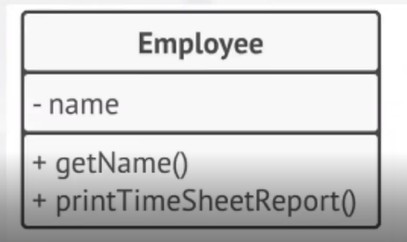
**1.1. Single Responsibility Principle:**

**1.** There should never more than one reason for a class to change. Every class should have only one responsibility.

**2.** The main goal of this principle is reducing complexity.

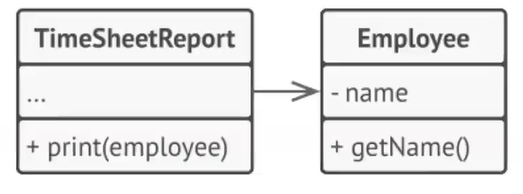
**3.** If a class does too many things, you must change it every time one of these things changes.

**- Example:**

In this example, we can see that the employee class contains the employee name. You have methods to obtain the employee name and a method that generates the employee timesheet report.

**- Problem:** In this example every time that we need to change the way to generate the employee timesheet report, then we will need to modify the Employee class.

**- Solution:**



We could create a new class "Reporter" in order to manage the timesheet reporting. With this solution, when the timesheet report changes, then the Employee class will not be affected.