

Lecture 9: Application Programming Interfaces I

Instructor: Musard Balliu, musard@chalmers.se

<http://www.cse.chalmers.se/~musard>

QUESTIONS?

Plan

- ▶ Last time

1. packages

2. modifiers

- ▶ Today's Plan:

1. Robot APIs

Objectives

Object-oriented programming is not at its best in small-scale programming (what we called “algorithm design” in Lecture 5).

To practice OOP we need to have a system of at least modest size.

B.W. Becker’s book *Java: Learning to Program with Robots* provides an excellent such system.

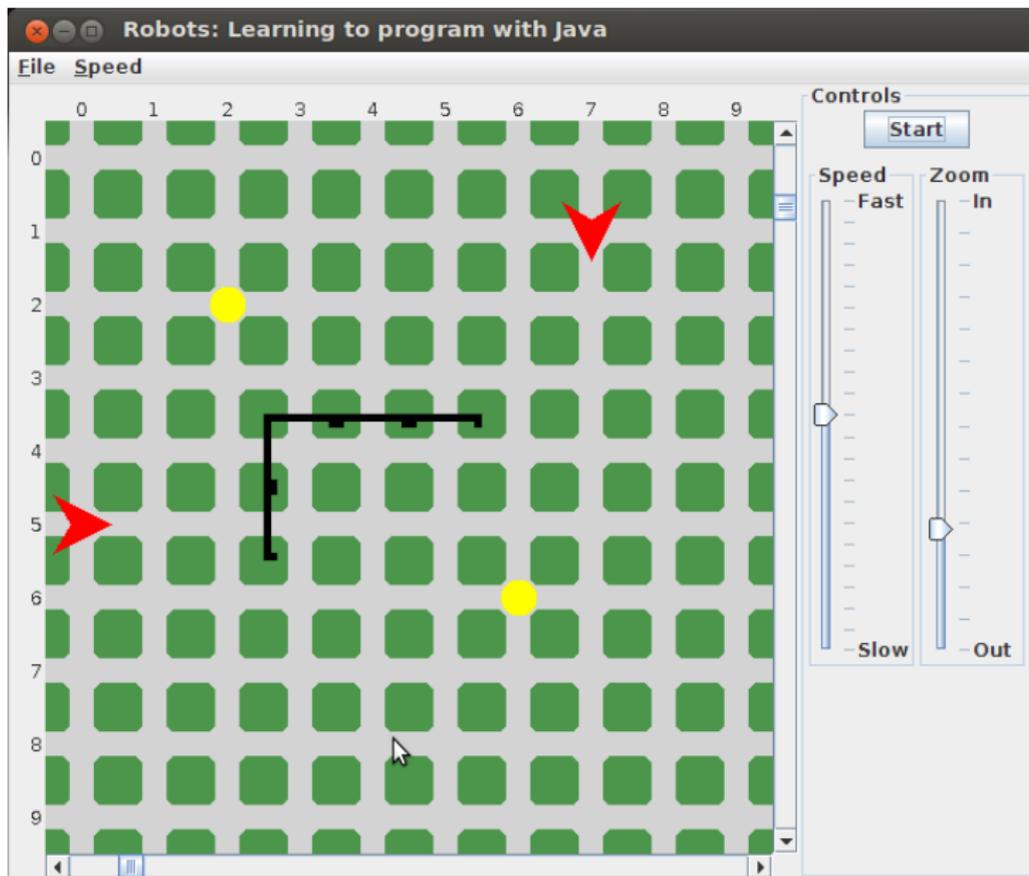
Objectives

Working with this system will enable us to

- ▶ experiment with OOP design;
- ▶ learn how to read documentation;
- ▶ learn how to work with an existing codebase.

All of these are vital in your future studies and career.

The Robot World



The Robot class

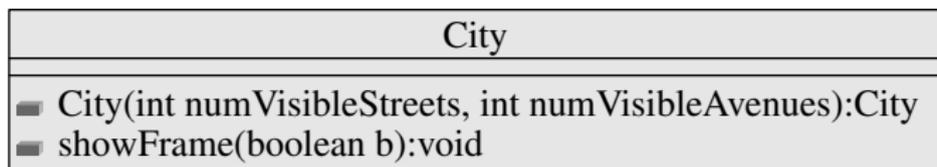
Robot	
🔒	int street
🔒	int avenue
🔒	int numThings
🔒	Direction direction
🔒	Robot(City city, int street, int avenue, Direction direction):Robot
🔒	move():void
🔒	pickThing():void
🔒	turnLeft():void
🔒	...

Documentation

The City class

All Robot constructors require a `City`.

The `City` class does not seem very interesting.



Documentation

We need to experiment a bit with `City`.

Initial code

Putting Things in a City

A city can contain

- ▶ generic Things
- ▶ Walls
- ▶ and, of course, Robots.

A first example

Creating a world

It's annoying to have to redo the creation every time.

It is better to package everything in a new kind of City!

A first world

Experimenting with Robot

- ▶ `move`, `turnLeft`, `pickThing`

- ▶ limited, explodes, not good

Initial code

Extending Robot

- ▶ New methods: `move(n)`, `turnRight`, `turnAround`

- ▶ doesn't explode

SmartRobot

Changing Things

The elements of the robot world can be customized: we can change the color, add labels, etc. Check the documentation!

RandomRobot

Having to write the move code for the robot is boring: we can add a “move about randomly” method.

RandomRobot

Homework

Read the first 3 chapters of Becker's book.