

# **C# / .NET Overview**

KC Java User's Group

by

Troy Landers

# What I'm Going to Talk About

- .NET Overview
- C# Overview
- Windows Forms
- ADO.NET

# How to Write C# for Free

- Go to [http://msdn.microsoft.com/netframework/downloads/framework1\\_1/](http://msdn.microsoft.com/netframework/downloads/framework1_1/)
- Get .NET Framework 1.1 SDK
- Download SharpDevelop, [www.sharpdevelop.com](http://www.sharpdevelop.com)
- Optionally get MSDE 2000, <http://msdn.microsoft.com/netframework/downloads/sdkmsde/default.aspx>

# .NET Overview

- Language independent (> 20 supported)
- CLR – run-time services, virtual machine, security, memory management
- FCL – foundation class libraries
- Code (C#, VB, ...) compiled into MSIL
- JIT compiler converts MSIL to machine code

# .NET Overview

- assembly – physical unit of sharing and reuse (e.g. DLL, EXE)
- versioning – end of DLL Hell

# C# Overview

- Hello World

```
using System;

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

# C# Overview

- namespaces instead of packages
- foreach statement - IEnumerable
- More built-in types than Java (e.g. decimal, unsigned types)
- Garbage collection
- Operator overloading
- Enumerations

# C# Overview

- Value and Reference types
- ref keyword allows pass by reference
- out keyword
- Boxing / unboxing

# C# Overview - Properties

- Syntactic sugar for getters and setters

```
private string stringProp;  
  
public string StringProp {  
    get {  
        return stringProp;  
    }  
    set {  
        stringProp = value;  
    }  
}
```

# C# Overview - Classes

```
[attributes] [access-modifiers] class identifier  
    [:base-class] {class body}
```

- Access modifiers:
  - public, private (default), protected, internal, protected internal
- static members, methods, constructors
- destructors for unmanaged resources
- Finalize() guaranteed to be called

# C# Overview - Classes

- `IDisposable.Dispose()` – for releasing resources prior to garbage collection
- `using` statement – calls `Dispose` when exit statement block
- Can nest classes – similar to static inner classes
- `structs` – similar to classes but are value types

# C# Overview – Inheritance

```
public class SubClass : ParentClass
```

- virtual keyword for polymorphic methods
- override for virtual base class methods
- new keyword when NOT overriding
- System.Object is root class

# C# Overview - Interfaces

```
public interface MyInterface :  
    BaseInterface2, BaseInterface1 { body }
```

```
public class AClass :  
    BaseClass, MyInterface { body }
```

# C# Overview – Operator Overloading

- Operators are static methods

```
public static MyType operator+(lhs, rhs)
```

- Not supported by VB.NET
- Indexers

```
type this [type arg] {get; set;}
```

# C# Overview - Delegates

- Reference type to encapsulate a method with a specific signature and return type
- Basically lightweight interfaces
- Multicast delegates
- event keyword

# Windows Forms

- Tool for building Windows apps
- Basic control classes provided for all standard Windows UI controls
- Based on forms for user interaction
- Heavy use of events (delegates)
- Closely related to Web Forms (ASP.NET)

# ADO.NET

- “Disconnected” data architecture
- DataSet – subset of database
- DataTable – represents a table
- DataAdapter – bridge between data source and DataSet
- DBConnection
- DBCommand

# ADO.NET

## Basic process

1. Create DataAdapter.
2. Fill DataSet.
3. Do something interesting with data...
4. Update database.

# ADO.NET

- Supports database relationships
- Some data-bound Windows Forms controls provided (e.g. DataGrid)
- Supports databases other than SQL Server

# What's Coming

- .NET Framework 2.0
- VS.NET 2005 (Whidbey),  
<http://lab.msdn.microsoft.com/vs2005/>
- SQL Server 2005 (Yukon),  
<http://www.microsoft.com/sql/2005/default.asp>

# Java to .NET Challenges

- Class libraries
- Development & opensource tools (NANT, NUnit)
- Microsoft-centric platform
- GUI-centric environment

# Resources

- Programming C#, Jesse Liberty
- Programming .NET Components, Juval Lowy
- C# Language Spec,  
<http://www.csharpfriends.com/Spec/index.aspx?specID=toc.htm>
- Mono Project, [www.mono-project.com](http://www.mono-project.com)
- WebMatrix, [www.asp.net/webmatrix](http://www.asp.net/webmatrix)