в коллекцию. Отсортировать коллекцию. Вывести в цикле содержимое коллекции.
5. Создать коллекцию класса List<Figure>. Сохранить объекты в коллекцию. Отсортировать коллекцию. Вывести в цикле содержимое коллекции.
6. Модифицировать класс разреженной матрицы (проект SparseMatrix) для работы с тремя измерениями – x,y,z. Вывод элементов в методе ToString() осуществлять в том виде, который Вы считаете наиболее удобным. Разработать пример использования разреженной матрицы для геометрических фигур.
7. Реализовать класс «SimpleStack» на основе односвязного списка. Класс SimpleStack наследуется от класса SimpleList (разобранного в пособии). Необходимо добавить в класс методы:
   * public void Push(T element) – добавление в стек;
   * public T Pop() – чтение с удалением из стека.
8. Пример работы класса SimpleStack реализовать на основе геометрических фигур.

**2. Листинг программного кода ЗАМЕНИТЬ**

using System;

namespace ProjectLaba

{

    interface IPrint

    {

        void Print();

    }

    public abstract class GeometricFigure

    {

        public virtual double getS() // Получить площадь данной фигуры

        {

            return 0;

        }

    }

    public class Rectangle : GeometricFigure, IPrint // Прямоугольник

    {

        double width;

        double height;

        public Rectangle(double \_width, double \_height)

        {

            this.width = \_width;

            this.height = \_height;

        }

        public override double getS()

        {

            return width\*height;

        }

        public virtual void Print()

        {

            Console.Write(ToString());

        }

        public override string ToString()

        {

            return "Rectangle: Area = " + getS();

        }

    }

    public class Square : Rectangle // Квадрат

    {

        private double side;

        public Square(double \_side) : base(\_side, \_side)

        {

            this.side = \_side; // Бесмыслица

        }

        public override void Print()

        {

            Console.Write(ToString());

        }

        public override string ToString()

        {

            return "Square: Area = " + getS();

        }

    }

    public class Circle : GeometricFigure // Круг

    {

        private double radius;

        public Circle(double \_radius)

        {

            this.radius = \_radius;

        }

        public override double getS()

        {

            return Math.PI\*radius\*radius; //Считать площадь

        }

        public virtual void Print()

        {

            Console.Write(ToString());

        }

        public override string ToString()

        {

            return "Circle: Area = " + getS();

        }

    }

    public class Program

    {

        static double inputVal(string prompt)

        {

            double a = 0;

            do

                Console.Write(prompt);

            while (!double.TryParse(Console.ReadLine(), out a));

            return a;

        }

        static int inputInt()

        {

            int a = 0;

            int.TryParse(Console.ReadLine(), out a);

            return a;

        }

        static void Main()

        {

            double param1, param2;

            int select;

            bool work;

            do

            {

                Console.Clear();

                Console.Write("Выберите фигуру:\n1 | Прямоугольник\n2 | Квадрат\n3 | Круг\n");

                select = inputInt();

                switch(select)

                {

                    case 1:

                        Console.Clear();

                        param1 = inputVal("Enter width: ");

                        param2 = inputVal("Enter height: ");

                        Rectangle rectangle = new Rectangle(param1, param2);

                        rectangle.Print();

                        break;

                    case 2:

                        Console.Clear();

                        param1 = inputVal("Enter side: ");

                        Square square = new Square(param1);

                        square.Print();

                        break;

                    case 3:

                        Console.Clear();

                        param1 = inputVal("Enter radius: ");

                        Circle circle = new Circle(param1);

                        circle.Print();

                        break;

                }

                Console.Write("\nExit? (1/0) \n"); // Можно заменить на человеческое [y/n]

                string s = Console.ReadLine();     // Просто потому что это же C#

                work = (s == "0" || s == "");      // Тогда: work = (s == "n" || s == "");

            } while (work);

        }

    }

}

**3. Тест работы ЗАМЕНИТЬ**

При первом запуске показывается главное меню:

Выберите фигуру:

1 | Прямоугольник

2 | Квадрат

3 | Круг

1. Если выбрать первый пункт, то последовательно будет предложено ввести width и height будущего прямоугольника.

Enter width: 10

Enter height: 20

Rectangle: Area = 200

В конечном итоге будет выведена площадь нашего прямоугольника.

2. Если выбрать второй пункт, то будет предложено ввести side будущего квадрата.

Enter side: 20

Square: Area = 400

В конечном итоге будет выведена площадь нашего квадрата.

3. Если выбрать третий пункт, то будет предложено ввести radius нашего будущего круга.

Enter radius: 10

Circle: Area = 314, 159265358979

В конечно итоге будет выведена площадь нашего круга.

После решения квадратного уравнения программа спрашивает у пользователя:

Exit? (1/0)

1 – выйти из программы

0 – продолжить работу с программой