Java OOP terminology

Class -> Abstraction (KIND,CATEGORY in English) is a structure that bundles functions(methods) and variables(fields/attributes) together.

Class can be

Objects: tangible - concreted

members of class:

|  |  |
| --- | --- |
| Variables/fields  Int a;  Float b; |  |
| Functions/met.  getName()  getNumber() |  |

Objects: tangible - concrete

instantiating of it (Object)

This instances obey inheritances.

Static class cannot be instantiated

Chair – **Class** define some properties – color, number of legs,

The chair – **Object**

**Static variables**: belongs to the class, each object doesn’t have a copy of that, doesn’t belongs to the instance.

**Not Static variable**: Any object has one copy. Created every instance.

**Static Methods ->** Belongs to the class.Can be called without create any instance, object

**Not static Methods ->** Stick to the object, doesn’t exist if class isn’t instantiated.

Object oriented programming:

Variables store data.

Variables: Inside classes

* Fields
* Properties:

Function: programs that (may) return values.

* Methods inside a class

**Static void main: main door of your program. We don't want to instantiate an object to call their methods.**

1. **Inheritance:** Can create sub-classes, **Hierarchy Chair** -> office chair > baby chair > easy chair > stool, etc… Sub classes inherit everything**. (Child class)**

1. **Encapsulation:** Certain members of a class private and other are public, it means accessible from outside.
2. **Polymorphism:** Same methods can have different behaviors depending on the context Different number of arguments. Or types.

Conventions: Class – Capital Letter – **M**yClass.java!