02/10/2024

Feedback on inheritance

Group 5 Members :

DEVOUCOUX Grégoire (Scribe) - LASFAR Nisrine - HACQUART Virgile - ROUIBAH Bilal (Group Leader) - ILEKTI Linda (Time Keeper)

Synthèse

Inheritance is a fundamental concept in object-oriented programming that allows one class to inherit properties and behaviors from another class named superclass. In the class inheriting the methods from the superclass, it allows us to have a smaller and clearer code with specific methods and attributes for this subclass.

This promotes code reuse, reduces duplication, and makes maintenance easier.

In the Turtle project, there is a superclass "Turtle," and three subclasses of turtles with specific characteristics: "FastTurtle," "SmartTurtle," and "ColorTurtle." There are two ways to name an inheritance relationship: specialization when moving from a subclass to the superclass, and generalization when moving from the superclass to the subclass. Abstraction is when we generalize, and refinement is when we specialize.
In the Java language, inheritance is identified by the "extends" keyword for example "public class FastTurtle extends Turtle".

On BlueJ, in Unified Modeling Language (UML), we can spot an inheritance by looking for a continuous arrow from the subclass to the superclass.

There are three ways to achieve inheritance:

- Extension: it involves adding new members (methods or attributes). It extends the capabilities of the superclass, making the subclass more specific. In the example of the Turtle project, it only involves “extension” because the all the subclasses turtles have the same characteristics as in the superclass but each has additional associated attributes and methods that are from the superclass TurtleG.
- Override: This means modifying a member. You can rewrite or reuse methods from the superclass.
- Overload: This is about duplicating a member by designing a new function with the same name as the superclass but with different parameters.

02/10/2024

In the SimpleSchool project we have several classes that illustrate the concept of inheritance well,for example the student and Teacher classes inherit from person, they add student and teacher specific functionality ,respectively, while reusing code from the person class.
The second example the Person class represents an individual with common attributes such as name, surname, and age. It serves as a basis for other classes.This allows students and teachers to share common characteristics while having specific methods.

Finally Inheritance is a powerful tool in object-oriented programming that, when used correctly, can significantly improve the quality and maintainability of code.