ETH Zürich

Chair of Software Engineering

Einführung in die Programmierung
Introduction to Programming

Prof. Dr. Bertrand Meyer

Exercise Session 4

Today

- Important concepts from the assignment
  - Calling a routine with precondition

- Programming in Eiffel
  - Points and circles

Calling a routine with precondition

```
class BUSINESS_CARD
create fill_in

feature {NONE} -- Initialization
  fill_in
    do
      …
```

```
Io.read_integer
set_age (Io.last_integer)
```

✗

```
      …
    end

feature -- Access
  age: INTEGER

feature -- Setting
  set_age (a_age: INTEGER)
    require
      age_non_negative: a_age >= 0
    do
      …
    end
```

```
Io.read_integer
if Io.last_integer >= 0 then
  set_age (Io.last_integer)
else
  -- abort
end
```

✓

- *A caller needs to guarantee a callee is only invoked in states where the callee's precondition is satisfied*
- *A routine needs to guarantee, at its exit, the postcondition is established*

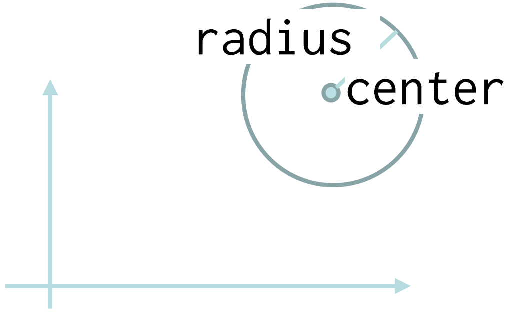
Programming in Eiffel

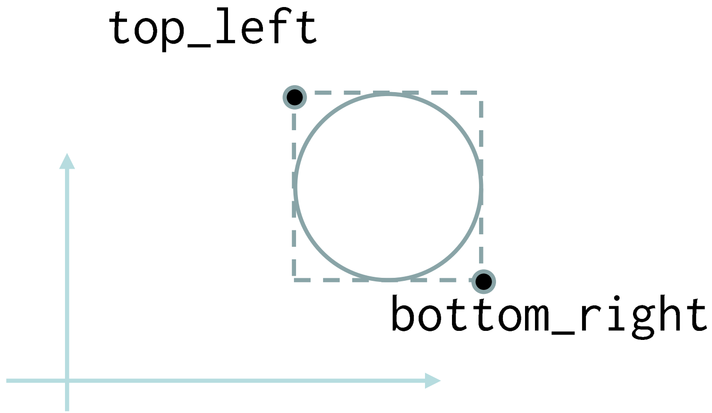
Points and Circles

- Write two classes `POINT` and `CIRCLE`, with which we can
  - initialize a point `p` at a specific location `(x, y)`,
  - get the `x` and `y` coordinates of `p`,
  - set the `x` and `y` coordinates of `p`,
  - print the state of `p` in format
    `(x=1, y=2)`,
  - initialize a circle `c` using its center point and its radius,
  - get the `center` and `radius` of c,
  - set the `center` and `radius` of c,
  - print the area of `c`, and
  - print the state of `c` in format
    `(center=(x=1, y=2), radius=1)`.

Change Is the Only Constant

- Instead of using the `center` and `radius`, you are asked to use the `top_left` and `bottom_right` vertices of its enclosing square to represent a circle.
  Please implement this change in `CIRCLE` without modifying any client/test code.





~ End ~