Polymorphism in Java

Manoj

Polymorphism in Java

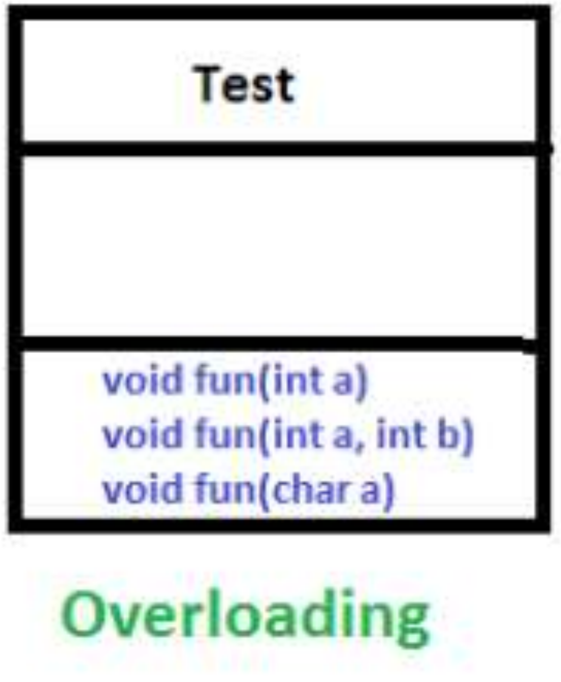
- poly means many or several and morph means faces/ behaviors or functionalities
- we can perform a *single action in different ways*.

Types of Java polymorphism

- Compile-time Polymorphism
- Runtime Polymorphism

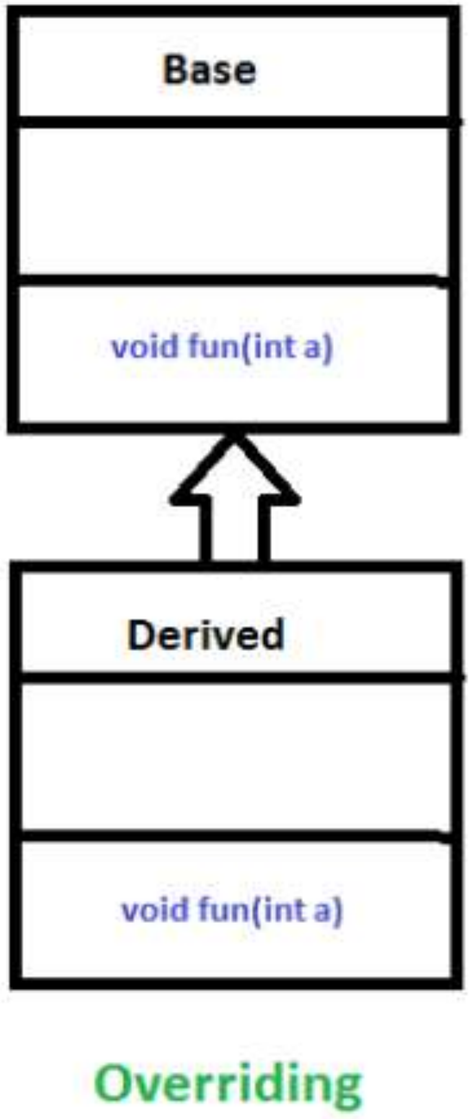
Compile-Time Polymorphism

- It is also known as static polymorphism.
- This type of polymorphism is achieved by method overloading



Runtime Polymorphism

- Also known as Dynamic Dynamic Binding or Dynamic Method Dispatch.
- The call to an overridden method is resolved dynamically at runtime
- This type of polymorphism is achieved by Method Overriding.



Method Overloading in Java

Method Overloading

- When a class has multiple methods with the same name, different parameter lists (different number or types of parameters), it is known as **Method Overloading**.

- Note : Method overloading in Java is not achieved by changing the return type.

Different ways to overload the method

- By changing number of arguments
- By changing the data type

`

Example

```
class MethodOverloading {
private static void display(int a){
System.out.println("Arguments: " + a);
}
private static void display(int a, int b){
System.out.println("Arguments: " + a + " and " + b);
}
public static void main(String[] args) {
display(1);
display(1, 4);
}
}
```

Method Overriding in Java

Method Overriding

- subclass has the same method as declared in the super class, it is known as **method overriding**.

Usage of Java Method Overriding

- Used to provide specific implementation of a method that is already provided by its super class.
- Used for runtime polymorphism

Rules for Java Method Overriding

- method must have same name as in the parent class
- method must have same parameter as in the parent class.
- must be IS-A relationship (inheritance)

Example of method overriding

```
class Vehicle{
void run()    {      System.out.println("Vehicle is running");    }
}
class Bike2 extends Vehicle{
void run()    {    System.out.println("Bike is running safely");    }

public static void main(String args[]){
Bike2 obj = new Bike2();
obj.run();
}
```

Out put

```
Bike is running safely
```

Thank you