

## 1. Namebinding

a.) dyn. namebinding:

- (1) legal, output: NT
- (2) legal
- (3) legal, output: NT
- (4) legal

b.) stat. namebinding:

- (1) legal, output: NT
- (2) illegal – static compiler diagnostic: there is no method named “M2” for class T
- (3) legal, output: NT
- (4) legal

c.) advantages of static namebinding

error reporting at compile time, no run-time checks (Smalltalk: Message not understood at run-time) -> fewer errors in production code

d.) overloading vs. redefinition

see page 109 in the scriptum (7.5.1 Redefinition vs. Overloading)

e.) C++:

- return\_type -> invariant
- controlling argument -> covariant
- parameter\_types -> invariant

## 2. Exceptions

```
1 with Ada.Text_IO; use Ada.Text_IO;
2
3 procedure Main is
4
5     E1, E2, E3 : exception;
6
7     procedure Q is
8     begin
9         raise E2;
10    exception
11        when E1 => Put_Line ("Q: E1"); raise E3;
12        when others => Put_Line ("Q: other exception"); raise E1;
13    end Q;
14
15    procedure P is
16    begin
17        begin
18            Q;
19        exception
```

```

20     when E1 => Put_Line ("P/1: E1"); Q; raise E2;
21 end;
22 begin
23     Q;
24 exception
25     when others => Put_Line ("P/2: other exception"); raise E1;
26 end;
27 exception
28     when E1 => Put_Line ("P/3: E1"); raise;
29             Put_Line ("P/3: after raise");
30
31     when E2 => Put_Line ("P/3: E2"); raise E3;
32 end P;
33
34 begin
35     P;
36 exception
37     when E1 => Put_Line ("Main: E1");
38     when E2 => Put_Line ("Main: E2");
39     when E3 => Put_Line ("Main: E3");
40     when others => Put_Line ("Main: other exception");
41 end Main;

```

The order of lines executed is (begin and end lines are included for normal, i.e. non-exception procedure entry and exit):

34, 35, 16, 17, 18, 8, 9, 12 (with output), 20 (with output),  
8, 9, 12 (with output), 28 (with output), 37 (with output), 41

The output is:

```

Q: other exception
P/1: E1
Q: other exception
P/3: E1
Main: E1

```