

Министерство науки и высшего образования Российской Федерации  
Муромский институт (филиал)  
федерального государственного бюджетного образовательного учреждения высшего образования  
«Владимирский государственный университет  
Имени Александра Григорьевича и Николая Григорьевича Столетовых»  
(МИВлГУ)

Факультет \_\_\_\_\_ ИТ  
Кафедра \_\_\_\_\_ ИТР

# Отчет по практике

Практика по получению первичных профессиональных умений и навыков  
(наименование практики)

Руководитель

Быков А.А.

(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Студент ПИН-121

(группа)

Ермилов М.В.

(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Муром 2022

## Содержание:

Введение .....	3
1. Краткая характеристика компании Ред Софт .....	5
2. Краткая характеристика МЗ РИП .....	6
3.Задание.....	7

					МИВУ 09.03.04-21.07			
Из	Лис	№ докум.	Подп.	Дата				
Разраб.		Ермилов М.В.			<p><u>Практика по получению</u>  <u>первичных профессиональных</u>  <u>умений и навыков</u></p>			
Провер.		Быков А.А.						
Н.контр.								
Утв.								
						Лит.	Лист	Листов
							2	22
						МИВУ ПИН - 121		

## Введение

Программная инженерия (промышленное программирование) обычно ассоциируется с разработкой больших и сложных программ коллективами разработчиков. Становление и развитие этой области деятельности было вызвано рядом проблем, связанных с высокой стоимостью программного обеспечения, сложностью его создания, необходимостью управления и прогнозирования процессов разработки.

В конце 60-х – начале 70-х годов прошлого века произошло событие, которое вошло в историю как первый кризис программирования. Событие состояло в том, что стоимость программного обеспечения стала приближаться к стоимости аппаратуры («железа»), а динамика роста этих стоимостей позволяла прогнозировать, что к середине 90-годов все человечество будет заниматься разработкой программ для компьютеров. Тогда и заговорили о программной инженерии (или технологии промышленного программирования, как это называлось в России) как о некоторой дисциплине, целью которой является сокращение стоимости программ. С тех пор программная инженерия прошла достаточно бурное развитие. Этапы развития программной инженерии можно выделять по-разному. Каждый этап связан с появлением (или осознанием) очередной проблемы и нахождением путей и способов решения этой проблемы.

Сам термин – software engineering (программная инженерия) - впервые был озвучен в октябре 1968 года на конференции подкомитета НАТО по науке и технике (г. Гармиш, Германия). Присутствовало 50 профессиональных разработчиков ПО из 11 стран. Рассматривались проблемы проектирования, разработки, распространения и поддержки программ. Там впервые и прозвучал термин «программная инженерия» как некоторая дисциплина, которую надо создавать и которой надо руководствоваться в решении перечисленных проблем. Вскоре после этого в Лондоне состоялась встреча 22-х руководителей проектов по разработке ПО. На встрече анализировались проблемы и перспективы развития ПО. Отмечалась возрастающее воздействие ПО на жизнь людей. Впервые серьезно заговорили о надвигающемся кризисе ПО.

Применяющиеся принципы и методы разработки ПО требовали постоянного усовершенствования . Именно на этой встрече была предложена концепция жизненного цикла ПО (SLC – Software Lifetime Cycle) как последовательности шагов-стадий, которые необходимо выполнить в процессе создания и эксплуатации ПО. Вокруг этой концепции было много споров. В 1970 г. У.У. Ройс (W.W. Royce) произвел 5 идентификацию нескольких стадий в типичном цикле и было высказано предположение, что контроль выполнения стадий приведет к повышению качества ПО и сокращению стоимости разработки.

					МИВУ 09.03.04-21.07	Лист
						4
Из	Лист	№ докум.	Подп.	Дата		

## Компания РЕД СОФТ

РЕД СОФТ - отечественный поставщик решений и услуг в области информационных технологий. Компания создает и осуществляет комплексные проекты в области хранения и управления данными на основе собственного промышленного прикладного программного обеспечения.

РЕД СОФТ это эффективная команда, имеющая более чем 15-летний опыт разработки в российском государственном и коммерческом секторе, а также за рубежом. Помимо решений, адаптированных под требования клиентов, компания создала собственную линейку продуктов, построенных на базе ПО с открытым кодом: РЕД ОС, СУБД Ред База Данных и Ред Платформа.

Среди заказчиков РЕД СОФТ числятся более 20 ведомств Российской Федерации, а также крупные частные компании.

РЕД СОФТ активно расширяет свою партнерскую сеть. Подписаны соглашения о технологическом партнерстве с рядом ведущих отраслевых компаний. Регулярно проводятся тестирования на совместимость продуктов, а также обмен мнениями и опытом.

					МИВУ 09.03.04-21.07	Лист
						5
Из	Лист	№ докум.	Подп.	Дата		

## **АО «Муромский завод радиоизмерительных приборов» (МЗ РИП)**

Акционерное общество «Муромский завод радиоизмерительных приборов» (АО «МЗ РИП») начинает свою историю с 1949 года и в настоящее время является одним из крупных современных предприятий отрасли.

Основу продукции АО «МЗ РИП» составляют высокоэффективные радиолокационные станции, а так же системы обнаружения воздушных целей и посадки самолетов. В настоящее время предприятие является головным в России по выпуску наземных радиолокаторов обнаружения низколетящих целей. Комплекс услуг, предоставляемых заводом партнерам, включает разработку, изготовление, модернизацию и сопровождение поставляемых систем, оборудования и специальной техники.

Для повышения эффективности своей деятельности и обеспечения спроса на продукцию предприятия на отечественных и зарубежных рынках АО «МЗ РИП» работает в тесном контакте с разработчиками радиолокационной техники и с государственной экспортной компанией АО «Рособоронэкспорт».

АО «МЗ РИП» располагает парком оборудования для различных видов механообработки, переработки пластмасс, развитой базой для проведения различного вида испытаний изделий с воздействием на них механо-климатических факторов.

Муромский завод радиоизмерительных приборов (МЗ РИП) - одно из крупных современных предприятий города Муром и Владимирской области, входит в состав ОАО "Концерн ВПО "Алмаз-Антей", выпускает сложную радиоэлектронную аппаратуру высокого качества, и является головным предприятием по выпуску наземных радиолокаторов обнаружения низколетящих целей. Продукция завода – радиолокационные станции обнаружения воздушных целей, радиолокационные системы посадки самолетов.

## Задание.

```
namespace lab7
{
    public class program
    {
        /// <summary>
        /// Динамическая коллекция фигур
        /// </summary>
        public static List<Figure> Pictures = new List<Figure>();
        static void Main(string[] arg)
        {
            Pictures.Add(new Circle(10));
            Pictures.Add(new Rectangle(10, 20));
            Pictures.Add(new Trapezium(10, 20, 5));
            Pictures[0].setPosition(new Position(10, 20));
            Pictures[1].setPosition(new Position(20, 20));
            Pictures[2].setPosition(new Position(50, 50));
        }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace lab7
{
    /// <summary>
    /// Объект для хранения и работы позиций
    /// </summary>
    public class Position
    {
        /// <summary>
        /// Координата X
        /// </summary>
        public double X;
        /// <summary>
        /// Координата Y
        /// </summary>
        public double Y;
        /// <summary>
        /// Конструктор
        /// </summary>
        /// <param name="x">Координата X</param>
        /// <param name="y">Координата Y</param>
        public Position(double x, double y)
        {
            X = x;
            Y = y;
        }
        /// <summary>
        /// Конструктор, который за X и Y берёт 0
        /// </summary>
        public Position() : this(0, 0) { }

        /// <summary>
        /// Форматирование строкового представления
        /// </summary>
        /// <returns>Сформатированное строковое представление</returns>
        public override string ToString()
        {
            return $"x:{X} y:{Y}";
        }
    }
}
```

Из	Лист	№ докум.	Подп.	Дата

МИВУ 09.03.04-21.07

Лист

7

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace lab7
{
    /// <summary>
    /// Объект для хранения фигуры и работы с ней.
    /// Создан в качестве родителя, чтобы от него наследовались остальные фигуры
    /// </summary>
    public abstract class Figure
    {
        /// <summary>
        /// Площадь фигуры
        /// </summary>
        public virtual double Square
        {
            get { return 0; }
        }
        /// <summary>
        /// Периметр фигуры
        /// </summary>
        public virtual double Perimeter
        {
            get { return 0; }
        }
        /// <summary>
        /// Базовый конструктор
        /// </summary>
        public Figure()
        {
        }

        private protected Position BottomLeft = new Position();

        /// <summary>
        /// Установить позицию фигуры
        /// </summary>
        /// <param name="p">координаты позиции, нижний левый угол</param>
        public void setPosition(Position p) { BottomLeft = p; }

        /// <summary>
        /// Позиция центра фигуры
        /// </summary>
        public virtual Position Center
        {
            get { return null; }
        }

        /// <summary>
        /// Форматирование строкового представления
        /// </summary>
        /// <returns>Сформатированное строковое представление</returns>
        public override string ToString()
        {
            return base.ToString();
        }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace lab7
{

```



```

/// <summary>
/// Объект для хранения и работы с фигурой круг
/// </summary>
public class Circle : Figure
{
    private double radius;
    public override double Square
    {
        get { return Math.PI * Math.Pow(radius, 2); }
    }
    public override double Perimeter
    {
        get { return Math.PI * radius * 2; }
    }
    public override Position Center
    {
        get { return BottomLeft; }
    }
    /// <summary>
    /// Конструктор фигуры круг
    /// </summary>
    /// <param name="Radius">Радиус круга</param>
    public Circle(double Radius)
    {
        radius = Radius;
    }
    public override string ToString()
    {
        return $"Круг: S = {Square}; P = {Perimeter}; position: ({Center})";
    }

    /// <summary>
    /// Установить позицию фигуры
    /// </summary>
    /// <param name="p">координаты позиции, центра</param>
    public void setPosition(Position p) { BottomLeft = p; }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace lab7
{
    /// <summary>
    /// Объект для хранения и работы с фигурой трапеция
    /// </summary>
    public class Trapezium : Figure
    {
        private double widthTop;
        private double widthBottom;
        private double height;
        private double sideRight;
        private double sideLeft;
        public override double Square
        {
            get { return height * (widthTop + widthBottom) / 2; }
        }
        public override double Perimeter
        {
            get { return widthTop + widthBottom + sideRight + sideLeft; }
        }
        public override Position Center
        {
            get { return new Position(BottomLeft.X + widthBottom / 2, BottomLeft.Y +
height / 2); }
        }
    }
}

```

```

    /// <summary>
    /// Конструктор фигуры трапеция
    /// </summary>
    /// <param name="WidthTop">Ширина верхнего основания</param>
    /// <param name="WidthBottom">Ширина нижнего основания</param>
    /// <param name="Height">Высота</param>
    public Trapezium(double WidthTop, double WidthBottom, double Height)
    {
        widthBottom = WidthBottom;
        widthTop = WidthTop;
        height = Height;
        double a = (widthBottom - widthTop) / 2;
        sideLeft = sideRight = Math.Sqrt(Math.Pow(a, 2) + Math.Pow(height, 2));
    }
    public override string ToString()
    {
        return $"Трапеция: S = {Square}; P = {Perimeter}; position: ({Center})";
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace lab7
{
    /// <summary>
    /// Объект для хранения и работы с фигурой прямоугольник
    /// </summary>
    public class Rectangle : Figure
    {
        private double width;
        private double height;
        public override double Square
        {
            get { return width * height; }
        }
        public override double Perimeter
        {
            get { return width * 2 + height * 2; }
        }
        public override Position Center
        {
            get { return new Position(BottomLeft.X + width / 2, BottomLeft.Y + height /
2); }
        }
        /// <summary>
        /// Конструктор прямоугольной фигуры
        /// </summary>
        /// <param name="Width">Ширина прямоугольника</param>
        /// <param name="Height">Длина прямоугольника</param>
        public Rectangle(double Width, double Height)
        {
            width = Width;
            height = Height;
        }
        public override string ToString()
        {
            return $"Прямоугольник: S = {Square}; P = {Perimeter}; position: ({Center})";
        }
    }
}

```

# Namespace lab7

## Classes

### Circle

Объект для хранения и работы с фигурой круг

### Figure

Объект для хранения фигуры и работы с ней. Создан в качестве родителя, чтобы от него наследовались остальные фигуры

### Position

Объект для хранения и работы позиций

### program

### Rectangle

Объект для хранения и работы с фигурой прямоугольник

### Trapezium

Объект для хранения и работы с фигурой трапеция

## Class program

Inheritance

System.Object  
program

Inherited Members

System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: lab7

Assembly: lab7.dll

### Syntax

```
public class program
```

					МИВУ 09.03.04-21.07	Лист
						11
Из	Лист	№ докум.	Подп.	Дата		

# Fields

## Pictures

Динамическая коллекция фигур

### Declaration

```
public static List<Figure> Pictures
```

### Field Value

Type	Description
List<Figure>	

# Class Position

Объект для хранения и работы позиций

### Inheritance

System.Object  
Position

### Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: lab7

Assembly: lab7.dll

### Syntax

```
public class Position
```

# Constructors

## Position()

Конструктор, который за X и Y берёт 0

### Declaration

```
public Position()
```

# Position(Double, Double)

Конструктор

## Declaration

```
public Position(double x, double y)
```

## Parameters

Type	Name	Description
System. Double	<i>x</i>	Координата X
System. Double	<i>y</i>	Координата Y

# Fields

## X

Координата X

## Declaration

```
public double X
```

## Field Value

Type	Description
System. Double	

## Y

Координата Y

## Declaration

```
public double Y
```

## Field Value

Type	Description
System. Double	

# Methods

## ToString()

Форматирование строкового представления

## Declaration

```
public override string ToString()
```

## Returns

Type	Description
System.String	Сформатированное строковое представление

## Overrides

System.Object.ToString()

# Class Figure

Объект для хранения фигуры и работы с ней. Создан в качестве родителя, чтобы от него наследовались остальные фигуры

## Inheritance

System.Object

Figure

Circle

Rectangle

Trapezium

## Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: **lab7**

Assembly: **lab7.dll**

## Syntax

```
public abstract class Figure
```

# Constructors

## Figure()

Базовый конструктор

## Declaration

```
public Figure()
```

# Properties

					МИВУ 09.03.04-21.07	Лист
						14
Из	Лист	№ докум.	Подп.	Дата		

## Center

Позиция центра фигуры

### Declaration

```
public virtual Position Center { get; }
```

### Property Value

Type	Description
Position	

## Perimeter

Периметр фигуры

### Declaration

```
public virtual double Perimeter { get; }
```

### Property Value

Type	Description
System. Double	

## Square

Площадь фигуры

### Declaration

```
public virtual double Square { get; }
```

### Property Value

Type	Description
System. Double	

## Methods

### setPosition(Position)

Установить позицию фигуры

### Declaration

```
public void setPosition(Position p)
```

### Parameters

Type	Name	Description
Position	<i>p</i>	координаты позиции, нижний левый угол

# ToString()

Форматирование строкового представления

## Declaration

```
public override string ToString()
```

## Returns

Type	Description
System. String	Сформатированное строковое представление

## Overrides

System.Object.ToString()

# Class Circle

Объект для хранения и работы с фигурой круг

## Inheritance

System.Object  
Figure  
Circle

## Inherited Members

System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: lab7

Assembly: lab7.dll

## Syntax

```
public class Circle : Figure
```

# Constructors

## Circle(Double)

Конструктор фигуры круг

## Declaration

```
public Circle(double Radius)
```

## Parameters



Type	Name	Description
System. Double	<i>Radius</i>	Радиус круга

# Properties

## Center

### Declaration

```
public override Position Center { get; }
```

### Property Value

Type	Description
Position	

### Overrides

Figure.Center

## Perimeter

### Declaration

```
public override double Perimeter { get; }
```

### Property Value

Type	Description
System. Double	

### Overrides

Figure.Perimeter

## Square

### Declaration

```
public override double Square { get; }
```

### Property Value

Type	Description
System. Double	

### Overrides

Figure.Square

# Methods

## setPosition(Position)

					МИВУ 09.03.04-21.07	Лист
						17
Из	Лист	№ докум.	Подп.	Дата		

Установить позицию фигуры

Declaration

```
public void setPosition(Position p)
```

Parameters

Type	Name	Description
Position	<i>p</i>	координаты позиции, центра

ToString()

Declaration

```
public override string ToString()
```

Returns

Type	Description
System. String	

Overrides

```
Figure.ToString()
```

Class Rectangle

Объект для хранения и работы с фигурой прямоугольник

Inheritance

System.Object  
Figure  
Rectangle

Inherited Members

Figure.setPosition(Position)  
System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: lab7

Assembly: lab7.dll

Syntax

```
public class Rectangle : Figure
```

# Constructors

## Rectangle(Double, Double)

Конструктор прямоугольной фигуры

### Declaration

```
public Rectangle(double Width, double Height)
```

### Parameters

Type	Name	Description
System. Double	<i>Width</i>	Ширина прямоугольника
System. Double	<i>Height</i>	Длинна прямоугольника

# Properties

## Center

### Declaration

```
public override Position Center { get; }
```

### Property Value

Type	Description
<a href="#">Position</a>	

### Overrides

[Figure.Center](#)

## Perimeter

### Declaration

```
public override double Perimeter { get; }
```

### Property Value

Type	Description
System. Double	

### Overrides

[Figure.Perimeter](#)

## Square

### Declaration

```
public override double Square { get; }
```

### Property Value

					МИВУ 09.03.04-21.07	Лист
						19
Из	Лист	№ докум.	Подп.	Дата		

Type	Description
System. Double	

**Overrides**  
[Figure.Square](#)

Methods

ToString()

**Declaration**  
`public override string ToString()`

Returns

Type	Description
System. String	

**Overrides**  
[Figure.ToString\(\)](#)

Class Trapezium

Объект для хранения и работы с фигурой трапеция

Inheritance

System.Object  
    [Figure](#)  
        Trapezium

Inherited Members

[Figure.setPosition\(Position\)](#)  
System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)

**Namespace:** [lab7](#)  
**Assembly:** lab7.dll

**Syntax**  
`public class Trapezium : Figure`

# Constructors

## Trapezium(Double, Double, Double)

Конструктор фигуры трапеция

### Declaration

```
public Trapezium(double WidthTop, double WidthBottom, double Height)
```

### Parameters

Type	Name	Description
System. Double	<i>WidthTop</i>	Ширина верхнего основания
System. Double	<i>WidthBottom</i>	Ширина нижнего основания
System. Double	<i>Height</i>	Высота

## Properties

### Center

#### Declaration

```
public override Position Center { get; }
```

#### Property Value

Type	Description
Position	

#### Overrides

Figure.Center

### Perimeter

#### Declaration

```
public override double Perimeter { get; }
```

#### Property Value

Type	Description
System. Double	

#### Overrides

Figure.Perimeter

### Square

#### Declaration

```
public override double Square { get; }
```

					МИВУ 09.03.04-21.07	Лист
						21
Из	Лист	№ докум.	Подп.	Дата		

Property Value

Type	Description
------	-------------

System. Double	
----------------	--

Overrides

[Figure.Square](#)

Methods

ToString()

Declaration

```
public override string ToString()
```

Returns

Type	Description
------	-------------

System. String	
----------------	--

Overrides

[Figure.ToString\(\)](#)