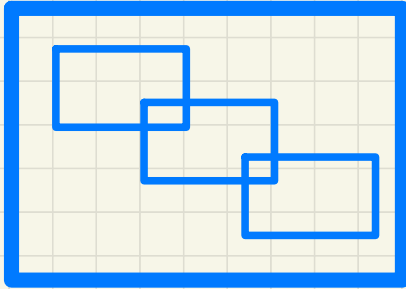


Separation of Concerns

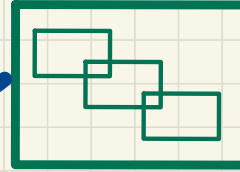
Slide 6

model



- Classes & Methods
- Methods
 - * constructors
 - * accessors: **return** statements
 - * mutators: **no return** statements
 - * containing **no** print statements

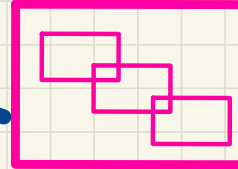
junit_tests



use

- Expected vs. Actual Values
- Methods
 - * calling methods from model
 - * assertions
 - * containing **no** print statements

console_apps

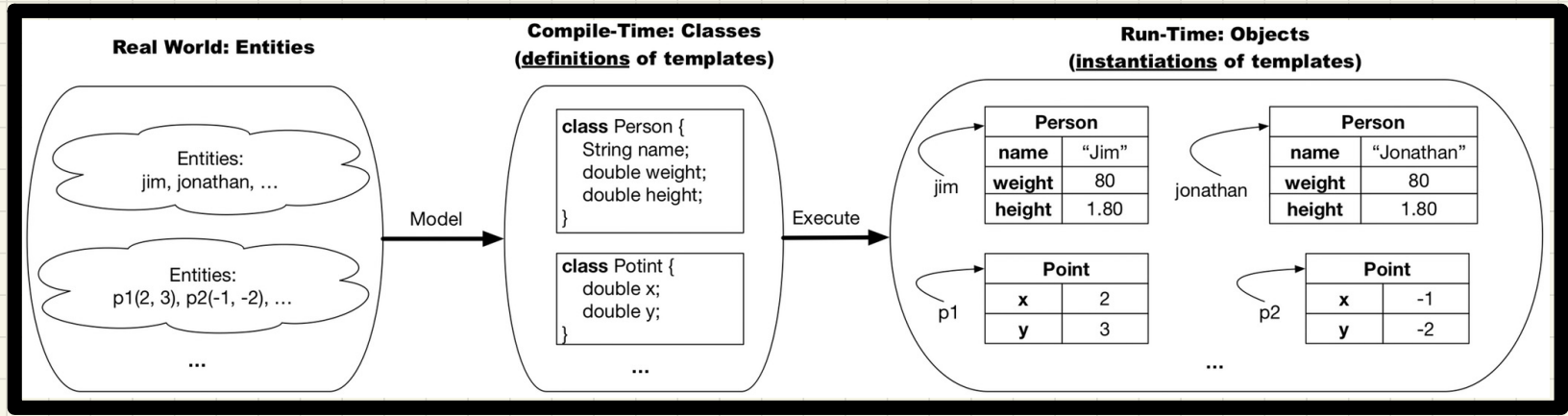


use

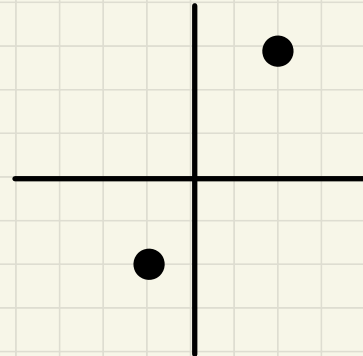
- main method (entry point of execution)
 - * reading inputs from keyboard
 - * calling methods from model
 - * producing outputs to console (print)
 - * containing **no** return statements

Observe-Model-Execute Process

Slide 7



Entities:
Attributes:
Changes:
Inquiries:
Template:



Entities:
Attributes:
Changes:
Inquiries:
Template:

Object Oriented Programming (OOP)

Slide 8

- Templates (compile-time Java classes)
 - + attributes (common around instances)
 - + methods
 - * constructors
 - * accessors/getters
 - * mutators/setters
 - + Eclipse: Refactoring
- Instances/Entities (runtime objects)
 - + instance-specific attribute values
 - + calling constructor to create objects
 - + using the "dot notation", with the right contexts, to:
 - * get attribute values
 - * call accessors or mutators

Modelling: from Entities to Classes

Identify Critical Nouns & Verbs

Example 1

Points on a two-dimensional plane are identified by their signed distances from the X- and Y-axes. A point may move arbitrarily towards any direction on the plane. Given two points, we are often interested in knowing the distance between them.

Example 2

A person is a being, such as a human, that has certain attributes and behaviour constituting personhood: a person ages and grows on their heights and weights.

OO Thinking: Templates vs. Instances

Templates

Common

Attribute Definitions
(Types)

Common

Behaviour Definitions
(Headers/API)

Instance-Specific

Attribute Values

Instance-Specific

Behaviour Occurrence