C# and Microsoft .NET

Conditional Statements and Loops

Trainer: Georgi Panayotov
E-mail: smg@smg-bg.net
Phone / Viber: +359877347912

PRAGMATIC
www.pragmatic.bg

Last time…

- Strings
  - Reference type, but **immutable**
  - Creation (=, +=, String.Concat(…), StringBuilder)
  - Comparison
  - Searching
  - Changing
  - Formatting (String.Format(…) and String Interpolation)
  - Globalization
- Feedback – TryParse(…), Facebook, …

Control Flow (if)

- Control of Flow statements

```
if ( condition )
{
  // the code goes here
}
else if ( condition )
{
   // the code goes here
}
else
{
    // the code goes here
}
```

Control Flow (switch)

- switch VS if

```
switch ( original_value )
{
   case value_1:
            // the code goes here
            break;
   case value_2:
            // the code goes here
            break;
   default:
            // the code goes here
}
```

Loops (while)

- Pre-condition loops

```
while (condition)
{
   // the code goes here
}
```

Loops (do/while)

- Post-condition loops

```
do
{
   // the code goes here
} while (condition);
```

Loops (for)

- Pre-condition loops

```
for (initializer; condition; increment)
{
    // the code goes here
}
```

Nested loops

- The **for** loop

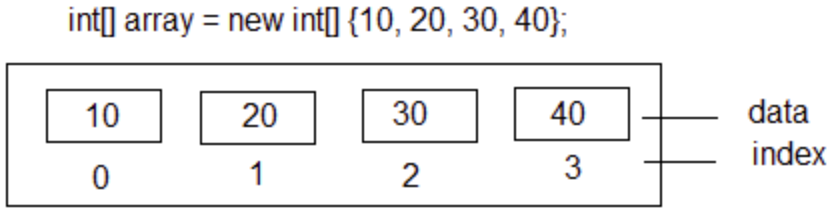
```
for (initializer; condition; increment)
{
    for (initializer; condition; increment)
    {
        // the code goes here
    }
}
```

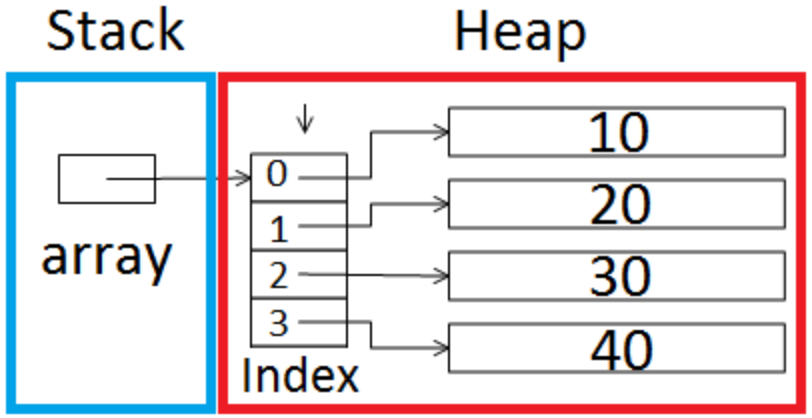
Arrays

- What is an array

  typeName[] arrayName = new typeName[size]

  typeName[] arrayName = { value1, value2, … }
- Arrays and Reference Types
- Collections… next time ☺





Loops (foreach)

- The **foreach** loop

```
foreach( type variable_name in collection)
{
   // the code goes here
}
```

Methods

- Why procedural programming?
- Declaring method

```
public static returnType methodName(type1 arg1, type2 arg2, ...)
{
  // the code goes here
  return result;
}
```

- The **void** return type
- The **out** and **ref** parameters

Recursion

- What is Recursion?

```
public static int Factoriel(int n)
{
   if (n == 0)
   {
     return 1;
   }
   return n*Factoriel(n-1);
}
```


Questions?