

Instruction – Inheritance

8.1 LetterPlus

Consider the class `Letter` from instruction 6. Go to the folder where the file `Letter.java` is or download it again if you have lost it.

We are going to add functionality to `Letter` objects, not by editing the class `Letter`, but by adding a subclass.

1. Create a file `LetterPlus.java` with a class definition with the same name. Have this class `LetterPlus` extend `Letter`. Objects of this class will be able to show complete words by showing them letter by letter. To achieve this, add the following:
 - (a) an instance variable `word` that contains the word to be shown;
 - (b) a constructor that initializes this word;
 - (c) a method `void play()` that shows the letters of this word one after each other with a 0.5 seconds interval; use `setText` and `pause`.
2. Test it via the Interactions pane of DrJava.

8.2 Vehicles

We design a set of classes for the administration of vehicles. In the following parts, test each step. Create objects in the Interactions pane of DrJava and call methods on them.

1. Create a class `Vehicle` with instance variable `registrationNumber`. Add a constructor that initializes this variable. Add a method `print` that prints the data of the object, e.g.,

```
Vehicle with reg.nr. 123456
```
2. Create a class `Car` that is a subclass of `Vehicle` and has an additional instance variable `fuelType`. Add an appropriate constructor and override the method `print`. This method should print something like

```
Vehicle with reg.nr. 654321
Vehicle type: Car
fuel type: petrol
```

Use a call to `super` in this method.
3. Add a class `Motorcycle`, subclass of `Vehicle`, with an additional instance variable that records whether the motorcycle has a sidecar (Dutch: *zijspan*) or not. Add an appropriate constructor and override `print`.
4. Add an instance variable `weight` to the class `Vehicle`. Adapt the method `print` accordingly. Observe that the `print` methods of `Car` and `Motorcycle` have the correct behavior, although you didn't change them.
5. Create a class `VehicleAdministration` with an instance variable `Vehicle[] vehicles`. Add a method `printVehicles` that prints the data of each `Vehicle` in the array. Create an array with two vehicles and test. Now create an array of 4 elements with the two `Vehicles` and a `Car` and a `Motorcycle`. Your `print` method should work for this as well, without modification. Explain.
6. Add a class `Truck`. Change the `print` method (having it print the vehicle type `Truck`). Since it doesn't have additional instance variables, it doesn't seem necessary to add a constructor. It doesn't compile. Why not? Add a constructor.