

Laboratory in1278LR Introduction to Programming with Java
Delft University of Technology, Faculty EWI, Software Engineering Research Group.
Group = 3; Remainder = 2

Assignment 6

Create a new directory with the name Assignment6.

Use this directory to save all programs that you will create as part of this assignment.

Please read the text of the exercises fully before embarking upon them.

Part 1 Employees

- Create a new class `SalaryExtraEmployee` as a subclass of the `Employee` class created in chapter 9. The `SalaryExtraEmployee` will be paid like a normal `Employee` (an employee that is paid by the week/month), but will receive an extra payment based upon the number of work items produced during the past week (e.g.: normal production => 100%; extra production => 110%). The percentage should be stored as a *static* constant. Override all of the necessary methods for this subclass. Then modify the class `TestEmployee` to demonstrate the proper operation of all three subclasses.
- Apart from the organization sketched above, another place in the inheritance hierarchy for the newly developed subclass might be appropriate. *Designate* (NOT: code) the other inheritance structure of the classes and explain to the student assistant the pros and cons of both organizations (see also Figure 9.1 of the book).

Part 2 Bicycle renting

We will develop a bicycle rent shop where we will rent various kinds of bicycles to the public.

We start off with a set of basic bicycles. For each bicycle we will store a rental price per kilometer, the kilometers traveled during the last rental period and the total number of kilometers traveled.

Currently, we have a number of brands of bikes in store:

Brand	Model	Rent price	Kilometers last rental	Total kilometers
String	String	double	int	int
e.g.: Batavus	basic	0.06 / km	326	33,456

- You are asked to construct a class `BasicBike` modeling such a bicycle. Provide constructors, getters and setters for a bike.
- Provide a method `toString()` returning all relevant data from an object of this class.
- Provide a method `rentBike()` renting a bike and returning the number of kilometers traveled during the last rental period (a random number within a reasonable range). When a bike is returned, this number is to be added to the Total kilometers driven with that bike.
- Provide a method `pay4Bike()` calculating the price of the last rental period.
- Provide a method `totalPayment()` calculating the price of the total kilometers driven with a particular bike.

Currently, we have a number of brands of bikes in store:

Brand	Model	Rent price	Kilometers last rental	Total Kilometers
Batavus	basic	0.06 / km	-	-
Union	bare-bones	0.045 / km	-	-
Sparta	standard	0.065 / km	-	-

- Construct a class `TestBike` in which you instantiate a number of bikes of various brands. Note that we will have zero or more bikes of each brand available. Rent several bikes and check for appropriate values after the bikes have been returned.

As business develops, we want to store our bikes in a `Vector`.

- Create a vector and store all bikes in the vector.
- Create a method to sum-up and print the total kilometers driven for all the bikes in the vector.

- Create a method to visit all the elements in the array and have *toString()* executed for them.

Customers indicate that they want more luxury (and more expensive) bikes: bikes having a gearing. We extend the class **BasicBike** into a class **LuxuryBike** adding bikes with a gearing (an int number indicating the number of gears). See the following table:

Brand	Model	Rent price	Kilometers last rental	Total Kilometers	Gearing
Batavus	luxus	0.09 / km	-	-	24
Giant	plus	0.11 / km	-	-	27
RIH	extra	0.045 / km	-	-	21

- Provide your class **LuxuryBike** with appropriate constructors, getters and setters. Redefine methods as needed.
- Adapt your class **TestBike** to store several luxury bikes in the vector.
- Redo the method to sum-up and print the total kilometers driven for all the bikes in the vector.
- Redo the method to visit all the elements in the array and have *toString()* executed for them.

Customers now indicate that they also want electrical bikes. We will store the kind of battery used (a String).

Brand	Model	Rent price	Kilometers last rental	Total Kilometers	Gearing	Battery
Batavus	speedy	0.13 / km	-	-	24	Nicad
Sparta	ion	0.145 / km	-	-	27	Lithium
RIH	volta	0.095 / km	-	-	21	Fuel cell

Extend the class **LuxuryBike** to a class **ElectricBike** and provide appropriate constructors, getters and setters. Redefine methods as needed.

- Adapt your class **TestBike** to store several electrical bikes in the vector.
- Redo the method to sum-up and print the total kilometers driven for all the bikes in the vector.
- Redo the method to visit all the elements in the vector and have *toString()* executed for them.
- Explain to the student assistant how inheritance has been used in your code and which advantages it gives.