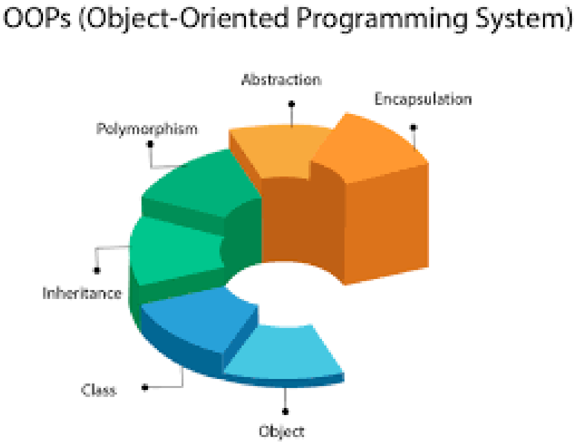
OOPs Concepts in Java

What is OOPS?

Object-Oriented Programming or OOPs are the programming languages that use objects. OOP binds data and related methods together. This helps to increase the maintainability and flexibility of the code.



The following are general OOPs concepts in Java

1. Object
2. Class
3. Inheritance
4. Polymorphism
5. Abstractions
6. Encapsulation

Object

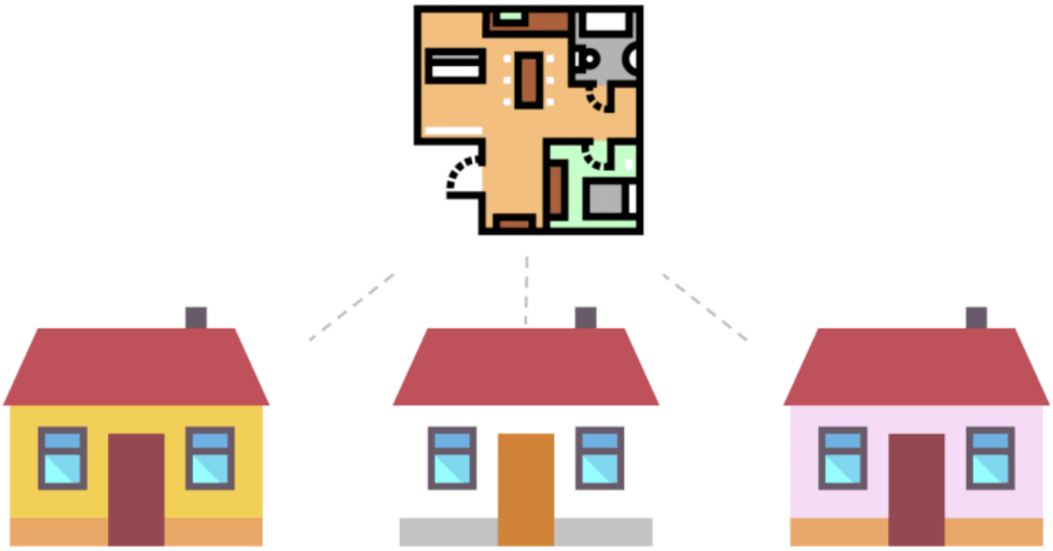
Object is an **instance** or **member** of the class.

Example: A dog is an object because it has states like color, name, breed, etc. as well as behaviors like wagging the tail, barking, eating, etc.

Class

A class is **a blueprint** or a **prototype** that is used to create objects. It represents properties and methods that are common to all objects of a specific type.

In a more simple way; a **class** is like a **blueprint** of a house and **objects** are the **houses** made from it



Inheritance

When one object **acquires the properties and behaviors** of a **parent object**, it is known as inheritance.

Inheritance provides code reusability. Therefore, a ***child class*** can ***extend*** the common behavior of its ***parent class****.*

parent/super class : The class whose features are inherited are known as the super class

child/sub class : The class that inherits behaviors of another class is known as the sub class

Polymorphism

The process of performing a single task in different ways is known as **Polymorphism.**

"Now Speak!"
"Quack"
"Woof"
"Meow"

Polymorphism in OOP is the concept that you can access objects of different types through the same interface. Each type can provide its own independent implementation of this interface.

As an example let's look at the animal class and its subclasses; dog, cat & duck. The animal class has a speak method. But the implementation of the speak method is different from each subclass.

Polymorphism in java is achieved by using;

a) Method Overloading: Keeping the same method signature but changing the method implementation

b) Method Overriding: Keeping the method name and return type same but changing the parameters

Abstraction

Abstraction is the process of **showing only the related** data and **hiding unnecessary details** from the user.

A real-life example of abstraction is a coffee machine.

Encapsulation

Binding (or wrapping) code and data together into a single unit are known as encapsulation

In encapsulation, the ***data of a class is hidden from other classes*** and can be ***accessed only through any member function*** of the class in which they are declared.

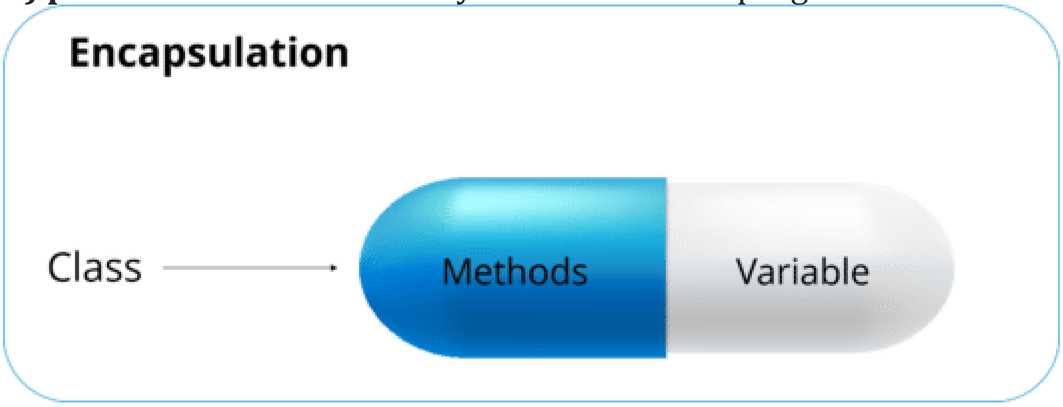
This process is done by using **access modifiers.** There are 4 access modifiers in Java.

a) private : Can be accessed only within the class

b) default: Can be accessed only within the package

c) protected: Accessible within the package and outside the package, only through child classes.

d) public: Can be accessed anywhere inside the program



Summary

OOP concepts in Java help you to structure your program more efficiently. The seven object-oriented principles we've explored here (abstraction, encapsulation, polymorphism, inheritance, association, aggregation, and composition) can help you reuse your code, prevent security issues, and improve the performance of your Java applications.

More from E.M.L.P Bandara
Blog Link : https://medium.com/@lahirupr471/oops-concepts-in-java-38c70a9e78a1