CC BY SA

Ada and Java

Martyn Pike

university.adacore.com

Copyright © AdaCore

GNAT AJIS

- **GNAT Ada-Java Interfacing Suite (AJIS)**
- **Provides an Ada binding to the low-level JNI**
- **Automates the generation of JNI “glue” code**
- **ASIS based command line Tool**
  - ada2java

Copyright © AdaCore

GNAT AJIS

- **GNAT Ada-Java Interfacing Suite (AJIS)**
- **Provides an Ada binding to the low-level JNI**
- **Automates the generation of JNI "glue" code**
- **ASIS based command line Tool**
  - ada2java

```
package Printer is

   procedure Print(Me : in String);

end Printer;
```

```
with Ada.Text_IO;

package body Printer is

   procedure Print(Me : in String) is
   begin
      Ada.Text_IO.Put_Line(Me);
   end Print;

end Printer;
```

Copyright © AdaCore

Running ada2java

```
package Printer is

   procedure Print(Me : in String);

end Printer;
```

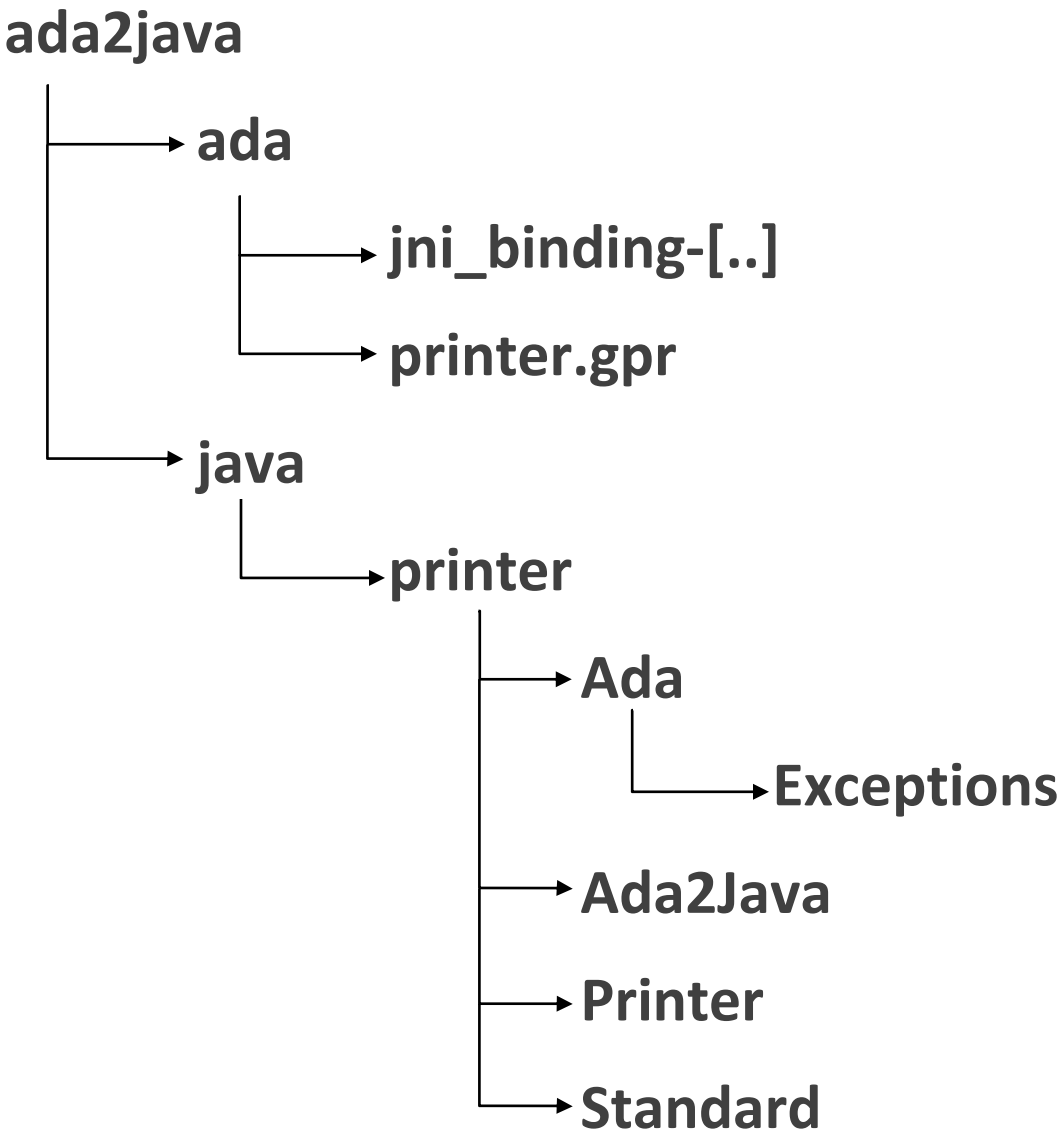
```
with Ada.Text_IO;

package body Printer is

   procedure Print(Me : in String) is
   begin
      Ada.Text_IO.Put_Line(Me);
   end Print;

end Printer;
```

```
ada2java printer.ads -b printer -o ada -c java -L printer
```

```
-b printer : Use this name as the base of generated Java packages

-o ada     : Generate the Ada glue code in a directory called ada

-c java    : Generate the Java interface in a directory called java

-L printer : This is the name of the library to be built
```

Copyright © AdaCore

Generated Code



Copyright © AdaCore

Hello World

```
import printer.Printer.*;
import printer.Standard.*;

public class My_Main {

   public static void main (String [] argv) {
      Printer_Package.Print(new AdaString("Hello World"));
   }
}
```

Copyright © AdaCore

Hello World

```
import printer.Printer.*;
import printer.Standard.*;

public class My_Main {

   public static void main (String [] argv) {
      Printer_Package.Print(new AdaString("Hello World"));
   }
}
```

```
ada2java printer.ads -b printer -o ada2java\ada -c ada2java\java -L printer
gprclean -Pada2java\ada\printer.gpr
gprbuild -p -Pada2java\ada\printer.gpr
set CLASSPATH=%CD%;%CD%\ada2java\java;%CD%\..\..\GNATGPL2014\lib\ajis.jar;%CLASSPATH%
set PATH=%CD%\ada2java\ada\lib;%PATH%
javac My_Main.java
java My_Main

Hello World
```

Copyright © AdaCore

pragma Annotate - AJIS

```
pragma Annotate (AJIS, *AJIS_annotation_identifier* {, *argument*});
```

- **AJIS.Annotations Identifiers**
  - Annotation_Renaming
  - Rename
  - Assume_Escaped
  - Attached
  - Bind
  - Monitor
  - Resolve_Ambiguous_Expression

Copyright © AdaCore

Mapping Ada Types to Java

- **Strings**
- **Scalar Types**
- **Arrays**
- **Simple Records**
- **Tagged Types**
- **Access Types**

Copyright © AdaCore

Mapping Ada Strings to Java

```
package String_Types is
   function Get_A_Fixed_String return String;
end String_Types;
```

```
/* AdaString.java */
[..]
final public boolean equals (java.lang.Object Right)
final public char Get_Element_At (int Index_1)
final public void Set_Element_At (int Index_1, char Value)
final public int First ()
final public int Last ()
final public int Length ()
final public java.lang.String toString ()
public AdaString (java.lang.String Str)
[..]
```

Copyright © AdaCore

Global Variables and Constants

```
package GlobVar is

   -- Example of a Global Variable
   My_Global : Integer := 40;

   -- Example of a Global Constant
   My_Constant : constant Integer := 50;

end GlobVar;
```

```
import globvar.GlobVar.*;

public class My_Main {

   public static void main (String [] argv) {
      System.out.println(GlobVar_Package.My_Global());
      GlobVar_Package.My_Global(60);
      System.out.println(GlobVar_Package.My_Global());
      System.out.println(GlobVar_Package.My_Constant());
   }
}
```

```
40
60
50
```

Copyright © AdaCore

Mapping Ada Arrays to Java

```
package Ada_Array is
  type An_Array is array(1..20) of Float;
end Ada_Array;
```

```
ada2java ada_array.ads -b adaarray -o ada2java\ada -c ada2java\java -L adaarray
```

```
/* An_Array.ada */
[..]
public An_Array ()
final public double Get_Element_At (int Index_1)
final public void Set_Element_At (int Index_1, double Value)
final public int First ()
final public int Last ()
final public int Length ()
[..]
```

Copyright © AdaCore

Mapping Ada Simple Records to Java

```
package Record_Types is
   type A_Nested_Record_Type is record
      Integer_Field : Integer;
   end record;

   type A_Record_Type is record
      Integer_Field : Integer;
      Float_Field : Float;
      Nested_Record : aliased A_Nested_Record_Type;
   end record;
end Record_Types;
```

```
/* A_Nested_Record_Type.java Operations */
public A_Nested_Record_Type ()    /* Constructor */

/* Getter */
final public int Integer_Field ()

/* Setter */
final public void Integer_Field (int Value)
```

```
/* A_Record_Type.java Operations */
public A_Record_Type ()    /* Constructor */

/* Getter */
final public adarecord.Record_Types.A_Nested_Record_Type Nested_Record ()

/* Setter */
final public void Nested_Record (adarecord.Record_Types.A_Nested_Record_Type Value)
```

Copyright © AdaCore

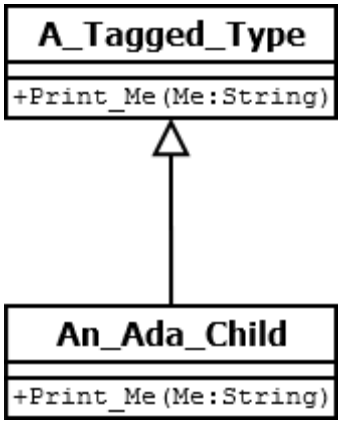
Mapping Ada Tagged Types to Java

```
package Tagged_Types is

   type A_Tagged_Type is tagged null record;

   procedure Print_Me (V : A_Tagged_Type; Me : String);

   type An_Ada_Child is new A_Tagged_Type with null record;

   procedure Print_Me (V : An_Ada_Child; Me : String);

   procedure Call_Print_Me (Str : String; Val : A_Tagged_Type'Class);

end Tagged_Types;
```

Copyright © AdaCore

Mapping Ada Tagged Types to Java

```
package Tagged_Types is

   type A_Tagged_Type is tagged null record;

   procedure Print_Me (V : A_Tagged_Type; Me : String);

   type An_Ada_Child is new A_Tagged_Type with null record;

   procedure Print_Me (V : An_Ada_Child; Me : String);

   procedure Call_Print_Me (Str : String; Val : A_Tagged_Type'Class);

end Tagged_Types;
```



Copyright © AdaCore

Mapping Ada Tagged Types to Java

```
package body Tagged_Types is

   procedure Print_Me (V : A_Tagged_Type; Me : String) is
   begin
      Put_Line ("FROM A TAGGED TYPE: " & Me);
   end Print_Me;

   procedure Print_Me (V : An_Ada_Child; Me : String) is
   begin
      Put_Line("FROM AN ADA CHILD: " & Me);
   end Print_Me;

   procedure Call_Print_Me (Str : String; Val : A_Tagged_Type'Class) is
   begin
      Print_Me(Val, Str);
   end Call_Print_Me;

end Tagged_Types;
```

Copyright © AdaCore

Mapping Ada Tagged Types to Java

```
public class My_Main {

   public static void main (String [] argv) {
      A_Tagged_Type v1 = new A_Tagged_Type ();
      An_Ada_Child v2 = new An_Ada_Child ();
      Tagged_Types_Package.Call_Print_Me (new AdaString ("V1"), v1);
      Tagged_Types_Package.Call_Print_Me (new AdaString ("V2"), v2);
   }

}
```

Copyright © AdaCore

Mapping Ada Tagged Types to Java

```
public class My_Main {

   public static void main (String [] argv) {
      A_Tagged_Type v1 = new A_Tagged_Type ();
      An_Ada_Child v2 = new An_Ada_Child ();
      Tagged_Types_Package.Call_Print_Me (new AdaString ("V1"), v1);
      Tagged_Types_Package.Call_Print_Me (new AdaString ("V2"), v2);
   }

}
```

FROM A TAGGED TYPE: V1
FROM AN ADA CHILD: V2

Copyright © AdaCore

Mapping Ada Tagged Types to Java

```
public class My_Main {

   static class A_Java_Child extends A_Tagged_Type {
      public void Print_Me (AdaString Me) {
        System.out.println ("FROM A JAVA CHILD: " + Me);
      }
   }

   public static void main (String [] argv) {
      A_Tagged_Type v1 = new A_Tagged_Type ();
      An_Ada_Child v2 = new An_Ada_Child ();
      A_Tagged_Type v3 = new A_Java_Child ();
      Tagged_Types_Package.Call_Print_Me (new AdaString ("V1"), v1);
      Tagged_Types_Package.Call_Print_Me (new AdaString ("V2"), v2);
      Tagged_Types_Package.Call_Print_Me (new AdaString ("V3"), v3);
   }

}
```

```
FROM A TAGGED TYPE: V1
FROM AN ADA CHILD: V2
FROM A JAVA CHILD: V3
```

Copyright © AdaCore

Mapping Ada Access Types to Java

```
package Access_Types is

   type Target_Length is record
      Value : Positive;
   end record;

   type Access_Length is access all Target_Length;

   procedure Use_Access_Type(Obj : in Access_Length);

end Access_Types;
```

Copyright © AdaCore

Mapping Ada Access Types to Java

```
package Access_Types is

   type Target_Length is record
      Value : Positive;
   end record;

   type Access_Length is access all Target_Length;

   procedure Use_Access_Type(Obj : in Access_Length);

end Access_Types;
```

```
procedure Use_Access_Type(Obj : in Access_Length) is
begin
   Put_Line(Obj.Value'Img);
end Use_Access_Type;
```

Copyright © AdaCore

Mapping Ada Access Types to Java

```
import adaaccess.Access_Types.*;

public class My_Main {

   public static void main (String [] argv) {
      Target_Length tgtLngth = new Target_Length();

      tgtLngth.Value(20);

      Access_Types_Package.Use_Access_Type(tgtLngth);
   }

}
```

Copyright © AdaCore

Mapping Ada Access Types to Java

```
import adaaccess.Access_Types.*;

public class My_Main {

   public static void main (String [] argv) {
      Target_Length tgtLngth = new Target_Length();

      tgtLngth.Value(20);

      Access_Types_Package.Use_Access_Type(tgtLngth);
   }

}
```

```
Exception in thread "main" com.adacore.ajis.NativeException: Value of Obj
cannot be escaped, because it's owned by the proxy. See pragma annotation
"Assume_Escaped" for more details.
  at com.adacore.ajis.internal.ada.Utils.checkEscapable(Utils.java:48)
  at adaaccess.Access_Types.Access_Types_Package.Use_Access_Type(Access_Types_Package.java:25)
  at My_Main.main(My_Main.java:10)
```

Copyright © AdaCore

Mapping Ada Access Types to Java

```
package Access_Types is

   type Target_Length is record
      Value : Positive;
   end record;

   type Access_Length is access all Target_Length;

   procedure Use_Access_Type(Obj : in Access_Length);
   pragma Annotate (AJIS, Assume_Escaped, False, Use_Access_Type, "Obj");

end Access_Types;
```

Copyright © AdaCore

Mapping Ada Access Types to Java

```
package Access_Types is

   type Target_Length is record
      Value : Positive;
   end record;

   type Access_Length is access all Target_Length;

   procedure Use_Access_Type(Obj : in Access_Length);
   pragma Annotate (AJIS, Assume_Escaped, False, Use_Access_Type, "Obj");

end Access_Types;
```

```
--[no-]assume-escaped
```

Copyright © AdaCore

Mapping Ada Access Types to Java

```
import adaaccess.Access_Types.*;

public class My_Main {

   public static void main (String [] argv) {
      Target_Length tgtLngth = new Target_Length();

      tgtLngth.Value(20);

      Access_Types_Package.Use_Access_Type(tgtLngth);
   }

}
```


Copyright © AdaCore

Ada calling Java Operations

```
package JPrinter is

   type Print_CB is access procedure (Str : String);

   procedure Call_Back(Meth : Print_CB; Str : String);
   pragma Annotate (AJIS, Assume_Escaped, False, Call_Back, "Meth");

end JPrinter;
```

Copyright © AdaCore

Ada calling Java Operations

```
package JPrinter is

   type Print_CB is access procedure (Str : String);

   procedure Call_Back(Meth : Print_CB; Str : String);
   pragma Annotate (AJIS, Assume_Escaped, False, Call_Back, "Meth");

end JPrinter;
```

```
with Ada.Text_IO; use Ada.Text_IO;

package body JPrinter is

   procedure Call_Back(Meth : Print_CB; Str : String) is
   begin
      Meth(Str);
   end Call_Back;

end JPrinter;
```

Copyright © AdaCore

Ada calling Java Operations

```
/* Print_CB.java */
public abstract class Print_CB implements com.adacore.ajis.IEscapable {
   abstract public void Print_CB_Body (jprinter.Standard.AdaString Str);
}
```

```
/* My_Printer.java */
public class My_Printer extends Print_CB {
   public void Print_CB_Body (jprinter.Standard.AdaString Str) {
      System.out.println(Str.toString());
   }
}
```

Copyright © AdaCore

Ada calling Java Operations

```
public class My_Main {
   public static void main (String [] argv) {
      My_Printer cb = new My_Printer();
      JPrinter_Package.Call_Back(cb, new AdaString("Hello World"));
   }
}
```

Hello World

Copyright © AdaCore

Name Clashes

```
package Name_Clashes is
   type I1 is new Integer;
   type I2 is new Integer;

   function F return I1;
   function F return I2;
end Name_Clashes;
```

Copyright © AdaCore

Name Clashes

```
package Name_Clashes is
   type I1 is new Integer;
   type I2 is new Integer;

   function F return I1;
   function F return I2;
end Name_Clashes;
```

name_clashes.ads:6:13: warning: name clash, can’t bind subprogram “F”

Copyright © AdaCore

Name Clashes

```
package Name_Clashes is
   type I1 is new Integer;
   type I2 is new Integer;

   function F return I1;
   pragma Annotate (AJIS, Rename, F, "F_I1");

   function F return I2;
   pragma Annotate (AJIS, Rename, F, "F_I2");
end Name_Clashes;
```

Copyright © AdaCore

Name Clashes

```
import nameclash.Name_Clashes.*;

public class My_Main {

   public static void main (String [] argv) {
       System.out.println(Name_Clashes_Package.F_I1());
       System.out.println(Name_Clashes_Package.F_I2());
   }
}
```

Copyright © AdaCore

Name Clashes

```
import nameclash.Name_Clashes.*;

public class My_Main {

   public static void main (String [] argv) {
       System.out.println(Name_Clashes_Package.F_I1());
       System.out.println(Name_Clashes_Package.F_I2());
   }
}
```

```
package body Name_Clashes is

   function F return I1 is
   begin
      return I1'First;
   end F;

   function F return I2 is
   begin
      return I2'Last;
   end F;

end Name_Clashes;
```

```
-2147483648
2147483647
```

Copyright © AdaCore

Overloaded Operators

Ada Operator	Java Name
=	OP_EQUAL
>	OP_GT
<	OP_LT
>=	OP_GE
<=	OP_LE
or	OP_OR
and	OP_AND
xor	OP_XOR
+	OP_PLUS
-	OP_MINUS
/	OP_DIV
*	OP_MUL
**	OP_EXP

Copyright © AdaCore

Overloaded Operators

Ada Operator	Java Name
=	OP_EQUAL
>	OP_GT
<	OP_LT
>=	OP_GE
<=	OP_LE
or	OP_OR
and	OP_AND
xor	OP_XOR
+	OP_PLUS
-	OP_MINUS
/	OP_DIV
*	OP_MUL
**	OP_EXP

```
package Op_Overload is

   subtype Example_T is Integer range 1..10;

   function "+"(Left, Right : Example_T) return Example_T;

   function ">="(Left, Right : Example_T) return Boolean;

end Op_Overload;
```

```
public final class Op_Overload_Package {
  static public int OP_PLUS (int Left, int Right)
  static public boolean OP_GE (int Left, int Right)
}
```

```
import opoverload.Op_Overload.*;

public class My_Main {

   public static void main (String [] argv) {
      System.out.println(
        Op_Overload_Package.OP_PLUS(1,5)
      );
      System.out.println(Op_Overload_Package.OP_GE(6,5));
   }
}
```

```
6
true
```

Copyright © AdaCore

Exceptions

```
package Except is

   An_Exception : exception;

   procedure Throw_An_Exception;

end Except;
```

```
package body Except is

   procedure Throw_An_Exception is
   begin
      raise An_Exception;
   end Throw_An_Exception;

end Except;
```

Copyright © AdaCore

Exceptions

```
/* An_Exception.java */
public final class An_Exception
  extends com.adacore.ajis.NativeException
  implements com.adacore.ajis.internal.ada.AdaException
{
   static public except.Ada.Exceptions.Exception_Occurrence
      createOccurrence (except.Standard.AdaString Message)
}
```

Copyright © AdaCore

Exceptions

```
/* An_Exception.java */
public final class An_Exception
  extends com.adacore.ajis.NativeException
  implements com.adacore.ajis.internal.ada.AdaException
{
   static public except.Ada.Exceptions.Exception_Occurrence
      createOccurrence (except.Standard.AdaString Message)
}
```

```
public class My_Main {

   public static void main (String [] argv) {
      try {
         Except_Package.Throw_An_Exception();
      } catch (An_Exception e) {
         System.out.println ("Exception thrown from Ada");
      }
   }
}
```

Copyright © AdaCore

Quiz

Copyright © AdaCore

Is this correct? (1/10)

YES
(click on the check icon)

NO
(click on the error location(s))

```
import quiz.Standard.*;

public class Q1 {

   public static void main (String [] argv) {
      java.lang.String MyString = new AdaString("Hello World");
      System.out.println(MyString);
   }
}
```

Copyright © AdaCore

Is this correct? (1/10)

NO

AdaString objects are not compatible with java.lang.String objects

```
import quiz.Standard.*;

public class Q1 {

   public static void main (String [] argv) {
      java.lang.String MyString = new AdaString("Hello World");
      System.out.println(MyString);
   }
}
```

```
import quiz.Standard.*;

public class Q1 {

   public static void main (String [] argv) {
      java.lang.String MyString = (new AdaString("Hello World")).toString();
      System.out.println(MyString);
   }
}
```

Copyright © AdaCore

? (2/10)

```
package Q2 is
   procedure Calc_Sum(X : Natural; Y : Natural);
   -- Can raise Constraint_Error
end Q2;
```

```
procedure Calc_Sum(X : Natural; Y : Natural) is
begin
   raise Constraint_Error;
exception
   when Constraint_Error =>
      Ada.Text_IO.Put_Line("Exception in Ada");
      raise;
end Calc_Sum;
```

```
public class Q2 {

   public static void main (String [] argv) {
      try {
         Q2_Package.Calc_Sum(10,20);
      } catch (Constraint_Error e) {
         System.out.println
            ("Exception thrown from Ada");
      }
   }
}
```

“Exception in Ada”

“Exception in Ada” & “Exception thrown from Ada”

“Exception thrown from Ada”

Copyright © AdaCore

? (2/10)

```
package Q2 is
   procedure Calc_Sum(X : Natural; Y : Natural);
   -- Can raise Constraint_Error
end Q2;
```

```
procedure Calc_Sum(X : Natural; Y : Natural) is
begin
   raise Constraint_Error;
exception
   when Constraint_Error =>
      Ada.Text_IO.Put_Line("Exception in Ada");
      raise;
end Calc_Sum;
```

```
public class Q2 {

   public static void main (String [] argv) {
      try {
         Q2_Package.Calc_Sum(10,20);
      } catch (Constraint_Error e) {
         System.out.println
            ("Exception thrown from Ada");
      }
   }
}
```

"Exception in Ada"

"Exception in Ada" &
"Exception thrown from Ada"

"Exception thrown from Ada"

Copyright © AdaCore

Is this correct? (3/10)

YES
(click on the check icon)

NO
(click on the error location(s))

```
package Q3 is

   type Nested_Array is array(1..30) of Natural;

   type Q3_Type is array(1..20) of aliased Nested_Array;

   procedure Set_Strings(
      The_Strings : in out Q3_Type
   );

end Q3;
```

```
package body Q3 is
   procedure Set_Strings(
      The_Strings : in out Q3_Type
   ) is
   begin
      The_Strings :=
         Q3_Type'(
            1      => (others => 6),
            2      => (others => 5),
            3      => (others => 4),
            7      => (others => 3),
            20     => (others => 9),
            others => (others => 1)
         );
   end Set_Strings;
end Q3;
```

```
import quiz.Q3.*;

public class Q3 {
   public static void main (String [] argv) {
      Q3_Type The_Strings = new Q3_Type();
      Q3_Package.Set_Strings(The_Strings);

      Nested_Array element =
        The_Strings.Get_Element_At(
           The_Strings.Last()
        );

      for (int i=1;i < element.Length(); i++) {
        System.out.print(element.Get_Element_At(i));
      }
   }
}
```

Copyright © AdaCore

Is this correct? (3/10)

YES

```
import quiz.Q3.*;

public class Q3 {
   public static void main (String [] argv) {
      Q3_Type The_Strings = new Q3_Type();
      Q3_Package.Set_Strings(The_Strings);

      Nested_Array element =
        The_Strings.Get_Element_At(
           The_Strings.Last()
        );

      for (int i=1;i < element.Length(); i++) {
        System.out.print(element.Get_Element_At(i));
      }
   }
}
```

999999999999999999999999999999

Copyright © AdaCore

? (4/10)

```
package Q4 is

   type Target is record
      Value : Positive := 20;
   end record;

   type Access_Target is not null access all Target;

   procedure Print_Target(Obj : Access_Target);

end Q4;
```

```
with Ada.Text_IO;

package body Q4 is

   procedure Print_Target(Obj : Access_Target) is
   begin
      Ada.Text_IO.Put_Line(Obj.Value'Img);
   end Print_Target;

end Q4;
```

```
public class Q4 {
   public static void main (String [] argv) {
      Target tgt = new Target();
      Q4_Package.Print_Target(tgt);
   }
}
```

20

40

Nothing – an Exception is raised

Copyright © AdaCore

? (4/10)

```
package Q4 is

   type Target is record
      Value : Positive := 20;
   end record;

   type Access_Target is not null access all Target;

   procedure Print_Target(Obj : Access_Target);
   pragma Annotate (AJIS, Assume_Escaped, False, Print_Target, "Obj");

end Q4;
```

```
with Ada.Text_IO;

package body Q4 is

   procedure Print_Target(Obj : Access_Target) is
   begin
      Ada.Text_IO.Put_Line(Obj.Value'Img);
   end Print_Target;

end Q4;
```

```
public class Q4 {
   public static void main (String [] argv) {
      Target tgt = new Target();
      Q4_Package.Print_Target(tgt);
   }
}
```

20

40

Nothing – an Exception is raised

Copyright © AdaCore

(5/10)

```
package Q5 is
   type Q5_Int is new Integer range 0..200;
   function "+"(X : Q5_Int; Y : Q5_Int) return Q5_Int;
   function "="(X : Q5_Int; Y : Q5_Int) return Boolean;
   function "-"(X : Q5_Int; Y : Q5_Int) return Q5_Int;
end Q5;
```

```
import quiz.Standard.*;
import quiz.Q5.*;

public class Q5 {
   public static void main (String [] argv) {
      System.out.print(
         Q5_Package.OP_PLUS(10,Q5_Package.OP_MINUS(50,10)) == 50
      );
   }
}
```

false

Code does not compile

true

Copyright © AdaCore

(5/10)

```
import quiz.Standard.*;
import quiz.Q5.*;

public class Q5 {
   public static void main (String [] argv) {
      System.out.print(
         Q5_Package.OP_EQUAL(Q5_Package.OP_PLUS(10,Q5_Package.OP_MINUS(50,10)),50)
      );
   }
}
```

false

Code does not compile

true

Copyright © AdaCore

Is this correct? (6/10)

YES
(click on the check icon)

NO
(click on the error location(s))

```
package Q6 is

   My_Global : Integer := 40;

   My_Constant : constant Integer := My_Global;

end Q6;
```

```
import quiz.Standard.*;
import quiz.Q6.*;

public class Q6 {
   public static void main (String [] argv) {
      System.out.print(Q6_Package.My_Constant(20));
   }
}
```

Copyright © AdaCore

Is this correct? (6/10)

NO

```
package Q6 is

   My_Global : Integer := 40;

   My_Constant : constant Integer := My_Global;

end Q6;
```

```
import quiz.Standard.*;
import quiz.Q6.*;

public class Q6 {
   public static void main (String [] argv) {
      System.out.print(Q6_Package.My_Constant(20));
   }
}
```

Copyright © AdaCore

(7/10)

```
package Q7 is

   type Print_CB is access procedure (Str : String);

   procedure Call_Back(Meth : Print_CB; Str : String);
   pragma Annotate (AJIS,                , False, Call_Back, "Meth");

end Q7;
```

Copyright © AdaCore

(7/10)

```
package Q7 is

   type Print_CB is access procedure (Str : String);

   procedure Call_Back(Meth : Print_CB; Str : String);
   pragma Annotate (AJIS, Assume_Escaped, False, Call_Back, "Meth");

end Q7;
```

Copyright © AdaCore

(8/10)

```
package Q8 is

type I2 is new Integer;

   function F return Integer;

   function F return I2;
   pragma Annotate (AJIS,       , F, "F_I2");

end Q8;
```

Copyright © AdaCore

(8/10)

```
package Q8 is

type I2 is new Integer;

   function F return Integer;

   function F return I2;
   pragma Annotate (AJIS, Rename, F, "F_I2");

end Q8;
```

Copyright © AdaCore

? (9/10)

```
package Q9 is

   type Parent_Type is abstract tagged record
      A_Field : Integer;
   end record;
   procedure Print_Me (V : Parent_Type; Me : String) is abstract;

   type Child_Type is abstract new Parent_Type with record
      Another_Field : Boolean := False;
   end record;

end Q9;
```

```
static class A_Java_Child extends Child_Type
{
   A_Java_Child() {
      A_Field(30);
   }

   public void Print_Me (AdaString Me) {
      System.out.println(
         Me + ":" + A_Field() + ":" + Another_Field()
      );
   }
}

public static void main (String [] argv) {
   A_Java_Child v2 = new A_Java_Child();
   v2.Print_Me(new AdaString ("Hello World"));
}
```

“Hello World:30:true”

“Hello World:30:false”

“Hello World:30:”
and then program crashes

Copyright © AdaCore

? (9/10)

```
package Q9 is

   type Parent_Type is abstract tagged record
      A_Field : Integer;
   end record;
   procedure Print_Me (V : Parent_Type; Me : String) is abstract;

   type Child_Type is abstract new Parent_Type with record
      Another_Field : Boolean := False;
   end record;

end Q9;
```

```
static class A_Java_Child extends Child_Type
{
   A_Java_Child() {
      A_Field(30);
   }

   public void Print_Me (AdaString Me) {
      System.out.println(
         Me + ":" + A_Field() + ":" + Another_Field()
      );
   }
}

public static void main (String [] argv) {
   A_Java_Child v2 = new A_Java_Child();
   v2.Print_Me(new AdaString ("Hello World"));
}
```

"Hello World:30:true"

"Hello World:30:false"

"Hello World:30:"
and then program crashes

Copyright © AdaCore

Is this correct? (10/10)

YES
(click on the check icon)

NO
(click on the error location(s))

```
package Q10 is

   type Java_CB is access procedure (Str : String);

   procedure Call_Back(Meth : Java_CB; Str : String);
   pragma Annotate (AJIS, Assume_Escaped, False, Call_Back, "Meth");

end Q10;
```

```
import quiz.Standard.*;
import quiz.Q10.*;

public class Q10 {

   public void Print_CB_Body (quiz.Standard.AdaString Str) {
      System.out.println(Str.toString());
   }

   public static void main (String [] argv) {
      Q10_Package.Call_Back(Print_CB_Body, new AdaString("Hello World"));
   }
}
```

Copyright © AdaCore

Is this correct? (10/10)

NO

```
package Q10 is

   type Java_CB is access procedure (Str : String);

   procedure Call_Back(Meth : Java_CB; Str : String);
   pragma Annotate (AJIS, Assume_Escaped, False, Call_Back, "Meth");

end Q10;
```

```
import quiz.Standard.*;
import quiz.Q10.*;

public class Q10 {

   public void Print_CB_Body (quiz.Standard.AdaString Str) {
      System.out.println(Str.toString());
   }

   public static void main (String [] argv) {
      Q10_Package.Call_Back(Print_CB_Body, new AdaString("Hello World"));
   }
}
```

Copyright © AdaCore

CC BY SA

university.adacore.com

Copyright © AdaCore