



# Sun Tech Talk 2

## Java 1.5 and NetBeans 5.0

**Pierre de Filippis**

---

Sun Campus Evangelist  
[de-filip@cs.wisc.edu](mailto:de-filip@cs.wisc.edu)



# Sun Academic Developer Initiative

# Review

- **Goal**

Prepare student developers to open source programming.

- **Tools**

Open software, open communities, open projects and free training.

- **Benefits**

Active participation of Sun in the code. Allows for regular improvements.

Leave email for more info...

# Java in Numbers



**Java**<sup>™</sup>

Sun Microsystems



**2.8B** **Java Devices**

**700M** **Java Powered PCs**

**4.5M** **Java Developers**

# Generics

- **Old**

```
List myList = new LinkedList();  
myList.add(new Integer(0));  
Integer x = (Integer) myList.iterator().next();
```

- **New**

```
List<Integer> myList = new LinkedList<Integer>();  
myList.add(new Integer(0));  
Integer x = myList.iterator().next();
```

# Enhanced for loop

- **Old**

```
ArrayList<Integer> list = new ArrayList<Integer>();  
for (Iterator i = list.iterator(); i.hasNext();) {  
    Integer value = (Integer)i.next();  
}
```

- **New**

```
ArrayList<Integer> list = new ArrayList<Integer>();  
for (Integer i : list) {...}
```

# Automatic Boxing/Unboxing

- **Old**

```
ArrayList<Integer> list = new ArrayList<Integer>();  
list.add(0, new Integer(42));  
int total = (list.get(0)).intValue();
```

- **New**

```
ArrayList<Integer> list = new ArrayList<Integer>();  
list.add(0,42);  
int total = list.get(0);
```

# Type Safe Enum

- **New**

```
public enum Operation {  
    PLUS    { double eval(double x, double y) { return x + y; } },  
    MINUS   { double eval(double x, double y) { return x - y; } },  
    TIMES   { double eval(double x, double y) { return x * y; } },  
    DIVIDE  { double eval(double x, double y) { return x / y; } };  
  
    // Do arithmetic op represented by this constant  
    abstract double eval(double x, double y);  
}
```

- **Usage**

```
for (Operation op : Operation.values()) {  
    System.out.printf("%f %s %f = %f%n", x, op, y, op.eval(x, y));  
}
```



# Varargs

- **New**

```
public static String format(String pattern,  
                             Object... arguments);
```

- **No longer need to pack arguments into an array explicitly**
- **Used in `printf`**

# Static Import

- **Old**

```
double r = Math.cos(Math.PI * theta);
```

- **New**

```
import static java.lang.Math.PI;
```

**or**

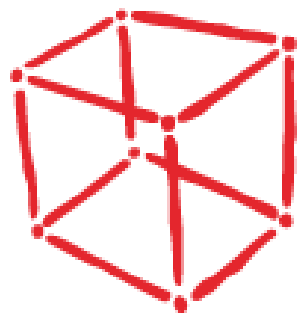
```
import static java.lang.Math.*;
```

```
double r = cos(PI * theta);
```

# Metadata

- **New**

```
/**
 * Describes the Request-For-Enhancement(RFE) that led
 * to the presence of the annotated API element.
 */
public @interface RequestForEnhancement {
    int    id();
    String synopsis();
    String engineer() default "[unassigned]";
    String date();      default "[unimplemented]";
}
@RequestForEnhancement(
    id        = 2868724,
    synopsis  = "Enable time-travel",
    engineer  = "Mr. Peabody",
    date      = "4/1/3007"
)
public static void travelThroughTime(Date destination) { ... }
```



# NetBeans

# Features

- **Features**

Open source IDE and platform  
Matisse GUI Builder (demo)  
Developer Collaboration  
NetBeans Profiler

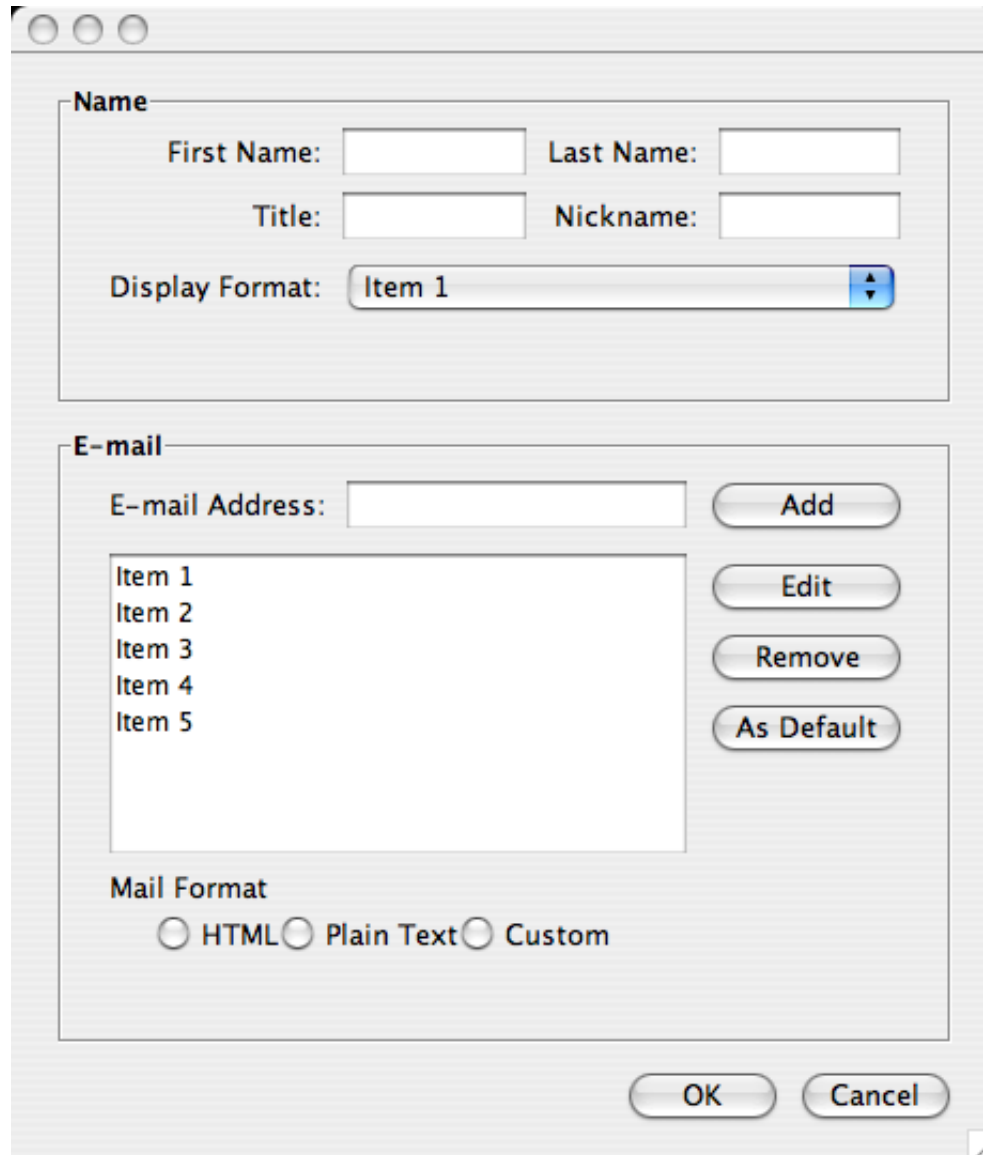
- **Get it Today**

<http://www.netbeans.org>



**NetBeans**.ORG

# Demo



**Name**

First Name:  Last Name:

Title:  Nickname:

Display Format:

**E-mail**

E-mail Address:

Item 1  
Item 2  
Item 3  
Item 4  
Item 5

**Mail Format**

HTML  Plain Text  Custom

TRANSFORM