Практикум 2. Работа со ссылочными типами

Упражнение 1. Определение типов как значимых или ссылочных

В этом упражнении вы создадите консольное приложение, определяющее тип объект (значимый или ссылочный).

1. В Visual Studio создайте проект консольного приложения. Назовите этот проект Mihalich-zver.
2. Создайте экземпляры следующих классов:

* SByte; и Byte; и Intl6;
* Int32;
* Int64;
* String
* Exception;

Как показано в следующем фрагменте кода:

// C#

SByte a = 0; Byte b = 0; Int16 с = 0; Int32 d = 0; Int64 e = 0; string s = "";

Exception ex = new Exception();

3. Добавьте созданные экземпляры в массив:  
// C#

object[] types = { a, b, c, d, e, s, ex };

4. В цикле foreach проверьте свойство object.GetType().IsValueType, чтобы определить,  
является ли тип значимым. Напишите код для вывода на экран имени и типа каждо  
го объекта:

// C#

foreach ( object о in types ) {

string type;

if (o.GetType().IsValueType) type = "Value type";

else

type = "Reference Type";

Console.WriteLine("{0}: {1}", o.GetType(), type );

}

5. Запустите консольное приложение и проверьте, правильно ли определяются типы.

Упражнение 2. Работа со строками и массивами

В этом упражнении вы создадите функцию сортировки строк.

1. В Visual Studio создайте проект консольного приложения. Назовите этот проект SortString.
2. Определите строковый тип и воспользуйтесь методом String.Split, чтобы разделить строку слова и поместить их в массив:

//c#

string s = "Microsoft .NET Framework 2.0 Application Development Foundation";

string[] sa = s.Split(“ “);

3. Вызовите метод Array.Sort для сортировки массива слов:  
// С#

Array.Sort(sa);

4. Вызовите метод String.Join, чтобы преобразовать массив слов в единую строку, а за  
тем выведите эту строку на консоль. Это делается следующим образом:

// C#

s = string.Join(" ", sa); Console.WriteLine(s);

5. Запустите консольное приложение, чтобы убедиться в корректности его работы.

Упражнение 3. Работа с потоками и исключениями

Разработчик написал приложение Windows Forms для просмотра текстовых файлов. Однако пользователи жалуются, что оно «весьма капризно». Опечатки при вводе имен файла, а также недоступные файлы приводят к остановке приложения из-за необрабо­танного исключения. Вы должны добавить к приложению код обработки исключений, который будет показывать пользователям понятные сообщения об ошибках, если файл окажется недоступным.

1. Скопируйте на жесткий диск папку Chapter01\Lesson2-ViewFile, и откройте версию проекта ViewFile для С#
2. Исключения возникают при попытке открыть файл. Следовательно, нужно изменить код, выполняющийся в ответ на событие showButton.Click. Добавьте код, перехваты­вающий все типы исключений и отображающий пользователю сообщение об ошиб­ке. Если исключение возникает после инициализации объекта TextReader, нужно зак­рыть этот объект, независимо от возникновения исключения. Для этого понадобит­ся два вложенных блока Try: один для перехвата исключений при инициализации объекта TextReader, а второй — для перехвата исключений, возникающих при чтении файла. Это показано на следующем примере кода:

// C#

try

{

TextReader tr = new StreamReader(locationTextBox.Text);

try

{ displayTextBox.Text = tr. ReadToEnd(); }

catch (Exception ex)

{ MessageBox.Show(ex.Message);}

finally

{ tr.Close(); }

}

catch (Exception ex)

{ MessageBox.Show(ex.Message); }

1. Запустите приложение. Сначала убедитесь, что оно успешно отображает содержимое файла. Затем введите недопустимое имя файла, чтобы проверить, появится ли сооб­щение об ошибке.
2. Добавьте код обработки исключений System.IO.FileNotFoundException и System.UnauthorizedAccessException, как показано в следующем фрагменте кода:

// C#

try

{

TextReader tr = new StreamReader(locationTextBox.Text); try

{ displayTextBox.Text = tr.ReadToEndO; } catch (Exception ex)

{ MessageBox.Show(ex.Message); }

finally

{ tr.Close(); }

}

catch (System.IO.FileNotFoundException ex)

{ MessageBox.Show("Sorry, the file does not exist."); }

catch (System.UnauthorizedAccessException ex)

{ MessageBox.Show("Sorry, you lack sufficient privileges."); }

catch (Exception ex)

{ MessageBox.Show(ex.Message); }

5. Снова запустите приложение и проверьте, появляется ли новое сообщение об ошиб­ке при вводе недопустимого имени файла.