ETH Zürich
Chair of Software Engineering

Einführung in die Programmierung
Introduction to Programming

Prof. Dr. Bertrand Meyer

Exercise Session 3

Today

- About submitting bug reports
- Important Concepts from the Lectures
- Programming in Eiffel
  - Object Creation, Routine Calls and Stacks
  - Points and Circles

Important Concepts from the Lectures

Command or Query?

- 1. Feature `name`, as in `Zurich.name`. Q
- 2. Feature `buildings`, as in `Zurich.buildings`. Q
- 3. Feature `add_line`, as in `Zurich.add_line (2, "tram")`. C
- 4. Feature `connecting_lines`, as in `Zurich.connecting_lines (central, polyterrasse)`. Q
- 5. Feature `move_all`, as in `Zurich.move_all (0.5)`. C
- 6. Feature `north`, as in `Zurich.north`. Q

Contract and Logic

- Contract
  - Precondition
  - Postcondition
  - Class invariant

- Logic
  - Truth assignment
  - Tautology
  - Contradiction
  - Ordinary vs. semistrict boolean operators
    - and vs. and then
    - or vs. or else

Programming in Eiffel

Object Creation, Routine Calls and Stacks

```
class CIRCLE

feature -- Access
  radius: REAL
      -- Radius of the circle.

feature -- Query
  area: REAL
      -- Area of the circle.
    do
      Result := 3.14 * radius ^ 2
    end

feature -- Status set
  set_radius (a_radius: REAL)
      -- Set `radius' with.
    do
      radius := a_radius
    end

end
```

```
class APPLICATION

create make

feature -- Initialization
  make
      -- APPLICATION ENTRY POINT.
    local
      l_circle: CIRCLE
      l_area: REAL
    do
      create l_circle
      l_circle.set_radius (1.0)
      l_area := l_circle.area

      io.put_string ("The area is ")
      io.put_real (l_area)
      io.put_string (".")
    end
end
```

Dynamic View

- At runtime (i.e., during the program execution), we have a set of objects (instances) created from the classes (types).

- The creation of an object implies that a piece of memory is allocated in the computer to represent the object itself.

- Objects interact with each other by calling features on each other.

Who are Adam and Eve?

- Who creates the first object?
  - The runtime creates a so-called root object, which then creates other objects.
  - You define the type of the root object in the project settings.

- How is the root object created?
  - The runtime calls a creation procedure of the root object.
  - You define this creation procedure in the project settings.
  - The application exits at the end of this creation procedure.

Points and Circles

- Write three classes POINT, CIRCLE, and APP
  - Class POINT has two attributes x and y of type REAL
  - Class CIRCLE has one attribute center of type POINT and another attribute radius of REAL
  - Class APP is the root class, and make is its root feature. Feature make needs to
    - initialize a point and a circle (with arbitrary states);
    - move the point to a new coordinate;
    - move the circle to a new location;
    - change the radius of the circle;
    - compute the area of the circle;
  - Decide by yourself what other features each class needs.

~ End ~