C#. Занятие 1.

Программа:

I.Язык C#.

1. C# и платформа Framework.Net (версии).
2. Решение и типы проектов.
3. Структура файла «исходника» (файл .cs) .
4. Типы данных.
5. Арифметика и математические функции.
6. Консольный ввод/вывод.
7. Чтение и запись в текстовый файл.
8. Условные выражения и операторы ветвления.
9. Циклы.
10. Массивы.
11. Исключения. Блок finally.
12. Классы. Члены классов. Модификаторы.
13. Поля и свойства.
14. Функции-члены: конструкторы, методы, операторы, свойства, финализаторы.
15. Индексаторы.
16. Наследование.
17. Интерфейсы.
18. Класс Array. Коллекции.
19. Коллекция List.
20. Поиск и сортировка.
21. Делегаты.
22. События.

II.Windows приложения.

1. Типы проекта: Empty, Windows Form, Windows Presentation Foundation
2. Окна, метки, поля, кнопки.
3. Компоненты выбора.
4. Меню и панели инструментов.
5. События мыши и клавиатуры.
6. Стандартные диалоги и дочерние окна.
7. Вкладки (страницы свойств).
8. Диаграммы.
9. Работа с локальными БД.

Литература:

1. Петцольд Ч., Программирование с использованием Microsoft Windows Forms. Мастер-класс, СПб.: Питер, 2006. – 432 стр.: ил.
2. Троелсен Э., Язык программирования C# 5.0 и платформа .Net 4.5, М.: Вильямс, 2013. – 1312 стр.: ил.
3. Нейгел К., Ивьен Б., Глинн Д., Уотсон К., Скиннер М., C# 2008 и платформа .Net 3.5 для профессионалов. М.: Вильямс, 2009. – 1392 стр.: ил.
4. Петцольд Ч. .Net Book Zero – http://www.charlespetzold.com . – 266 стр. (формат PDF).
5. Петцольд Ч. Программирование для Microsoft Windows на C#. В 2-х томах. М.: Русская редакция, 2002. – 576 стр.: ил., 624 стр.: ил.

C# и платформа Framework.Net (версии)
MSIL (Microsoft Intermediate Language), CIL (Common Intermediate Language), IL
Среда CLR (Common Language Runtime)
CLS

Где взять - Framework.Net или Visual Studio

Решение (solution) и типы проектов – Empty, Console, Windows Form, Windows Presentation Foundation.

Создание нового проекта

- Create Project (на панели на Start Page) или меню File->New->Project

Далее в окне New Project выбрать в левой панели тип (Project types) язык Visual C#->Windows затем в правой панели (Templates) нужный тип шаблона:
Empty Project, Console Application, Windows Forms Application или WPF Application

Внизу окна задать имя проекта (Name), выбрать место расположения (Location) и имя решения (Solution Name). Будет создана папка для решения и вложенная в нее папка для проекта. Имя решения по умолчанию совпадает с именем проекта – можно не заполнять. Можно также снять флажок Create directory for solution.

- Можно создать пустое решение: тип – Other Project Types->Visual Studio Solutions, шаблон – Blank Solution. Тогда проекты можно добавлять в это решение, выбрав в проводнике решения (Solution Explorer) контекстное меню для решения (верхняя строчка) и команду Add->New Project… Далее появится окно Add New Project, аналогичное окну New Project.
- Добавлять в имеющееся решение новые проекты можно также, как и в пустое решение.

Выбор стартового проекта.

Если в решении имеются несколько проектов, требуется указать стартовый проект. Это можно сделать

- или в контекстном меню решения – Set Start Up Projects…
- или в контекстном меню проекта – Set as Start Up Project.

Файлы проекта.

В проект можно (а в пустой – нужно) добавлять файлы, в частности содержащие код на языке C#. Это делается

- либо в контекстном меню проекта: Add->New Item,
- либо в главном меню Project->Add New Item

В любом случае появится окно Add New Item. В левой панели (Categories) можно выбрать пункт Code, затем в правой панели Templates выбрать шаблон Code File. Внизу окна можно задать имя файла (Name) с расширением .cs (си шарп). Файл появится в проводнике в группе файлов проекта.

В файле папки решения имеется файл решения (тип .sln – Microsoft Visual Studio Solution). Вызов этого файла в Windows вызовет Microsoft Visual Studio с загруженным решением. В папке проекта располагаются, в частности файлы с кодом (.cs) и две папки: bin и obj. В папке bin имеется папка debug. Именно эта папка считается текущей папкой – здесь будет храниться exe файл, и здесь ищутся файлы, если при операциях чтения/записи файлов не задано полное имя файла.

Ссылки (References)

В проводнике в проект входит группа ссылок (References). В пустом проекте эта группа также пустая. В ней отображаются ссылки на библиотеки (.dll), добавленные к проекту помимо стандартных (например, System.dll). Для большинства программ необходимы и другие ссылки, например, при работе с XML файлами требуется ссылка System.XML (System.XML.dll). При создании Windows -приложения с использованием Microsoft Windows Forms требуются System.Drawing и System.Windows.Forms и т.д.

Добавить требуемую ссылку можно, выбрав в контекстном меню группы References проекта пункт Add Reference…, затем в окне Add Reference нужную вкладку (например, для System.XML – вкладку .Net) и выбрать из списка ссылку System.XML.

Структура файла кода (файл .cs) .

```
//using System;
namespace first
{
   class first
   {
      static void Main()
```

```
      {
         System.Console.WriteLine("press any key");
         System.Console.ReadKey(true); //не надо если F5 - CTRL+F5: Debug -> Start Without
//Debugging
      }
   }
}
```

Директивы using. Пространства имен. Классы.

Точка входа - Main(…).

Модификатор static. Модификаторы public и private. Main - вобще то private, но private можно опустить, и можно public.

Тип Main любой - void, int… Параметры либо отсутсnвуют, либо массив strings - (strings[] args)

Выбор нужной точки входа.

Если в нескольких классах имеется Main, необходимо уточнить,
в каком классе точка входа:
- либо опция /main при вызове компилятора в командной строке:

csc 1.cs 2.cs /main 2.cs
- либо конт.меню проекта->Properties->вкладка Application и в списке Startup objects выбрать класс в котором находится нужная Main - здесь, например, second.second

Классы, в которых есть Main, называются ОБЪЕКТ ПРИЛОЖЕНИЯ

Типы данных. Типы данных CLS. Приведение типов. Арифметика.

```
      //int i = 143;
      //float f = 12.34f;   //need suffix f !!!
      //double d = 56.789;
      //string s = "vasia";
      //long k;
      //k = long.Parse(Console.ReadLine());
      // type above is equal to lower:
      System.Int32 i = 143;
      System.Single f = 12.34f;   //need suffix f !!!
      System.Double d = 56.789;
      System.String s = "vasia";
      //       System.String s = Console.ReadLine();
      //       System.String s = f.ToString();
      //       System.String s = 555.678d.ToString();
      //       System.String s = System.Convert.ToString(555.67);
      System.Int64 k;
      k = Int64.Parse(Console.ReadLine());  // the  same: long.Parse

// Integer types:
// sbyte (System.SByte), ushort (System.UInt16), uint (System.UInt32) and
// ulong (System.UInt64) - is absent in CLS  !!!
```

Консольный ввод/вывод.

```
//*************************************************************************
      Console.WriteLine("{1}\n{2} {0}", i, f, s);
//!!!!! эквивалентно:
      Console.WriteLine("f=" +f+ "\ns=" +s+ " i=" + i);
//*************************************************************************
// При выводе:
// Порядок {} вывода значений произволен, Значений должно быть НЕ МЕНЬШЕ чем {...}
```

```
// - {5:d6} - вывести 5-е значение как целое, 6 позиций, слева - нули
// - {1:f2} или {1:e2} - 2 ПОСЛЕ ЗАПЯТОЙ
// - {1:g2} - ВСЕГО выводимых цифр
// - f - вещественное с фиксированной запятой
// - e - вещественное с плавающей запятой
// - g - вещественное с плавающей или (фиксированной) запятой
// - d - десятичное целое, x - шестнадцатиричное целое
// - ЦЕЛЫЕ можно выводить по форматам d, x, f, e, g
// - ВЕЩЕСТВЕННЫЕ НЕЛЬЗЯ выводить по форматам d и x (только f, e, g)
//*************************************************************************
// При вводе:
// - Метод System.Console.ReadKey без параметров или с параметром false
//   отображает вводимый символ на мониторе
// - Метод System.Console.ReadKey(true) с параметром true
//   НЕ ОТОБРАЖАЕТ вводимый символ на мониторе
// - Метод System.Console.ReadLine ВОЗВРАЩАЕТ строку - класс string
//   для преобразования введенного значения в число
//   необходимо использовать методы из пространства System.Convert:
//     System.Convert.ToString(7.7), Convert.ToInt32("-222") and so on
//   или метод Parse из пр-в System.ТИП: int.Parse(...), Int32.Parse(...) and so on
// - !!! При вводе с клавиатуры или при использовании метода Parse:
//      Single.Parse("-8,8") используется ЗАПЯТАЯ (а НЕ точка !!!)
//*************************************************************************
       Console.WriteLine("int {0:d}; {1:x}; {2:f}; {3:e3}; {4:g}; {5:d5}", i, i, i, i, i, i);
       Console.WriteLine("long {0:d}; {1:x}; {2:f}; {3:e3}; {4:g}; {5:d5}", k, k, k, k, k, k);
       Console.WriteLine("float {0:f}; {1:f1}; {2:e}; {3:e3}; {4:g}; {5:g3}", f, f, f, f, f, f);//нельзя {d}
     Console.WriteLine("double {0:f}; {1:f1}; {2:e}; {3:e3}; {4:g}; {5:g3}", d, d, d, d, d, d);// нельзя {d}
 //*************************************************************************
      Console.WriteLine("abs(d) = {0}", Math.Abs(d));
       Console.WriteLine("b**3 = {0}", Math.Pow(b, 3));
       Console.WriteLine("Sqrt(b) = {0}", Math.Sqrt(b));
       if (a > b) d = a;        else d = b;
       Console.WriteLine("max(a,b) = {0}", d);

// operation % is defined both for integer and float (double):
       Console.WriteLine("8 % 3 = {0}", 8 % 3);
       Console.WriteLine("8.5 % 3.2 = {0}", 8.5 % 3.2); //here 2.1 =8.5 - (2)*3.2
//*************************************************************************
```

Чтение и запись в текстовый файл.

```
// SimpleWrite:
// new line here need: \r\n. Mistake symbol in file: \n or \r
       Console.WriteLine("Text files");
       Console.WriteLine("by default write file in bin\\debug\\");
       string s = "test for System.IO.StreamWriter\r\nmethod StreamWriter.WriteLine";
// If 2-nd parameter equal true - at the end of file. If false or by default- erase
       System.IO.StreamWriter sw = new StreamWriter("FileText.txt", true);
       sw.WriteLine(s);
       sw.Close();
//SimpleRead:
       Console.WriteLine("Read file from bin\\debug\\:\n");
       System.IO.StreamReader sr = new StreamReader("FileText.txt");
       string s;
       Console.WriteLine("all file in string:");
       s = sr.ReadToEnd();
       Console.WriteLine(s);
       sr.Close();
```