



The University of North Carolina at Chapel Hill

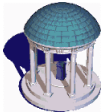
COMP 144 Programming Language Concepts
Spring 2002

Lecture 39: Case Study: C# and .NET

Felix Hernandez-Campos
April 29

COMP 144 Programming Language Concepts
Felix Hernandez-Campos

1

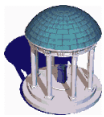


C# and .NET

- In 2000, Microsoft releases a new language, C#, heavily influenced by Java and C++
 - *Is there anything new from the programming languages point of view?*
- Microsoft is making it the key stone in their new development strategy (.NET)
 - *Big bucks... big evil...*
- Let's have a brief look at it, so you can put it on your resumes or simply laugh at Microsoft, depending on your point of view about the world
 - *I'm neutral (yeah, right...)*

COMP 144 Programming Language Concepts
Felix Hernandez-Campos

2



Hello World

- Java

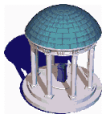
```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello world!");  
    }  
}
```

- C#

```
class HelloWorld {  
    static void Main(string[] args) {  
        System.Console.WriteLine("Hello world!");  
    }  
}
```

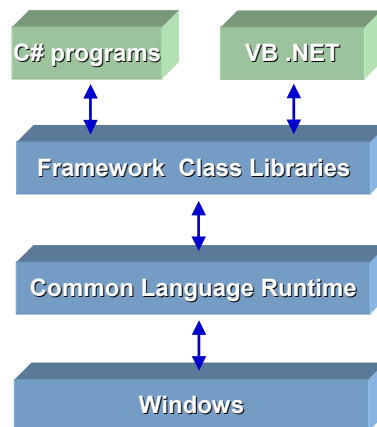
COMP 144 Programming Language Concepts
Felix Hernandez-Campos

3



Motivation for C#

- .NET
 - New development framework that promises to simplify Windows programming
 - » COM/DCOM is hard to learn
 - Heavy on component orientation
 - Language independence run-time system
 - » Common Language Runtime (CLR)



COMP 144 Programming Language Concepts
Felix Hernandez-Campos

4

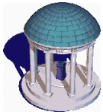


Common Language Runtime

- It can execute .NET program in an intermediate representation, the *Common Language Interface* (CLI)
- CLR is designed to work well in a multi-language environment
 - Java Virtual Machines is rather Java-oriented
 - CLR is supposed to work well with imperative programming languages (*e.g.*, C, Pascal) and statically typed object oriented languages (*e.g.*, C#, Eiffel)
 - Many language have compilers for CLR at different stages of development, including Python, Perl, Scheme, Haskell, Prolog,...

COMP 144 Programming Language Concepts
Felix Hernandez-Campos

5



Motivation for C#

- Rapid application development (RAD)
 - Visual development tools/languages such as Visual Basic and Delphi, are very popular and useful
 - » Remember Java Beans lecture
 - C# is optimized for such development model
- Platform-independence
 - CLR and CLI
- Access to platform-native resources
 - A more direct approach than the one taken by Java Native Interface (JNI)

COMP 144 Programming Language Concepts
Felix Hernandez-Campos

6



C# Syntax

Comparison with Java

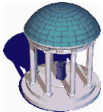
- If/then/else

Java

```
int i = 0;
if (i == 0) {
    i = 1;
} else {
    i = 2;
}
```

C#

```
int i = 0;
if (i == 0) {
    i = 1;
} else {
    i = 2;
}
```



C# Syntax

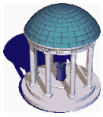
- Switch

Java

```
int i = 0;
switch (i) {
    case 0:
        i = 1;
        break;
    case 1:
        i = 2;
        break;
    default:
        i = -1;
        break;
}
```

C#

```
int i = 0;
switch (i) {
    case 0:
        i = 1;
        break;
    case 1:
        i = 2;
        break;
    default:
        i = -1;
        break;
}
```



C# Syntax

- While

Java

```
int i = 0;
while (i++ < 10) {
}
```

C#

```
int i = 0;
while (i++ < 10) {
}
```

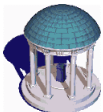
- Do/While

Java

```
int i = 0;
do {
} while (i++ < 10);
```

C#

```
int i = 0;
do {
} while (i++ < 10);
```



C# Syntax foreach

Java

```
import java.util.Vector;
public static int sum(Vector v) {
    int sum = 0;
    for (int j = 0; j < v.size(); j++) {
        Integer i = (Integer)v.elementAt(j);
        sum = sum + i.intValue();
    }
    return sum;
}
```

C#

```
using System.Collections;
static int SumList(ArrayList theList) {
    int sum = 0;
    foreach (int j in theList) {
        sum = sum + j;
    }
    return sum;
}
```



C# Syntax

- Break/Continue

Java

```
int i = 0;
while (i++ < 10) {
    if (i < 5) continue;
    break;
}
```

C#

```
int i = 0;
while (i++ < 10) {
    if (i < 5) continue;
    break;
}
```

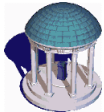
- Return

```
public void
returnNothing() {
    return;
}
public int returnOne() {
    return 1;
}
```

```
public void
returnNothing() {
    return;
}
public int returnOne() {
    return 1;
}
```

COMP 144 Programming Language Concepts
Felix Hernandez-Campos

11



C# Syntax

- Object instantiation

Java

```
Something s =
    new Something();
```

C#

```
Something s =
    new Something();
```

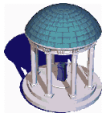
- Exclusive access

```
synchronized(this) {
    // do something
}
```

```
lock(this) {
    // do something
}
```

COMP 144 Programming Language Concepts
Felix Hernandez-Campos

12



C# Syntax

try/catch/finally

Java

```
try {  
    throw new SampleException();  
} catch (SampleException ex) {  
} finally {  
}
```

C#

```
try {  
    throw new SampleException();  
} catch (SampleException ex) {  
} finally {  
}
```

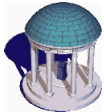
- **catch** clause is optional
- **catch** argument is optional

```
try {  
    throw new SampleException();  
} catch {} finally {  
}
```

- No **throws** keyword

COMP 144 Programming Language Concepts
Felix Hernandez-Campos

13



C# Syntax

- Class definition

Java

```
class Foo extends Bar {  
    ...  
}
```

C#

```
class Foo: Bar {  
    ...  
}
```

- Interface definition

```
interface IFoo extends IBar {  
    ...  
}
```

```
interface IFoo : IBar {  
    ...  
}
```

- Interface implementation

```
class Foo implements IFoo,  
IBaz {  
    ...  
}
```

```
class Bar: IFoo, IBaz {  
}
```

COMP 144 Programming Language Concepts
Felix Hernandez-Campos

14



Other C# Features

- C# provides Java-style garbage collection
- C# implements a Java- and Delphi-style value/reference-type system
 - Variables of primitive types also act like objects (unlike Java primitive objects)

Java

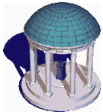
```
Integer iobj = new Integer(12);  
System.out.println(iobj.toString());
```

C#

```
Console.WriteLine(12.ToString());
```

COMP 144 Programming Language Concepts
Felix Hernandez-Campos

15



Other C# Features

- Enumerations

```
enum description: ulong {  
    Good,  
    Bad,  
    Ugly  
};
```

- Properties (forced getter and setters)

```
TextBlock tb;  
if (tb.backgroundColor == Color.green) {  
    // "get" is called for comparison  
    tb.backgroundColor = Color.red; // "set" is called  
} else {  
    tb.backgroundColor = Color.blue; // "set" is called  
}
```

COMP 144 Programming Language Concepts
Felix Hernandez-Campos

16



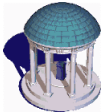
Other C# Features

- Get/set

```
public class TextBlock {  
    // Assume Color is an enum  
    private Color _bgColor;  
    private Color _fgColor;  
    public Color backgroundColor {  
        get {  
            return _bgColor;  
        }  
        set {  
            _bgColor = value;  
        }  
        //... and so on...  
    }  
}
```

COMP 144 Programming Language Concepts
Felix Hernandez-Campos

17



Other C# Features

- Delegates

- Safe method reference (solved with interfaces in Java)

```
delegate int Comparator(object a, object b);  
  
class Quicksort {  
    static void sort(Comparator c, object[]  
        objectsToSort) {  
        // ... quicksort logic leading to a comparison  
        if (c(objectsToSort[left],  
            objectsToSort[pivot]) > 0) {  
            // recursive call...  
        } else { // ...and so on...  
        }  
    }  
};
```

COMP 144 Programming Language Concepts
Felix Hernandez-Campos

18



References

- Perry, *C#, the natural progression*
 - <http://www.javaworld.com/javaworld/jw-08-2000/jw-0804-itw-csharp.html>
- Johnson, *C#: A language alternative or just J--?*
 - <http://www.javaworld.com/javaworld/jw-08-2000/jw-0804-itw-csharp.html>
- Meijer and Gough, *Technical Overview of the Common Language Runtime*
 - <http://research.microsoft.com/~emeijer/Papers/CLR.pdf>