

# Security Engineering

## Problem Session 1

WS2018/19

Prof. Stefan Lucks, Nathalie Jolanthe Dittrich

`<first>.<middle>.<lastname> (at) uni-weimar.de`

Bauhaus-Universität Weimar

October 26, 2018

# First Assignment

- About getting familiar with Ada
- Installing the GNAT toolkit

# GNAT Project Files

```
project PS1 is

  for Main use ("animals.adb",
               "box_geometry.adb",
               "calculator.adb",
               "hello_world.adb",
               "loops.adb",
               "simple_calculator.adb",
               "vowels.adb");
  for Source_Dirs use "src";
  for Object_Dir use "bin";

  package Compiler is
    for Default_Switches ("ada") use ("-g", "-gnato",
                                     "-gnatwa", "-gnatwe", "-gnatQ", "-gnat12");
  end Compiler;

  package Builder is
    for Default_Switches ("ada") use ("-g");
  end Builder;

end PS1;
```

# Control Structures

What do the loops in the following program fragments do?

```
with Ada.Text_IO;

Ada.Text_IO.Put_Line("1st_loop");
for I in 1 .. 3 loop
  Ada.Text_IO.Put(Integer'Image(I));
end loop;

Ada.Text_IO.Put_Line("2nd_loop");
for I in 3 .. 1 loop
  Ada.Text_IO.Put(Integer'Image(I));
end loop;

Ada.Text_IO.Put_Line("3rd_loop");
for I in reverse 1 .. 3 loop
  Ada.Text_IO.Put(Integer'Image(I));
end loop;

Ada.Text_IO.Put_Line("4th_loop");
for I in reverse 3 .. 1 loop
  Ada.Text_IO.Put(Integer'Image(I));
end loop;
```

# Control Structures

Does the compiler generate warnings or error messages? If so, why

```
with Ada.Text_IO;

Ada.Text_IO.Put_Line("1st_loop");
for I in 1 .. 3 loop
  Ada.Text_IO.Put(Integer' Image(I));
end loop;

Ada.Text_IO.Put_Line("2nd_loop");
for I in 3 .. 1 loop
-- warning: loop range is null, loop will not execute
  Ada.Text_IO.Put(Integer' Image(I));
end loop;

Ada.Text_IO.Put_Line("3rd_loop");
for I in reverse 1 .. 3 loop
  Ada.Text_IO.Put(Integer' Image(I));
end loop;

Ada.Text_IO.Put_Line("4th_loop");
for I in reverse 3 .. 1 loop
-- warning: loop range is null, loop will not execute
  Ada.Text_IO.Put(Integer' Image(I));
end loop;
```

- Does not compile when treating warnings as errors (`-gnatwe`)

# Mini Projects

- Apply by email
- Students who are going to present mini projects will be notified via email and on the website

# Next Steps

- Types: Records
- Containers: Arrays, Vectors
- Abstraction: Packages
- Abstraction: Generics
- Error Handling, Randomization, Enums, etc.

# Arrays and Containers

```
with Ada.Containers.Ordered_Maps;
with Ada.Containers.Vectors;

generic
  type Item_Type is private;
  type Key_Type is private;
  with function "<" (A, B: Key_Type) return Boolean is <>;
  with function "=" (A, B: Item_Type) return Boolean is <>;
package Foo is
  type Item_Array is array (Natural range <>) of Item_Type;
  procedure Add_Item (Item: Item_Type);
  procedure Add_Item_To_Map (Item: Item_Type; Key: Key_Type);
private
  package Item_Vectors is new Ada.Containers.Vectors (
    Element_Type => Item_Type,
    Index_Type => Natural);
  use Item_Vectors;
  Item_Vector_Instance: Item_Vectors.Vector;
  -- A package-wide instance of an Item_Vectors.

  package Item_Maps is new Ada.Containers.Ordered_Maps
    (Element_Type => Item_Type,
     Key_Type      => Key_Type,
     "<"           => "<",
     "="          => "=");
  use Item_Maps;
  Item_Map_Instance: Item_Maps.Map;
  -- A package-wide instance of an Item_Maps.
end Foo;
```



# Pre- and Post-Conditions

- `-gnat12` activates Ada2012 features
- `-gnata` activates assertions
- Pragas to really activate assertions:

```
pragma Assertion_Policy(Pre => Check);  
pragma Assertion_Policy(Post => Check);
```

Questions?