

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федерального государственного бюджетного образовательного
учреждения
высшего образования
**«РОССИЙСКИЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ
Г.В.ПЛЕХАНОВА»**
Техникум Пермского института (филиала)

Сопроводительная записка по дисциплине
«Поддержка и тестирование программных модулей»

по теме:

«Создание и запуск модульных тестов для управляемого кода»

...

Руководитель

_____ /Берестов Д.Б./

« ____ » _____ 2026 г.

Исполнитель

_____ /Рожкован А.А./

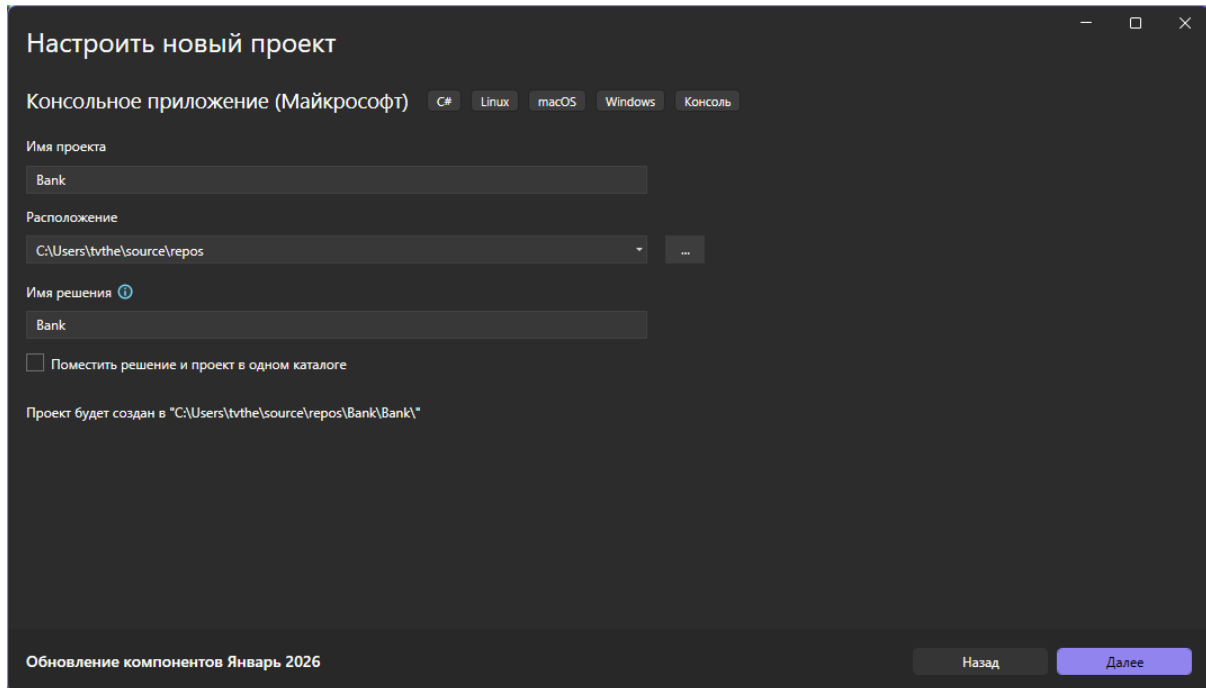
« ____ » _____ 2026 г.

Пермь, 2026 г.

Создание проекта для тестирования	2
Создание проекта модульного теста.....	4
Создание метода теста	6
Исправление кода и повторный запуск тестов	6
Создание нового метода теста	7

Создание проекта для тестирования

Для начала я создаю проект, выбираю шаблон Консольное приложение (см рис 1.)



(рис 1.)

Далее заменяю содержимое файла Program.cs кодом, предложенным методичкой (см рис 2.)

```

using System;
namespace BankAccountNS
{
    /// <summary>
    /// Bank account demo class.
    /// </summary>
    Ссылка: 4
    public class BankAccount
    {
        private readonly string m_customerName;
        private double m_balance;
        Ссылка: 0
        private BankAccount() { }
        Ссылка: 1
        public BankAccount(string customerName, double balance)
        {
            m_customerName = customerName;
            m_balance = balance;
        }
        Ссылка: 0
        public string CustomerName
        {
            get { return m_customerName; }
        }
        Ссылка: 1
        public double Balance
        {
            get { return m_balance; }
        }
        Ссылка: 1
        public void Debit(double amount)
        {
            if (amount > m_balance)
            {
                throw new ArgumentOutOfRangeException("amount");
            }
            if (amount < 0)
            {
                throw new ArgumentOutOfRangeException("amount");
            }
            m_balance += amount; // intentionally incorrect code
        }
    }
}

Ссылка: 0
public static void Main()
{
    BankAccount ba = new BankAccount("Mr. Bryan Walton",
    11.99);
    ba.Credit(5.77);
    ba.Debit(11.22);
    Console.WriteLine("Current balance is ${0}", ba.Balance);
}
}

```

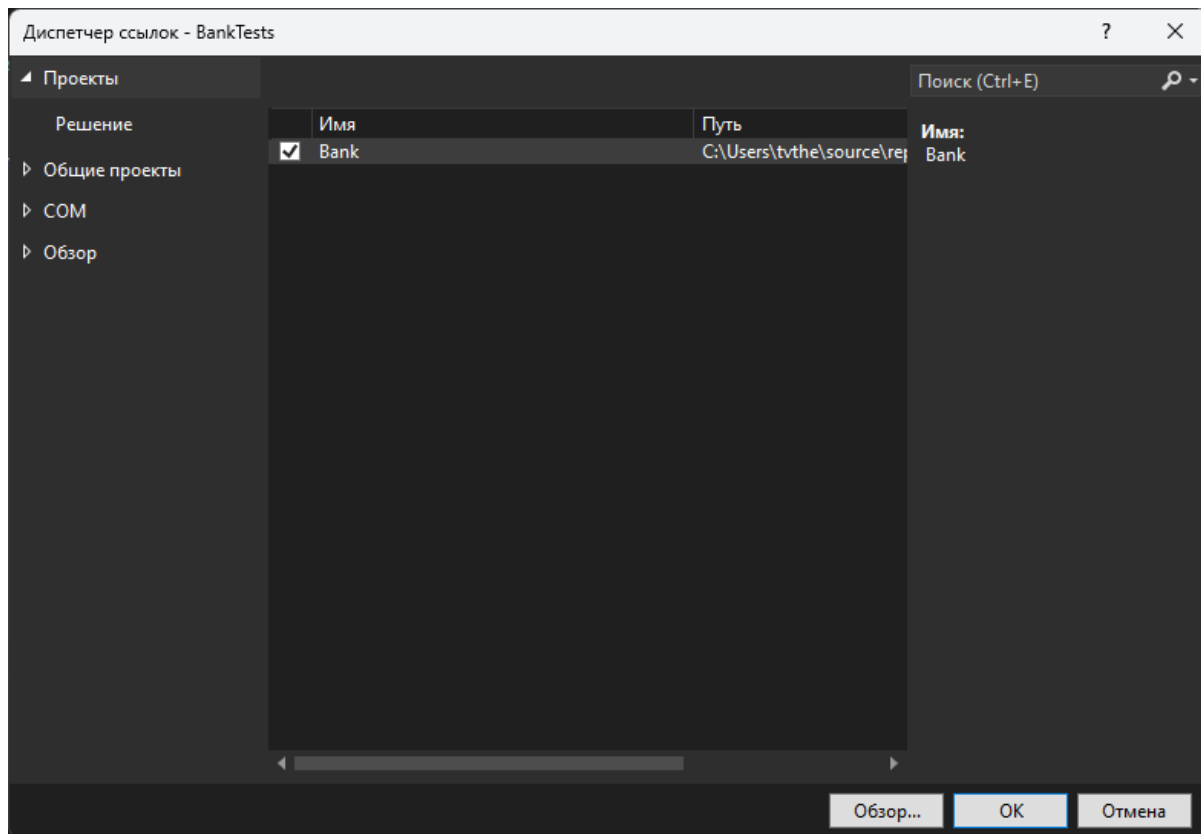
(рис 2.)

Переименовываю файл Programm.cs в BankAccount.cs в обозревателе решений.

В меню сборки нажимаю построить решение.

Создание проекта модульного теста

Создаю проект, выбираю шаблон MSTest. Называю проект BankTests и добавляю в решение Bank. В проекте BankTests добавляю ссылку на проект Bank (см рис 3.)



(рис 3.)

Далее переименовываю файл Test1.cs в BankAccountTests.cs.

Файл BankAccountTests.cs содержит вот такой код (см рис 4)

```
namespace BankTests
{
    [TestClass]
    Ссылка: 0
    public sealed class BankAccountTests
    {
        [TestMethod]
        Ссылка: 0
        public void TestMethod1()
        {
        }
    }
}
```

(рис 4.)

Добавляю оператор using BankAccountNS.

Создание метода теста

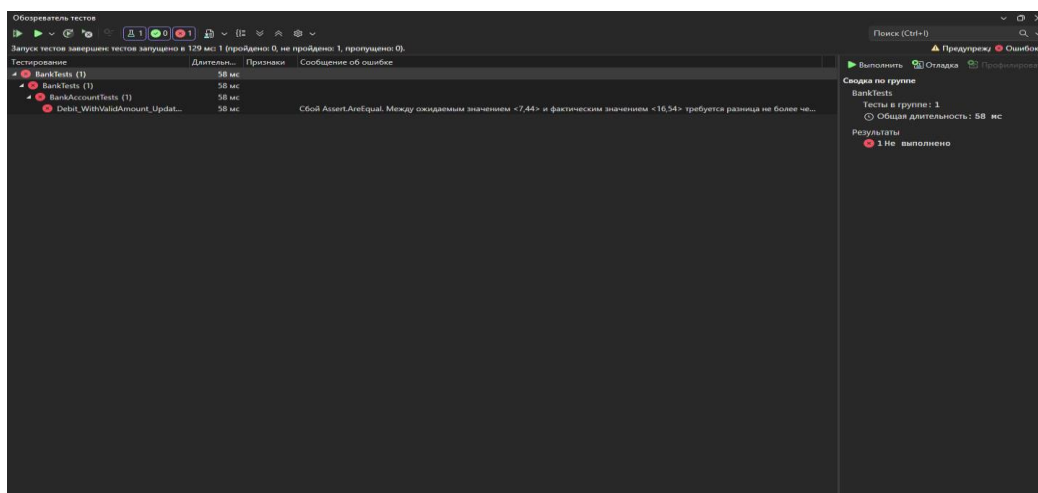
Добавляю метод (см рис 5.) в класс BankAccountTests.

```
using BankAccountNS;
namespace BankTests
{
    [TestClass]
    Ссылка: 0
    public sealed class BankAccountTests
    {
        [TestMethod]
        Ссылка: 0
        public void Debit_WithValidAmount_UpdatesBalance()
        {
            // Arrange
            double beginningBalance = 11.99;
            double debitAmount = 4.55;
            double expected = 7.44;
            BankAccount account = new BankAccount("Mr. Bryan Walton", beginningBalance);
            // Act
            account.Debit(debitAmount);
            // Assert
            double actual = account.Balance;
            Assert.AreEqual(expected, actual, 0.001, "Account not debited correctly");
        }
    }
}
```

(рис 5.)

Дальше в меню сборки нажимаю простроить решение, открываю обозреватель тестов нажимая клавиши CTRL + E, T.

Выбираю запустить все для того, чтобы выполнить тест. (см рис 6.)



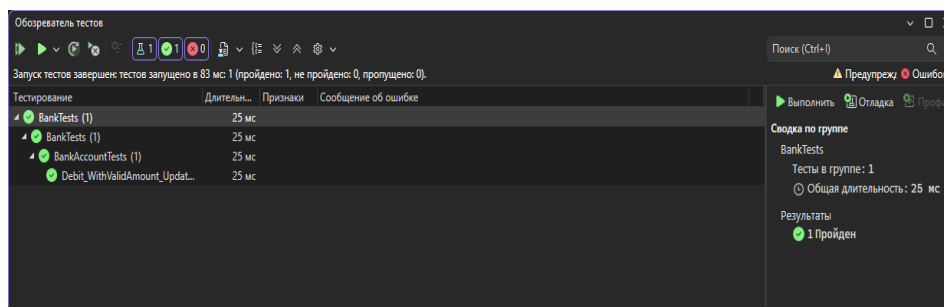
(рис 6.)

Исправление кода и повторный запуск тестов

Модульный тест обнаружил ошибку: сумма списания добавляется на баланс счета, вместо того чтобы вычитаться.

Для исправления этой ошибки в файле BankAccount.cs заменяю строку `m_balance += amount;` на `m_balance -= amount;`

Повторяю запуск теста (см рис 7). Теперь тест пройден.



(рис 7.)

Создание нового метода теста

Сначала я определил две константы добавил в класс BankAccount (см рис 8.)

```
public const string DebitAmountExceedsBalanceMessage = "Debit amount exceeds balance";  
public const string DebitAmountLessThanZeroMessage = "Debit amount is less than zero";
```

(рис 8.)

Затем заменил два условных оператора в методе Debit (см рис 9.)

```
public void Debit(double amount)  
{  
    if (amount > m_balance)  
    {  
        throw new System.ArgumentOutOfRangeException("amount", amount,  
            DebitAmountExceedsBalanceMessage);  
    }  
    if (amount < 0)  
    {  
        throw new System.ArgumentOutOfRangeException("amount", amount,  
            DebitAmountLessThanZeroMessage);  
    }  
}
```

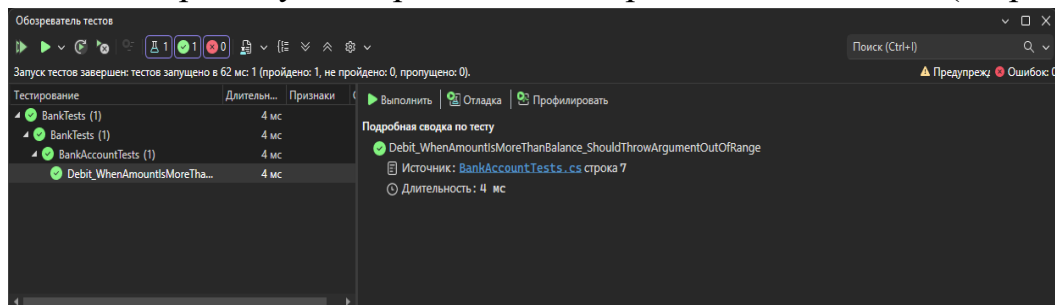
(Рис 9.)

Метод тестирования выглядит так (рис 10.)

```
using BankAccountNS;
namespace BankTests
{
    [TestClass]
    Ссылка: 0
    public sealed class BankAccountTests
    {
        [TestMethod]
        Ссылка: 0
        public void Debit_WhenAmountIsMoreThanBalance_ShouldThrowArgumentOutOfRangeException()
        {
            // Arrange
            double beginningBalance = 11.99;
            double debitAmount = 20.0;
            BankAccount account = new BankAccount("Mr. Bryan Walton", beginningBalance);
            // Act
            try
            {
                account.Debit(debitAmount);
            }
            catch (System.ArgumentOutOfRangeException e)
            {
                // Assert
                StringAssert.Contains(e.Message, BankAccount.DebitAmountExceedsBalanceMessage);
                return;
            }
            Assert.Fail("The expected exception was not thrown.");
        }
    }
}
```

(рис 10.)

Провожу тестирование, тестирование выполнено (см рис 11.)



(рис 11)