**Выполнил:** Штырев Глеб Сергеевич, группа ИСТб-23-2

# Лабораторное задание №1

## Задача на If

### Постановка задачи

**Задача №12.** Дано натуральное число 1≤n≤9999, определяющее стоимость товара в копейках. Выразить стоимость в рублях и копейках, например, 3 рубля 21 копейка, 15 рублей 5 копеек, 1 рубль ровно и т. п.

### Словесное описание алгоритма

**Main:**

1. Объявляем целочисленную переменную n и присваиваем ей значение, введенное пользователем;

2. Выводим на экран значение, полученное при подаче переменной n на вход метода Convert класса Logic;

**Convert:**

1. Объявляем строковую переменную outMessage;

2. Объявляем целочисленную переменную x и пробуем присвоить ей значение int.Parse(n)

2. Если значение локального аргумента n не является целым числом:

2.1 Присваиваем переменной outMessage значение «некорректный ввод»;

3. Иначе:

3.1 Если значение x выходит за пределы 1≤n≤9999:

3.1.1 Присваиваем переменной outMessage значение «число за пределом ОДЗ»;

3.2 Иначе:

3.2.1 Если значение x делится нацело на 100:

3.2.1.1 Присваиваем переменной outMessage значение (x / 100) + " рублей ровно";

3.2.2 Иначе:

3.2.2.1 Присваиваем переменной outMessage значение x / 100 + " рублей " + x % 100 + " копеек";

4. Возвращаем значение переменной outMessage;

### Таблица спецификаций

**Main**

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| n | Стоимость товара в копейках | Текст |

**Convert**

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| outMessage | Выводящееся сообщение с ответом или ошибкой | Текст |
| x | Стоимость товара в копейках | Целое положительное число |

### Таблица тестов

| Номер теста | Что проверяем | Входные данные | Выходные данные |
| --- | --- | --- | --- |
| 1 | Вводим текст | n = "AAAAAAAAA" | outMessage = "некорректный ввод" |
| 2 | Вводим число за пределами ОДЗ | n = "0" | outMessage = "число за пределом ОДЗ" |
| 3 | Вводим количество копеек, делящееся на 100 | n = "5500" | outMessage = "55 рублей ровно" |
| 4 | Вводим количество копеек, не делящееся на 100 | n = "5545" | outMessage = "55 рублей 45 копеек" |

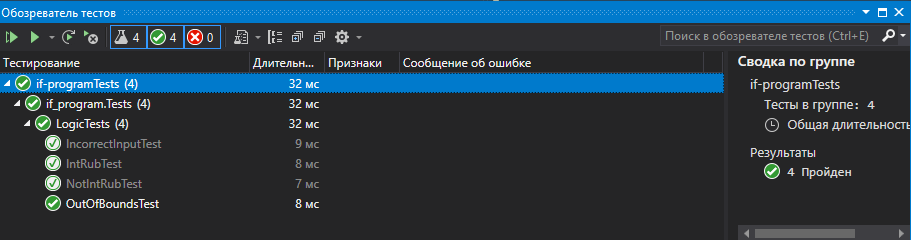
### Код программы

|  |
| --- |
| using System;  namespace if\_program  {  public class Logic  {  public static string Convert(string n)  {  string outMessage;  if (!int.TryParse(n, out int x))  {  outMessage = "некорректный ввод";  }  else  {  if (x < 1 || x > 9999)  {  outMessage = "число за пределом ОДЗ";  }  else  {  if (x % 100 == 0)  {  outMessage = (x / 100) + " рублей ровно";  }  else  {  outMessage = x / 100 + " рублей " + x % 100 + " копеек";  }  }  }  return outMessage;  }  }  class Program  {  static void Main(string[] args)  {  Console.WriteLine("Введите натуральное число 1<=n<=9999");  string n = Console.ReadLine();  Console.WriteLine(Logic.Convert(n));  Console.ReadKey();  }  }  } Console.WriteLine(averageSum);  Console.ReadLine(); |

### Код тестов

|  |
| --- |
| using Microsoft.VisualStudio.TestTools.UnitTesting;  namespace if\_program.Tests  {  [TestClass()]  public class LogicTests  {  [TestMethod()]  public void IncorrectInputTest()  {  string n = "AAAAAAAAA";  Assert.AreEqual(Logic.Convert(n), "некорректный ввод");  }  [TestMethod()]  public void OutOfBoundsTest()  {  string n = "0";  Assert.AreEqual(Logic.Convert(n), "число за пределом ОДЗ");  }  [TestMethod()]  public void IntRubTest()  {  string n = "5500";  Assert.AreEqual(Logic.Convert(n), "55 рублей ровно");  }  [TestMethod()]  public void NotIntRubTest()  {  string n = "5545";  Assert.AreEqual(Logic.Convert(n), "55 рублей 45 копеек");  }  }  } |

### Результаты тестирования



### Ссылка на ГИТ

<https://github.com/Glebwen/TP-Lab1>