

CSC207 - Java Classes

Ilir Dema

Summer 2016

A problem: The Weresquirrel

Now and then the young programmer Jacques transforms into a squirrel. In order figure out his condition, Jacques has kept a diary of his daily activities. Start by helping Jacques compile and analyze his diary!



Jacques diary

- ▶ In order to solve a problem using software, one should be able to solve the problem "by hand".
- ▶ That is, one should be able to prescribe an *algorithm* that solves the problem.
- ▶ An **algorithm** is a self-contained formal set of computational steps to be performed in order to solve a problem.
- ▶ Algorithms can be programmed using various programming languages and as such executed by computers.
- ▶ The starting point is the *data set*.
- ▶ Jacques data:





Day 1: False, cauliflower, ice cream, brushed
teeth, cycling, work

Day 2: True, spaghetti, peanuts, computer, weekend

Day 3: False, potatoes, ice cream, brushed
teeth, computer, weekend

Next Step - Data Analysis

$$\phi = \frac{n_{11}n_{00} - n_{10}n_{01}}{\sqrt{(n_{10} + n_{11})(n_{00} + n_{01})(n_{01} + n_{11})(n_{00} + n_{10})}}$$

 No pizza, no squirrel 76	 Pizza, no squirrel 9
 No pizza, squirrel 4	 Pizza, squirrel 1

Objects, Data Types and Classes.

- ▶ **Object** is a general term that stands for many things.
 - ▶ Examples of objects: a user, a circle, a car.
 - ▶ Objects have **properties** and **behaviours**.
 - ▶ A user has a **ID**, a **password**, and a **profile**.
 - ▶ Also a user has behaviours like **changePassword**.
 - ▶ Note properties can be simple data (ID, ...),
 - ▶ also properties can be complex objects (**profile**).
- ▶ A **Data Type** is a set (or a collection) of objects or values of the same nature.
 - ▶ Examples: integers, characters, users, ...
 - ▶ In Java we have **primitive** data types (integers, characters, booleans, ...)
 - ▶ and **reference** data types (users, cars, ...).
- ▶ A **class** is a blueprint specifying a complex data type. We do also refer by the term class to the collection of the objects specified by the said blueprint.

Java versus Python

- ▶ In Python
 - ▶ values have types
 - ▶ variables do not have types, they are just names
 - ▶ just try `print((3.1).as_integer_ratio())`
- ▶ In Java, variables have types
- ▶ In Python, a statement is a program.
- ▶ In Java, it is not:
 - ▶ need a class
 - ▶ need a main method

Defining Classes in Java

- ▶ Instance variables

```
public class Circle {  
    private int radius;  
}
```

- ▶ `radius` **is an instance variable**. Each object/instance of the `Circle` class has its own `radius` variable.
- ▶ `this` is an instance variable that you get without declaring it.
- ▶ It is like `self` in Python.
- ▶ Its value is the address of the object whose method has been called.

Constructors

- ▶ A constructor has:
 - ▶ the same name as the class
 - ▶ no return type (not even void)
- ▶ A class can have multiple constructors, as long as their signatures are different (overloading).
- ▶ If you define no constructors, the compiler supplies one with no parameters and no body (default constructor).
- ▶ If you define any constructor for a class, the compiler will no longer supply the default constructor.

Methods

- ▶ A method must have a return type declared. Use void if nothing is returned.

- ▶ The form of a return statement:

```
return expression;
```

If the expression is omitted or if the end of the method is reached without executing a return statement, nothing is returned.

- ▶ Must specify the accessibility. For now:
 - ▶ `public` - callable from anywhere
 - ▶ `private` - callable only from this class
- ▶ Variables declared in a method are local to that method.

Parameters

- ▶ When passing an argument to a method, you pass what's in the variable's box:
 - ▶ For class types, you are passing a reference. (Like in Python.)
 - ▶ For primitive types, you are passing a value. (Python can't do anything like this.)
- ▶ This has important implications!
- ▶ You must be aware of whether you are passing a primitive or object.

Access Modifiers

- ▶ Classes can be declared public or package-private.
- ▶ Members of classes can be declared public, protected, package-protected, or private.

Modifier	Class	Package	Subclass	World
<code>public</code>	Yes	Yes	Yes	Yes
<code>protected</code>	Yes	Yes	Yes	No
<code>default (package private)</code>	Yes	Yes	No	No
<code>private</code>	Yes	No	No	No