

Seminar 1: Designing a class

24292-Object Oriented Programming

1 Introduction

The objective of this seminar is to learn how to design a class, the fundamental unit in object-oriented programming. Each class has a number of attributes and methods that are to be set to complete the design. The seminar will consist of two design exercises that each requires the definition of a new class. In the first exercise you will design a class representing football players. In the second exercise you will design a class representing football teams.

For each attribute that you define you must specify:

- its visibility (public or private)
- its type (integer, real, char, string, ...)
- its name

For each method that you define you must specify:

- its visibility (public or private)
- its name
- its parameters including name and type
- its return type (the method may not return anything)

Note that it is *not* necessary to implement the code of each method during the seminar (the code will instead be implemented in the lab session).

Remember that a constructor method is called each time an instance of that class is created. Each instance can have different attribute values and the constructor method must initialize / assign them. Moreover, we have methods to access to class attributes: a "getter" method returns the value of an attribute, and a "setter" method changes the value of an attribute.

The solution to these exercises will be implemented in Java during Laboratory session 1.

The classes and methods that will be designed and implemented in the seminar and the lab are meant to be part of a larger application that manages football leagues. Potentially, such an application could be used to import the results of football games, maintain statistics, print league tables and a list of top scorers, etc.

2 Review

- Explain with examples what is a class and what is an instance.
- Explain with an example the difference between a call to a function and a message sent to an instance in the Object Oriented Programming paradigm.
- Give an example of class.

3 Exercise 1: The Player class

In the first exercise we will design a class representing football players. Each football player has a gender, a name, an age and a nationality. In addition, a player has several associated statistics, which can be summarized as follows:

- Number of matches played
- Number of tackles
- Number of passes
- Number of shots
- Number of assists (i.e. passes leading to goals)
- Number of goals scored

It should be possible to find out the gender, name, age and nationality of a football player. In addition, the class should contain a method that updates a player's statistics based on a recent match. To do so, it is necessary to indicate how the player performed in the given match according to the relevant parameters (number of tackles, and so on).

4 Exercise 2: The Team class

In the second exercise we will design a class representing football teams. Each football team has a name, an associated country, and a gender classification (male, female or mixed). Each team also has a list of associated players. In addition, a team has several associated statistics, which can be summarized as follows:

- Number of matches played
- Number of wins
- Number of ties
- Number of losses

- Number of goals scored
- Number of goals against

It should be possible to find out the name, country and gender classification of a given team. It should be possible to add or remove players, as long as they are of the correct type (female teams can only have female players, male teams can only have male players, while mixed teams can have both). In addition, the class should contain a method that updates a team's statistics based on a recent match played. Again, to do so it is necessary to indicate how the team performed in the match.