

World of Robots

Master 1 Génie Physiologique et Informatique

Schedules



Group 14 : Laurène DONG, Marion DOSPITAL, Laurantine LAIR, Thibault NIGGEL, Florian PAILLAS, Ophélie SOCHARD

In general, inheritance allows for the compartmentalization and reuse of previously written and used code, so that one thing inherits from another. In our example, there is an inheritance relationship between the 3 classes called subclasses, namely "TortueRapide," "TortueCouleur," and "TortueSavante," and the superclass called "TurtleV2." In other words, the 4 subclasses inherit from the superclass. This inheritance relationship is also visible in the code because we used the "extends" keyword, which allows us to establish an inheritance relationship in Java: "public class TortueCouleur extends TurtleV2." Among the three types of inheritance that exist: extension, override, and overload, we are performing extension because we are adding new attributes and methods to expand the capabilities of the superclass and give the subclass a more specific behavior.

Indeed, the attributes and methods of the superclass "TurtleV2" are also found and used in the 4 subclasses: x and y position, direction, drawing, displaying the state, displaying the turtle, and moving forward. However, each subclass has an additional characteristic that makes them more special:

- TortueSavante: methods to make it speak in addition.
- TortueRapide: specific methods because its way of moving is different. It can accelerate and decelerate, unlike the others.
- TortueCouleur: methods for creating colored tracers.

In this project, inheritance allows us to save lines of code across different classes, reuse the code from the superclass because many methods and attributes are common among the 4 classes, anticipate code changes if any, and highlight the specificity of each subclass.