



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №1

«Базовые средства ООП в языке C#»

ДИСЦИПЛИНА: «Объектно-ориентированное программирование»

Выполнил: студент гр. ИУК4-42Б _____ (Карельский М.К.)
(Подпись)

Проверил: _____ (Былинка М.И.)
(Подпись)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2022

Цель: приобрести навыки реализации классов и их основных элементов на языке C#.

Задачи:

1. Изучить синтаксис создания классов в языке C#
2. Научиться реализовывать классы средствами языка C#.

Вариант №7

Класс: Масса

Состояние (поля): величина массы в граммах

Методы конструирования: создание величины массы, заданной в граммах; в фунтах

Свойства: масса в граммах; в фунтах; в унциях

Операции: сложение и вычитание

Примечание:

1 фунт = 453,59237 г;

1 унция = 1/16 фунта = 28,349523125 г

Листинг:

Program.cs

```
namespace LW_1
{
    public class Weight
    {
        private const float GramsInPound = 453.59237f;
        private const float GramsInOunce = 28.349523125f;

        private float _grams;

        private Weight(float grams) { _grams = grams; }
        public static Weight CreateFromGrams(float grams) {
            return new Weight(grams);
        }
        public static Weight CreateFromPounds(float pounds) {
            return new Weight(pounds * GramsInPound);
        }

        public float Grams
        {
            get { return _grams; }
            set { _grams = value; }
        }
        public float Pounds
        {
            get { return _grams / GramsInPound; }
            set { _grams = value * GramsInPound; }
        }
        public float Ounces
        {
```

```

        get { return _grams / GramsInOunce; }
        set { _grams = value * GramsInOunce; }
    }

    public static Weight operator+(Weight weight_1, Weight
weight_2) {
        return CreateFromGrams(weight_1.Grams +
weight_2.Grams);
    }
    public static Weight operator-(Weight weight_1, Weight
weight_2) {
        return CreateFromGrams(weight_1.Grams -
weight_2.Grams);
    }
}

public class LW_1
{
    public static void Main() {}
}
}

```

UnitTest.cs

```

using Microsoft.VisualStudio.TestTools.UnitTesting;
using LW_1;

namespace WeightTest
{
    [TestClass]
    public class WeightTest
    {
        [TestMethod]
        public void TestCreateFromGrams()
        {
            Assert.AreEqual(50.4f,
Weight.CreateFromGrams(50.4f).Grams);
        }
        [TestMethod]
        public void TestCreateFromPounds()
        {
            Assert.AreEqual(2.3f,
Weight.CreateFromPounds(2.3f).Pounds);
        }
        [TestMethod]
        public void TestGrams()
        {
            Weight weight = Weight.CreateFromGrams(0);
            weight.Grams = 1.5f;
            Assert.AreEqual(1.5f, weight.Grams);
        }
        [TestMethod]
        public void TestPounds()
        {

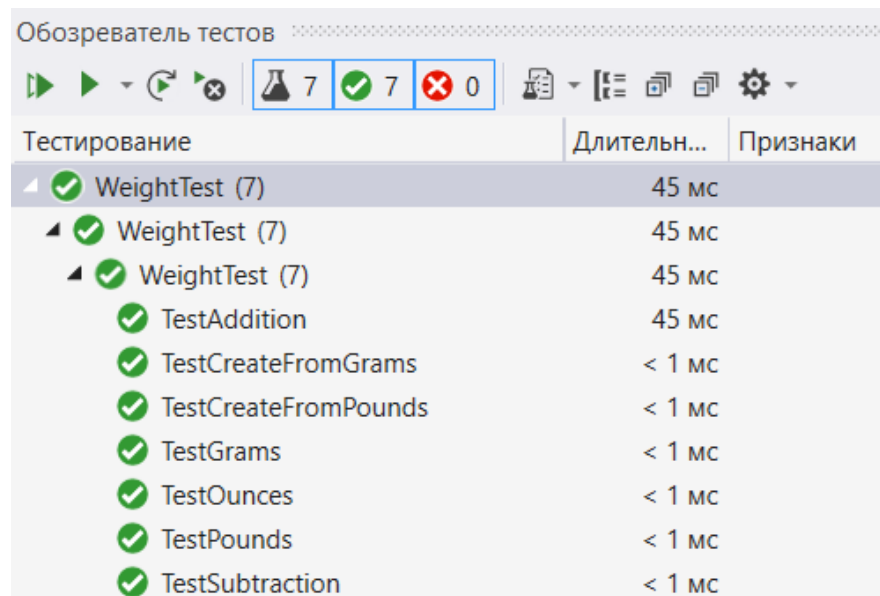
```

```

    {
        Weight weight = Weight.CreateFromGrams(0);
        weight.Pounds = 12.42f;
        Assert.AreEqual(12.42f, weight.Pounds);
    }
    [TestMethod]
    public void TestOunces()
    {
        Weight weight = Weight.CreateFromGrams(0);
        weight.Ounces = 65.3f;
        Assert.AreEqual(65.3f, weight.Ounces);
    }
    [TestMethod]
    public void TestAddition()
    {
        Weight weight_1 = Weight.CreateFromGrams(4.2f);
        Weight weight_2 = Weight.CreateFromGrams(5.5f);
        Assert.AreEqual(9.7f, (weight_1 + weight_2).Grams);
    }
    [TestMethod]
    public void TestSubtraction()
    {
        Weight weight_1 = Weight.CreateFromGrams(20.5f);
        Weight weight_2 = Weight.CreateFromGrams(12.2f);
        Assert.AreEqual(8.3f, (weight_1 - weight_2).Grams);
    }
}
}

```

Результат:



Тестирование	Длительн...	Признаки
WeightTest (7)	45 мс	
WeightTest (7)	45 мс	
WeightTest (7)	45 мс	
TestAddition	45 мс	
TestCreateFromGrams	< 1 мс	
TestCreateFromPounds	< 1 мс	
TestGrams	< 1 мс	
TestOunces	< 1 мс	
TestPounds	< 1 мс	
TestSubtraction	< 1 мс	

Рис. 1. Unit-тест

Вывод: в ходе работы были получены практические навыки создания классов, закрытых конструкторов, свойств, unit-тестов, перегрузки операторов на C#.