Лабораторная работа №1

Цель работы

Изучение структуры программы на языке C# и приобретение навыков её компиляции и отладки.

Упражнение 1. Создание простой программы

Прежде чем приступить к изучению синтаксиса языка, рассмотрим структуру C# - программы:

- Объявление пространства имен.
- Класс.
- Методы класса.
- Атрибуты класса.
- Метод Main.
- Реализация логики программы.
- Комментарии.

Напишите программу, выводящую строку “Hello world”:

```
using System;
namespace HelloWorldApplication
{
   class HelloWorld
   {
      static void Main(string[] args)
      {
         /* my first program in C# */
         Console.WriteLine("Hello World");
         Console.ReadKey();
      }
   }
}
```

Для компиляции и запуска нажмите **F5.**

Рассмотрим код программы:

- Первая строка **using System;** ключевое слово using используется для включения пространства имен **System** в приложение. Директив **using** может быть сколько угодно много.

- Далее идет объявление пространства имен директивой **namespace**. Пространство имен содержит набор классов. Пространство имен **HelloWorldApplication** содержит только один класс **HelloWorld**.
- После идет объявление класса директивой **class**. Класс **HelloWorld** содержит данные и методы, определяющие логику работы. Класс может содержать множество методов.
- Класс **HelloWorld** определяет метод **Main**, который является входной точкой C# - программы (**entry point**).
- Следующая строка ограничена символами комментария **/*…*/**, она игнорируется компилятором.
- Работа метода определяется выражением **Console.WriteLine("Hello World")**. **WriteLine –** это метод класса **Console**, определенного в пространстве имен **System**. Выражение приводит к выводу строки Hello World на экран.
- Последняя строчка кода **Console.ReadLine()** означает, что программа ожидает нажатия клавиши на клавиатуре и не дает окну консоли закрыться сразу же после вывода текста.

Упражнение 2. Компиляция и запуск C#-программы из командной строки

В этом упражнении необходимо откомпилировать и запустить Hello World программу из командной строки, для этого:

- Запустите **Visual Studio .NET Command Prompt (Start->All Programs->Visual Studio .NET->Visual Studio .NET Tools->Visual Studio .NET Command Prompt)**.
- Командой **cd** перейдите в каталог с исходным кодом написанной ранее Hello World программы.
- Откомпилируйте программу с помощью следующей команды: **csc /out:HelloWorld.exe Program1.cs.**
- Запустите программу, набрав в командной строке ее название **HelloWorld.**

Упражнение 3. Добавление в C#-программу обработчика исключительных ситуаций.

В этом упражнении необходимо реализовать программу с обработчиком исключительных ситуаций, который будет отлавливать ошибки времени выполнения. Программа будет запрашивать два числа, делить первое на второе и выводить полученный результат.

- В методе **Main()** напишите код, запрашивающий у пользователя первое число.
- Считайте введенное пользователем число и присвойте полученное значение переменной **temp** типа string.
- Сконвертируйте тип данных **string** в **int** выражением **int i = Int32.Parse(temp)**.
- Аналогично сделайте для второго числа:

```
Console.WriteLine("Please enter the first integer");
string temp = Console.ReadLine();
int i = Int32.Parse(temp);
Console.WriteLine("Please enter the second integer");
temp = Console.ReadLine();
int j = Int32.Parse(temp);
```

- Создайте новую переменную *k* типа **int**, в которую будет помещен результат деления числа *i* на *j*.
- Добавьте код, выводящий значение *k* на экран.
- Запустите программу, проверьте работу при введении ненулевых значений (*i*,*j* не равны нулю).
- Проверьте работу программы при *j* = 0.
- В программе возникнет исключение (деление на ноль).
- Добавьте в программу обработчик исключительных ситуаций, для этого поместите код метода **Main()** внутрь блока **try {}**:

```
try
{
    Console.WriteLine (...);
    ...
    int k = i / j;

    Console.WriteLine(...);
}
```

- В методе **Main()** после **try** добавьте **catch,** внутри которого должно выводиться сообщение об ошибке:

```
catch(Exception e)
{
    Console.WriteLine("An exception was thrown: {0}", e);
}
```

- Итоговый текст метода **Main()** будет выглядеть примерно так:

```
public static void Main(string[ ] args)
{
   try {
    Console.WriteLine ("Please enter the first integer");
         string temp = Console.ReadLine( );
         int i = Int32.Parse(temp);

    Console.WriteLine ("Please enter the second integer");
         temp = Console.ReadLine( );
         int j = Int32.Parse(temp);

         int k = i / j;
         Console.WriteLine("The result of dividing {0} by
    {1} is {2}", i, j, k);
   }
   catch(Exception e) {
         Console.WriteLine("An exception was thrown: {0}",
    e);
   }
}
```

- Протестируйте код обработчика исключительных ситуаций.

Упражнение 4. Реализуйте программу для ввода сведений о студенте, в том числе ФИО, номер группы, возраст, название университета и вывода данных на консоль. Реализуйте проверку на пустые значения с генерированием исключений.