16. Polymorphism with toString() method

- What are the advantages of Polymorphism?

Source code with Polymorphism is flexible, reusable, easy to read, understand and management. Polymorphism also supports the Open-Closed Principle, which states that classes should be open for extension but Closed for modification. Moreover, you can use polymorphism to handle objects of the same type in a general way, making data management and manipulation easier.

- How is Inheritance useful to achieve Polymorphism in Java?

+ Subclassing: Inheritance allows to create subclasses (also known as child class) that inherit attributes and behaviors from a superclass.

+ Method overriding: Subclasses can override methods defined in their superclass. When a method is called on an object of the subclass, the overridden method in the subclass is invoked, allowing for polymorphic behavior.

+ Code reusability: Inheritance promotes code reuse by allowing subclasses to inherit attributes and behaviors from their superclass. Polymorphism further enhances code reuse by allowing subclasses to provide specialized implementations of methods.

- What are the differences between Polymorphism and Inheritance in Java?

1. Purpose

+ Polymorphism: ability of objects of different types to be treated as objects of a common superclass.

+ Inheritance: a mechanism where a new class is derived from an existing class. It enables code reuse by allowing subclasses to inherit attributes and behaviors from their superclass.

2. Relationship

+ Polymorphism: While inheritance facilitates polymorphism by allowing subclasses to override superclass methods, polymorphism can also be achieved through interfaces, method overloading, and other mechanisms.

+ Inheritance: Inheritance is the mechanism by which one class inherits properties and behaviors from another class. It establishes a hierarchical relationship between classes, where subclasses inherit members from their superclass.

3. Usage

+ Polymorphism: Polymorphism is primarily used to achieve flexibility and dynamic behavior in method invocation. It allows for code to be written in a more general way, where specific implementations are determined at runtime based on the actual types of objects.

+ Inheritance: nheritance is used for code reuse, where common attributes and behaviors are defined in a superclass and inherited by subclasses. It promotes code organization, abstraction, and modularity by facilitating the creation of specialized subclasses.