

Topic 0:

Inheritance and Polymorphism

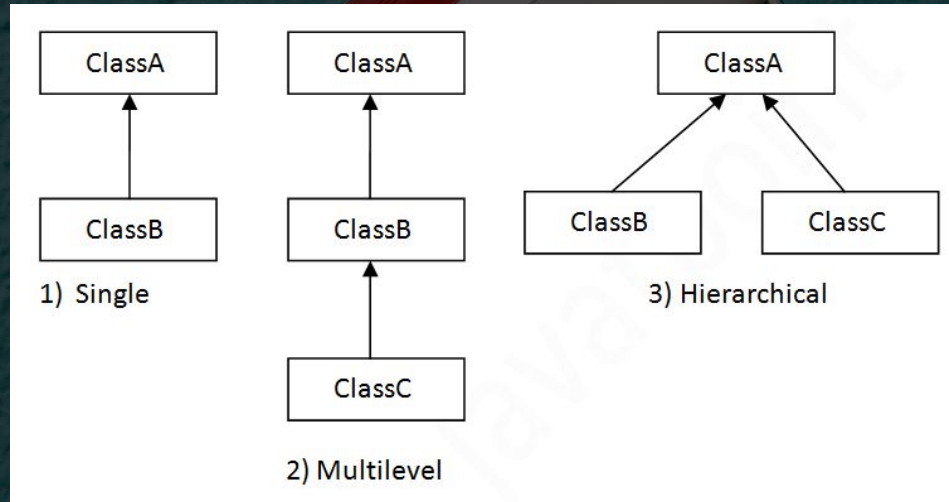
Inheritance

One object acquires all the properties and behaviors of a parent object

The extends keyword indicates that you are making a new class that derives from an existing class.

The meaning of "extends" is to increase the functionality of a class through:

- Code Reusability
- Method Overriding



Polymorphism

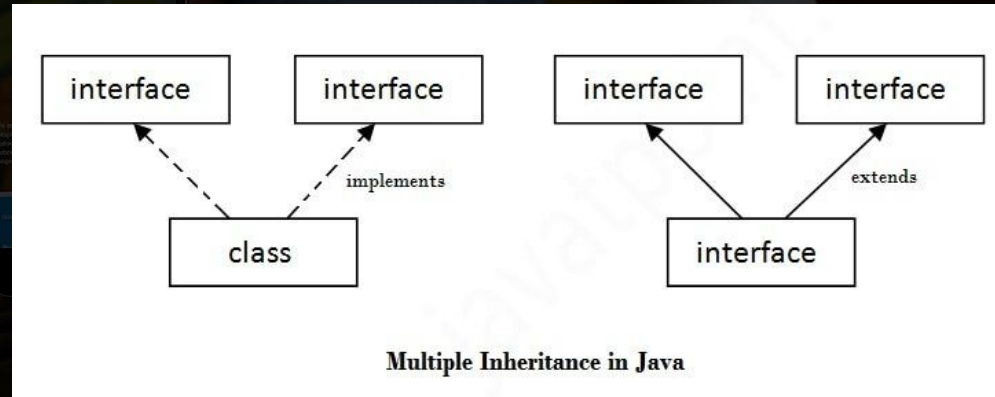
We can perform a single action in different ways

It puts more focus on expected actions more than exact implementations

It is commonly associated with Interface design.

Interfaces are “implemented” into classes

Interfaces can also be “extended” from other interfaces



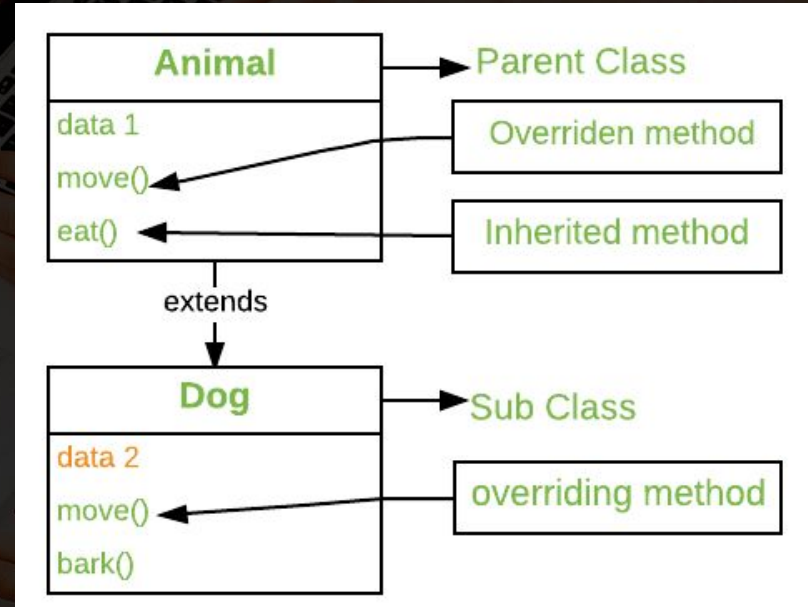
Polymorphism

Method Overriding is key for Runtime Polymorphism

Overriding allows a subclass to provide a specific implementation of a method that is already provided by one of its super-classes

It is the type of the object being referred to that determines which version of an overridden method will be executed

Static and Final methods can not be overridden



Abstract classes and Abstract methods

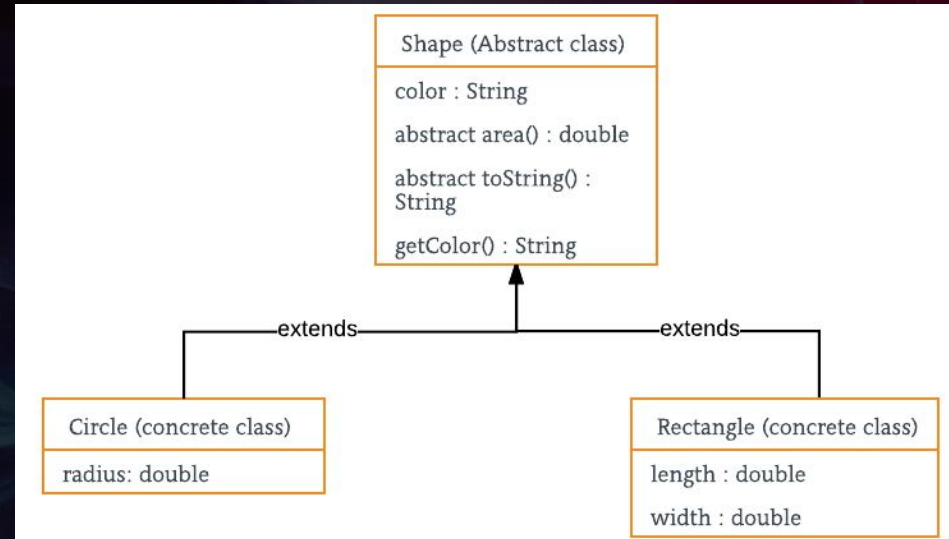
Both are declared with the “abstract” keyword

An abstract class is a class that is declared with abstract keyword.

An abstract method is a method that is declared without an implementation.

An abstract class may or may not have all abstract methods. Some of them can be concrete methods

An abstract class can not be directly instantiated with the new operator.



Interfaces

Shift the focus from state of an Object to its expected behaviour

Interfaces specify what a class must do and not how.

All the methods in interface are declared with empty body and are public and all fields are public, static and final

If a class implements an interface and does not provide method bodies for all functions specified in the interface, it must be declared abstract

```
interface Moveable
```

```
int AVERAGE-SPEED=40;  
void move();
```

what you declare

```
interface Moveable
```

```
public static final int AVERAGE-SPEED=40;  
public abstract void move();
```

what the compiler sees