

# Лабораторная работа № 7

## Использование переменных ссылочного типа

### В лабораторной работе используется Microsoft Visual Studio 2008

#### Упражнение 1 – Добавление методов с параметрами в класс

В этом упражнении необходимо внести изменения в программу (файлы BankAccount.cs, Test.cs и AccountType.cs). В класс BankAccount надо добавить метод TransferFrom, обеспечивающий перевод денег с одного счета на другой. Необходимо написать программу проверки корректности перевода денег.

1. Откройте Visual Studio 2008.
2. В меню File выберите New, затем выберите Project.
3. Из списка шаблонов выберите Console Application. Впишите название проекта, например, Bank.
4. Изучите программу в файле BankAccount.cs. В файле дано описание класса BankAccount.
5. Отредактируйте класс. Добавьте метод TransferFrom. Метод не возвращает значений и должен иметь модификатор public. У метода должно быть два входных параметра: accForm и amount. Параметр accForm – это ссылка на объект BankAccount. Параметр amount имеет значение типа decimal.
6. В теле метода напишите два выражения, которые выполняют следующее:  
Дебетуйте значение amount из баланса accForm с использованием метода Withdraw.  
Проверьте, что снятие денег со счета было успешным. Если это так, то величина кредита amount в балансе текущего счета должна быть учтена. Для этого используйте метод Deposit.
7. Откомпилируйте программу. Откорректируйте ошибки и сохраните результат. Назначение created.accNo компилируется без ошибок, пока BankAccount был структурой.
8. Проверьте работу программы. Откройте файл Test.cs. Он содержит класс Test.
9. В метод Main добавьте код, который создает два объекта b1 и b2 на основе класса BankAccount. Инициализируйте значение баланса для каждого объекта = \$100. Для инициализации используйте метод Populate. Добавьте код, который отображает на консоль тип, номер и баланс для каждого счета (объекты b1 и b2) до перевода денег.
10. Добавьте код вызова метода TransferFrom и переведите \$10 с одного счета на другой.
11. Добавьте код, который отображает на консоль тип, номер и баланс для каждого счета (объекты b1 и b2) после перевода денег.
12. Откомпилируйте программу. Откорректируйте ошибки и сохраните результат. Убедитесь, что данные вводятся корректно. Покажите результат преподавателю. Полученный листинг упражнения 1 сохраните для отчета.

#### Упражнение 2 – Использование методов со ссылочными параметрами

В этом упражнении необходимо внести изменения в программу, находящуюся в файле Utils.cs. Нужно добавить статический метод Reverse в класс Utils. Метод читает строку символов и переносит ее в другую переменную в обратном порядке.

1. Откройте Visual Studio 2008.
2. В меню File выберите New, затем выберите Project.
3. Из списка шаблонов выберите Console Application. Впишите название проекта, например, Utils.

4. Добавьте метод `Reverse` как `public static` в класс `BankAccount`. Метод не возвращает значение и имеет единственный параметр-ссылку `s` типа `string`.
5. В методе `Reverse` создайте переменную типа `string sRev`. Она предназначена для сохранения результата обработки данных. Инициализируйте переменную `sRev` как `""`.
6. Напишите код, который преобразует строку. Для этого напишите цикл `for`, в котором надо выделять по одному символу из строки `s`, начиная с конца. Для этого используйте свойство `Length`, последовательно переходя от конца строки к началу. Вы можете использовать для обозначения строки нотацию массива, выделяя из нее по одному символу. Последний символ массива находится в позиции `Length-1`, а последний в позиции `0`. добавляйте символ в конец `sRev`.
7. Протестируйте метод `Reverse`. Для этого отредактируйте класс `Test`.
8. В методе `Main` класса `Test` объявите переменную `message` типа `string`. Прочитайте ее значение с консоли и присвойте его переменной `message`. Передайте значение строки в метод `Reverse` как ссылочный параметр. Отобразите результат на консоль.
9. Откомпилируйте программу. Откорректируйте ошибки и сохраните результат. Убедитесь, что данные вводятся корректно. Покажите результат преподавателю. Полученный листинг упражнения 2 сохраните для отчета.

### Упражнение 3 – Преобразование символов файла в верхний регистр

В этом упражнении необходимо написать программу, которая считывает текст из одного файла и переписывает в другой файл, переводя все буквы в верхний регистр.

1. Откройте Visual Studio 2008, если она не была открыта.
2. В меню `File` выберите `New`, затем выберите `Project`.
3. Из списка шаблонов выберите `Console Application`. Впишите название проекта, например, `CopyFileUpper`.
4. Отредактируйте файл `CopyFileUpper.cs`. Добавьте выражение `using` для пространства имен `System.IO`;
5. Добавьте в метод `Main` объявление двух переменных типа `string sFrom` и `sTo`. В этих переменных будут находиться имена входного и выходного файлов.
6. Объявите переменную типа `StreamReader` и назовите ее `srFrom`. Эта переменная предназначена для хранения ссылки на входной файл.
7. Объявите переменную типа `StreamWriter` и назовите ее `swTo`. Эта переменная предназначена для хранения ссылки на выходной поток.
8. Напишите подсказку пользователю, чтобы он (она) ввел(а) имя входного файла, напишите команду чтения введенного имени и сохранения его в переменной `sFrom` типа `string`.
9. Напишите подсказку пользователю, чтобы он (она) ввел(а) имя выходного файла, напишите команду чтения введенного имени и сохранения его в переменной `sTo` типа `string`.
10. Создайте блок `try-catch`. Он необходим для выявления различных исключений. Первый блок `catch FileNotFoundException` должен реагировать на несуществующий файл. Второй блок `Exception` должен реагировать на все остальные исключения. Напишите сообщения для пользователей для каждого типа исключений.
11. В блоке `try-catch` создайте новый объект `StreamReader`, используя имя входного файла в переменной `sFrom` и сохраните его в переменной-ссылке `srFrom`.
12. Аналогично создайте новый объект `StreamWriter`, используя имя входного файла в переменной `sTo` и сохраните его в переменной-ссылке `swTo`.
13. Добавьте цикл `while`, который работает до тех пор, пока метод `Peek()` из входного потока не возвратит значение `-1`. Внутри цикла

Используйте метод `ReadLine()` для обработки данных из входного потока, результат чтения каждой следующей строки записывается в переменную `sBuffer` типа `string`.

Примените метод `ToUpper` к переменной `sBuffer`.

Используйте метод `WriteLine`, чтобы отправить переменную `sBuffer` в выходной поток.

14. После завершения работы цикла метод закройте входной и выходной потоки.

15. Откомпилируйте программу. Откорректируйте ошибки и сохраните результат. Убедитесь, что данные вводятся корректно. Покажите результат преподавателю. Полученный листинг упражнения 3 сохраните для отчета.

16. Проверьте работу программы. Откройте командную строку и перейдите в каталог, где находится Ваш проект в папку `bin\debug`. Запустите программу `CopyFileUpper`. В качестве имени входного файла укажите `CopyFileUpper.cs`. Имя выходного файла `Test.cs`. Убедитесь, что весь текст программы в этом файле представлен буквами в верхнем регистре.

17. Полученный листинг упражнения 3 сохраните для отчета.

## Упражнение 4 – Проверка реализации интерфейса

В этом упражнении необходимо написать программу, которая будет добавлять статический метод `IsItFormattable` в класс `Utils`.

1. Откройте Visual Studio 2008, если она не была открыта.
2. В меню `File` выберите `New`, затем выберите `Project`.
3. Из списка шаблонов выберите `Console Application`. Впишите название проекта, например, `InterfaceTest`.
4. Отредактируйте класс `Utils`. Создайте метод `IsItFormattable` как `public static`. Метод должен возвращать тип `bool` и иметь один параметр `x` типа `object`. Используйте оператор `is` для того, чтобы определить передается ли объект, поддерживающий `System.IFormattable`. Если да, то вернуть `true`, если нет, то `false`.
5. Протестируйте метод `IsItFormattable`. Отредактируйте класс `Test`. В методе `Main` объявите и инициализируйте три переменные `i`, `ul` и `s` типов `int`, `ulong` и `string` соответственно. Значения для инициализации `0`, `0` и «Test». Передайте каждую переменную в `Utils.IsItFormattable()` и выведите на консоль результат каждого вызова.
6. Откомпилируйте программу. Откорректируйте ошибки и сохраните результат. Убедитесь, что для переменных типов `int` и `ulong` выводится значение `true`, а для `string` - `false`. Покажите результат преподавателю.
7. Полученный листинг упражнения 4 сохраните для отчета.

## Упражнение 5 – Работа с интерфейсами

В этом упражнении необходимо написать метод `Display`, который использует оператор `as` для определения передачи объекта как параметра в интерфейс `IPrintable`.

1. Откройте Visual Studio 2008, если она не была открыта.
2. В меню `File` выберите `New`, затем выберите `Project`.
3. Из списка шаблонов выберите `Console Application`. Впишите название проекта, например, `TestDisplay`. Используйте исходный код, который находится в файлах `Test.cs`, `Utils.cs`, `IPrintable.cs` и `Coordinate.cs`. В файле `IPrintable.cs` представлено описание интерфейса `IPrintable`, вызывающего метод `Print`. В классе `Coordinate` применяется интерфейс `IPrintable`. В объекте `Coordinate` есть пара чисел, которые определяют позицию двумерном пространстве.

4. Отредактируйте класс Utils. Добавьте метод Display как public static void. У этого метода должен быть один параметр item, типа object, передаваемый как значение. В методе Display объявите переменную интерфейса ip типа IPrintable. Конвертируйте параметр item в ссылку на основе интерфейса IPrintable, используя оператор as. Сохраните результат в переменной ip. Если значение переменной ip не null, то используйте интерфейс IPrintable для вызова метода Print. Если значение переменной ip null, то используйте Console.WriteLine для отображения результата с помощью метода ToString() вместо параметра.

5. Протестируйте метод Display. Отредактируйте класс Test. В методе Main создайте переменную num типа int. Присвойте ей значение 65. Создайте переменную msg типа string. Присвойте ей значение "A String". Инициализируйте переменную типа Coordinate как конструктор Coordinate с = new Coordinate(21.0,68.0). Передайте эти три переменные в Utils.Display.

6. Откомпилируйте программу. Откорректируйте ошибки и сохраните результат. Покажите результат преподавателю.

7. Полученный листинг упражнения 5 сохраните для отчета.

Natalie E. Novakova