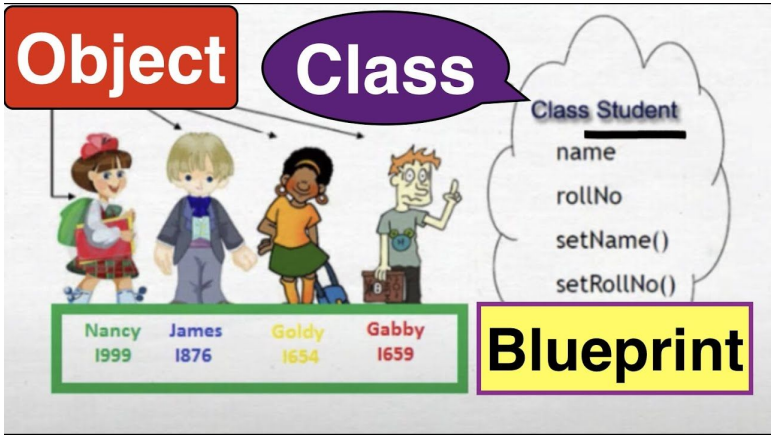
Java

Topics

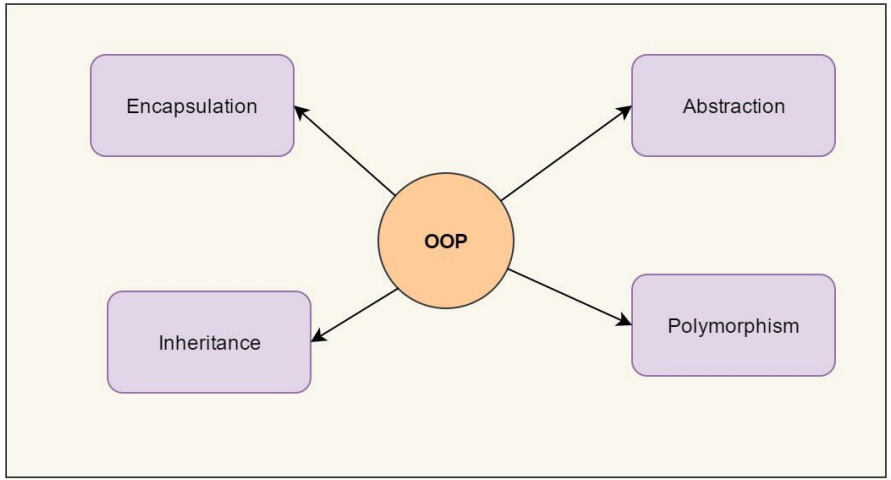
- Java OOPs Concepts
- Exception Handling in Java
- Java Collection
- Streams and File Handling in Java

Java Object-oriented programming System

- **Object** - It is a basic unit of Object Oriented Programming and represents the real life entities. Object wraps data and some functionality in it.
- **Class -** A class is a user defined blueprint or prototype from which objects are created

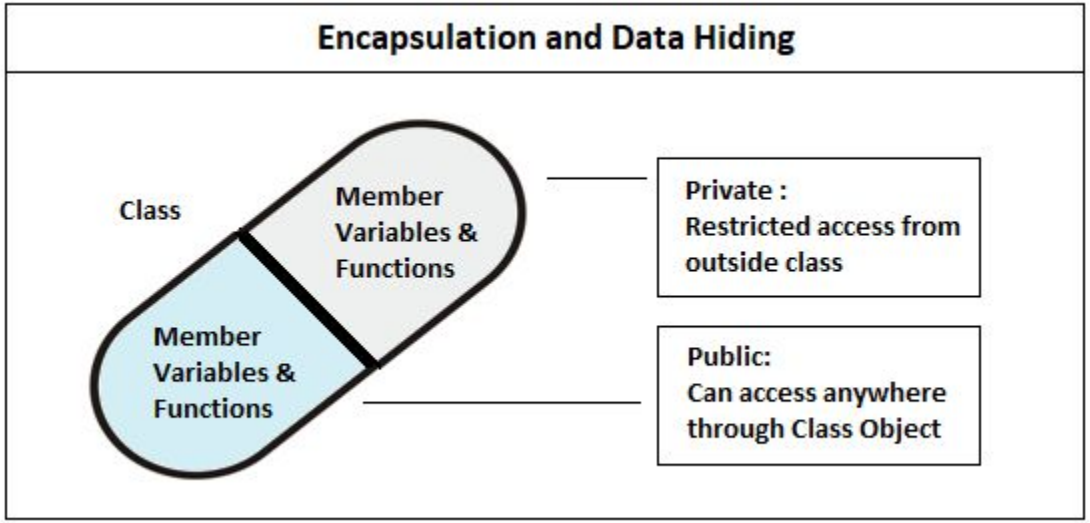


OOPS principal

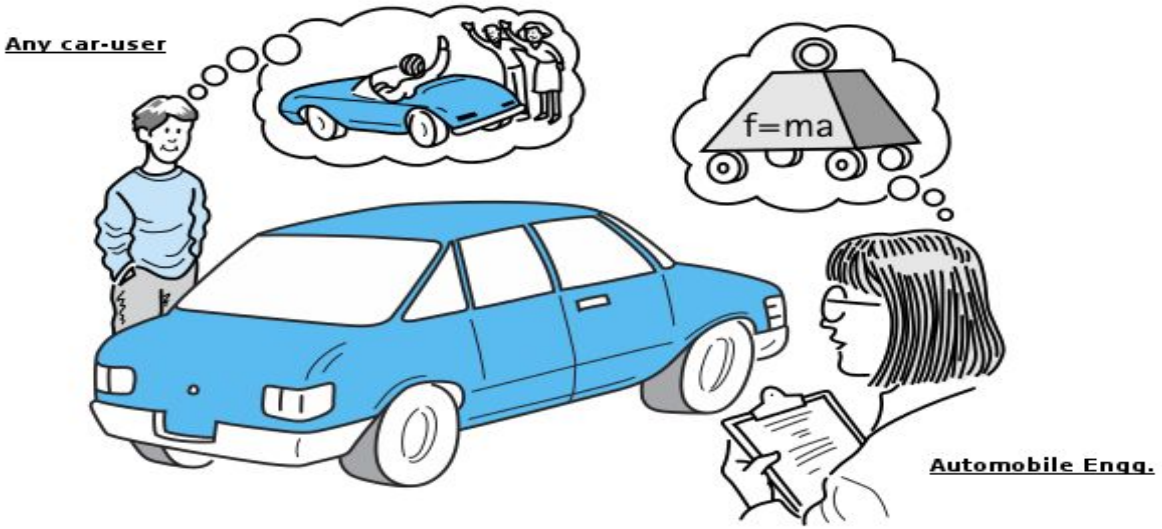


Four Pillars of Object Oriented Programming

Encapsulation



Abstraction



An abstraction includes the essential details relative to the perspective of the viewer

Abstract Class

- It cannot be instantiated.
- It can contain concrete methods and abstract method

When to use abstract class

When to use an abstract class

- An abstract class is a good choice if we are using the inheritance concept since it provides a common base class implementation to derived classes.
- An abstract class is also good if we want to declare non-public members. In an interface, all methods must be public.
- If we want to provide common, implemented functionality among all implementations of our component, use an abstract class.

Interface

- **Java interface** is an `abstract` class which provides data abstraction. Abstraction is a way to provide users with only relevant and essential information and hide the rest.
- Java interface is an **abstract class** which is used to group related methods, classes or data with empty bodies.
- Has static constant and abstract method

When to use interfaces

When to use an interface

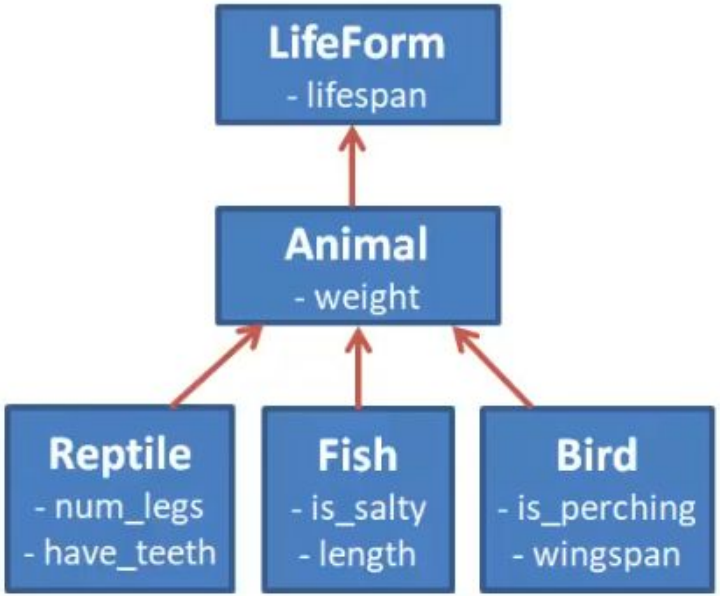
- Abstract classes should be used primarily for objects that are closely related, whereas interfaces are best suited for providing a common functionality to unrelated classes.
- Example : An interface named IFly. Can be implemented by a Bird class and Aeroplane class.

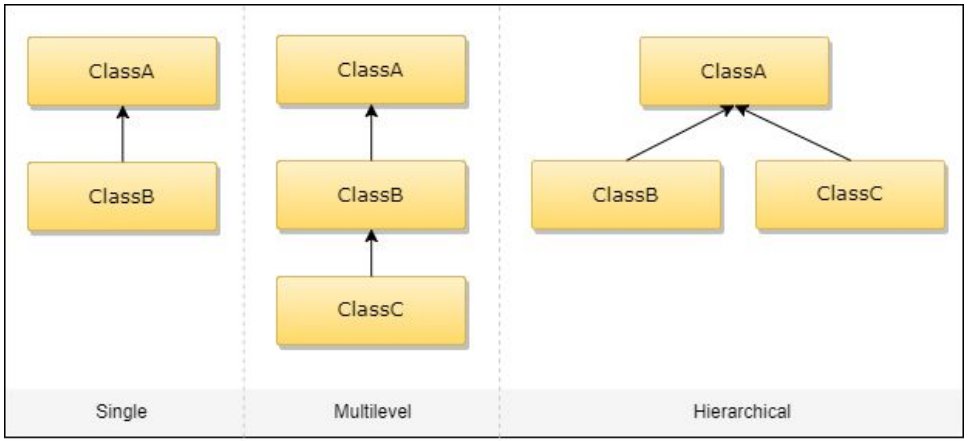
```
Interface IFly{
fly ();
}
```

Differences

- Each method of an interface is an abstract method.
- An interface is implemented by another class. It is never extended by a class.
- An interface, however, may extend multiple interfaces.

Inheritance

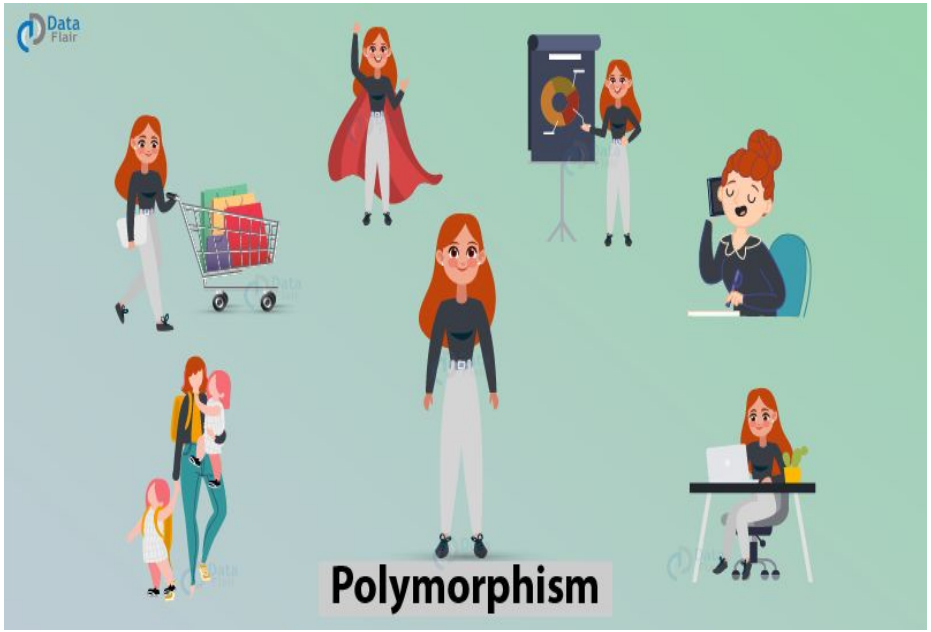




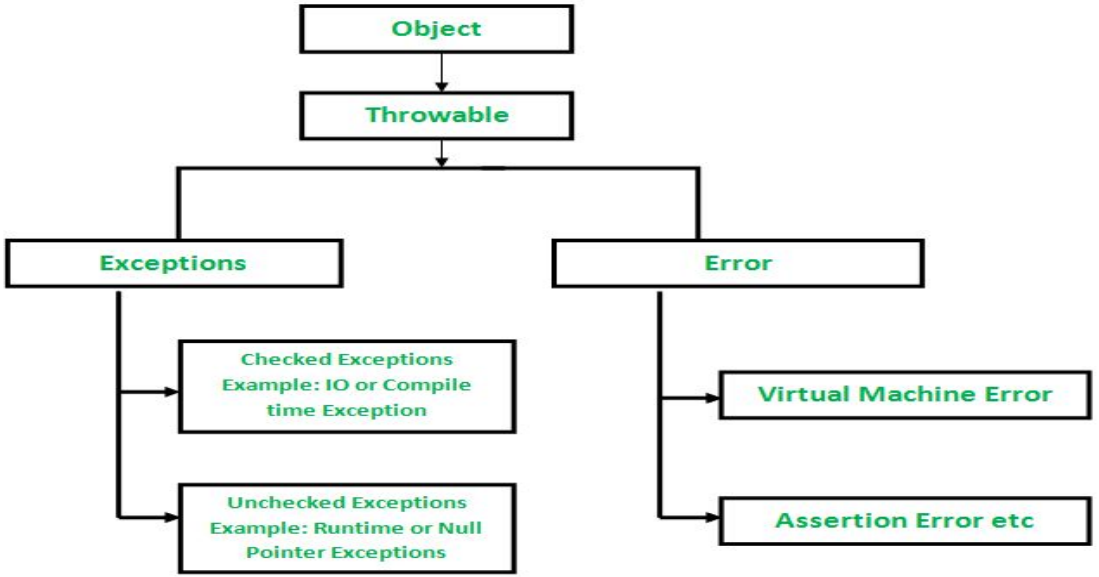
Java Inheritance

Polymorphism

- The word Polymorphism can be broken into two words – ‘**poly**’ means **‘many’** and **‘morph’** means ‘forms’. So, polymorphism means many forms.
- Polymorphism is the ability for a data or message to be processed in more than one form. It is a concept by which a single operation can be performed in multiple different ways.



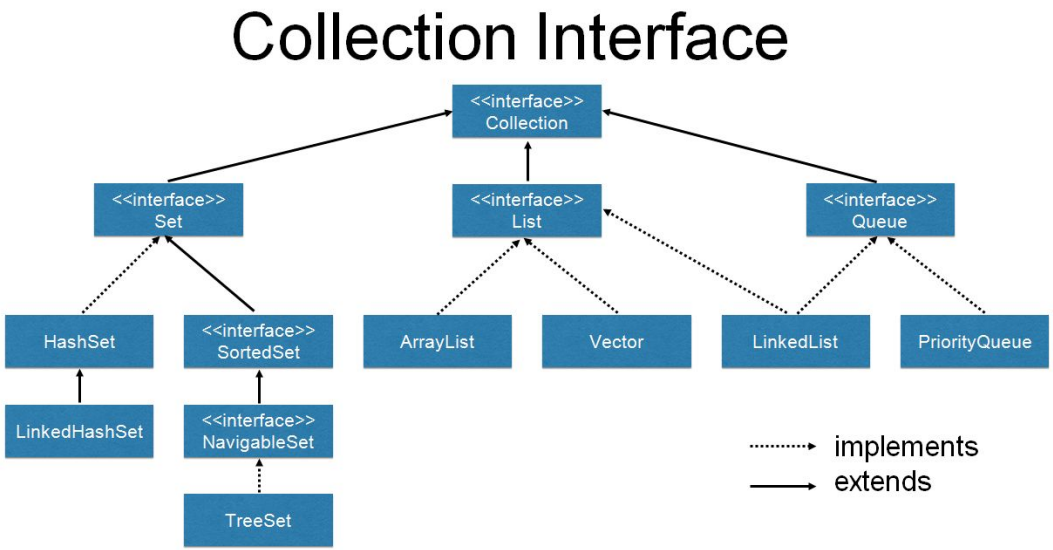
Exception Handling in Java



Wrapper class

- A Wrapper class is a class whose object wraps or contains primitive data types
- They convert primitive data types into objects. Objects are needed if we wish to modify the arguments passed into a method (because primitive types are passed by value).
- The classes in java.util package handles only objects and hence wrapper classes help in this case also.
- Data structures in the Collection framework, such as ArrayList and Vector, store only objects (reference types) and not primitive types.
- An object is needed to support synchronization in multithreading.
- Wrapper class is final. Object of the wrapper class is immutable.

Java Collection



Streams and file handling in java