Ada 95 on the JVM
Tea for 2 and 2 for Tea

Franco Gasperoni
gasperon@act-europe.fr
Overview

What’s Next?

Ada: when Software Matters

JGNAT: Ada environment for the JVM
Hardware evolution means ...

... BIG changes ahead ...
IDC Study …

✔ Information appliances outsell PCs by 2002

✔ By 2004 global market of 80 million units

✔ By 2004 18 Billion $ market
What will make or break a company reputation in this market?

... Software.
The Software Challenge

- ✔ Simplicity of use
- ✔ Portability
- ✔ Security
- ✔ Reliability
A Platform for Pervasive Computing ...
Java Benefits

✔ Portability (write once run anywhere)

✔ Built in security
... But

SECURE $\neq$ RELIABLE
So what’s wrong with Java?

```java
public static void send_bytes_to_port (byte first, byte last) {
    for (byte b = first; b <= last; b++) {
        // massage b & send it
        ...
    }
}
```
Increasing Reliability

- Java Language
- Java Virtual Machine
- Ada
- APIs and Libraries
Why use Ada on the JVM?

✔ Only language for the JVM that scales up
  – Separate specification & implementation

✔ Built-in reliability covers all software aspects
  – not just pointers or array bounds

✔ Direct support for
  – systems programming
  – real time systems
  – information systems
Reliability:
The Programming language matters
What was the programmer’s intent?

```ada
int var = 8265;
int far = 0252;
int bar = 9292;

var: Integer := 8265;
far: Integer := 2#1010_1010#;
bar: Integer := 16#58C#;
```
A program is written once
  - but read many times by different developers

Ada emphasizes readability
  - at the lexical & syntactic level

Is Murphy right?
Entertaining Examples

\[ x = y \rightarrow z; \]

```java
boolean some_condition;
boolean another_condition;
...
if (some_condition = another_condition){
    ...
}
```
The Lure of Cut & Paste

```c
static int some_func (int x) {...};
int x, y, z;
...
x = y--;some_func (z);
```
public float sumPositive (float [] values) {
    float sum = 0.0;
    if (values.length > 1)
        for (int i = 0; i < values.length; i++)
            if (values [i] > 0.0)
                sum += values [i];
    else
        sum = values [0];
    return sum;
}
Can’t Dangle

type Vector is array (Integer range <>) of Float;

... 

function Sum_Positive (values : Vector) return Float is
  sum : Float := 0.0;
begin
  if (values’length > 1) then
    for i in values’range loop
      if (values (i) > 0.0) then
        sum := sum + values (i);
      end if;
    end loop;
  else
    sum := values (values’first);
  end if;
  return sum;
end Sum_Positive;
Don’t Wrap Around

```ada
byte b = 127;
b++;  
```

```ada
type Byte is range -128 .. 127;
b : Byte := 127;
...
b := b + 1;
Exception raised
```
public static void send_bytes_to_port (byte first, byte last)
{
    for (byte b = first; b <= last; b++) {
        // massage b & send it
        ...
    }
}

procedure send_bytes_to_port (first : Byte; last : Byte) is
begin
    for b in first .. last loop
        -- massage b & send it
        ...
    end loop;

public class A {
    static { /* A initialization starts */ }
    public static int x = B.g();
    public static int f () { return A.x + 1; }
    static { /* A initialization ends */ }
}

public class B {
    static { /* B initialization starts */ }
    public static int y = A.f();
    public static int g () { return B.y + 1; }
    static { /* B initialization ends */ }
}
public class A {
    static { /* A initialization starts */ }
    public static int x = B.g();
    public static int f () { return A.x + 1; }
    static { /* A initialization ends */ }
}

public static void main (String args[]) { 
    if (args.length >0) {
        System.out.println (A.x);
        System.out.println (B.y);
    } else {
        System.out.println (B.y);
        System.out.println (A.x);
    }
}

public class B {
    static { /* B initialization starts */ }
    public static int y = A.f();
    public static int g () { return B.y + 1; }
    static { /* B initialization ends */ }
}
Elaboration in Ada

- Clearly defined set of rules
- Dynamic elaboration checks
- Elaboration control

- Previous example
  - circularity error/warning
  - exception raised
Overloading Confusions
public class A {
    public static void p (float f) { ... }
}

A.p (float)

Ada: Compile time error

void client () {
    A.p (1);
}

A.p (int)

Ada: OK - A.p (int)

public class A {
    public static void p (float f) { ... }
    public static void p (int i) { ... }
}
public class Base {}
public class Deriv extends Base {
public class Try {
    public static void p (Base obj) { ... }
}

void client () {
    Try.p (new Deriv ());
}

Ada: either
    Compile time error (1st case)
or
    Ambiguous call (2nd case)
Java does not enforce program consistency
public class Base {}
public class Deriv extends Base {}
public class Try {
    public static void p (Base obj) { ... }
}

void client () {
    Try.p (new Deriv ());
}

Compile Try.java: Try.p (Base)
Compile Client.java: Try.p (Deriv)

Ada: Binder consistency error
Some Java Missing Features
Scalar Abstraction

```ada
for (int w = start_weight; w <= end_weight; w++) {
    for (int len = start_length; len <= end_length; len++)
        do_something (w, len);
}

type Weight is range 0 .. 1_000;
type Length is range 0 .. 3_000;
procedure do_something (len : Length; w : Weight);

for w in start_weight .. end_weight loop
    for len in start_length .. end_length loop
        do_something (w, len);
    end loop;
end loop;
```

Compile time error
Unsigned Integers

\textbf{type} A\_Byte \textbf{is mod} 2^{\ast}\ast8;

\textbf{type} Hash\_Index \textbf{is mod} 1021;
**Fixed Point Types**

```ada
type Dollars is delta 0.01 digits 11;
  -- from -999_999_999.99 .. +999_999_999.99

type Volts is delta 0.001 digits 6 range 0.0 .. 240.0;
  -- from 0.0 .. 240.00 with 3 digits of precision
  -- e.g. 135.459 volts
```
type Integrand is access function (x : Float) return Float;

function Integrate (f : Integrand; a, b : Float);

function Fun (x : Float) return Float;

val : Float := Integrate (Fun’access, 1.0, 10.5);
Some More Missing Features

✔ Object layout in memory

✔ Real-Time programming
  - Doing things only at the API level has its problems

✔ Much more ........
JGNAT
Complete Ada Solution
100% Java compatible