Contents

- Java OOPS Concepts: Object, Class, Inheritance, Polymorphism, Abstraction, Encapsulation
- Exception Handling in Java
- Java Collection

Classes and Objects in Java

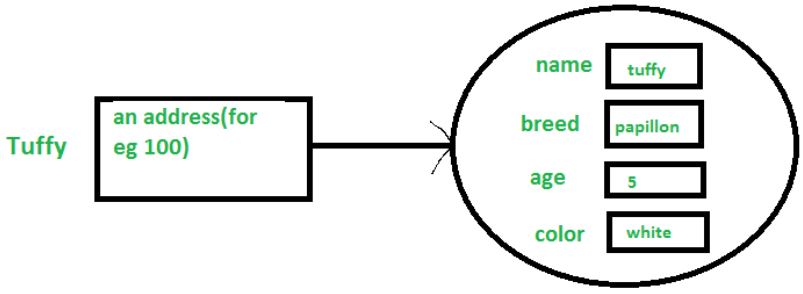
A **CLASS** is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. In general, class declarations can include these components, in order:

1. **Modifiers**: A class can be public or has default access
2. **class keyword:** class keyword is used to create a class.
3. **Class name:** The name should begin with an initial letter (capitalized by convention).
4. **Superclass(if any):** The name of the class's parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent.
5. **Interfaces(if any):** A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.
6. **Body:** The class body surrounded by braces, { }.

CONSTRUCTORS : Constructors are used for initializing new objects. Fields are variables that provides the state of the class and its objects, and methods are used to implement the behavior of the class and its objects.

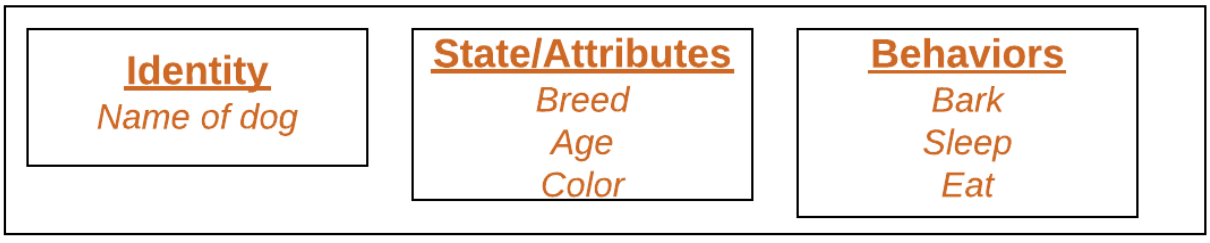
POINTS TO DISCUSS : Class , Abstract Class , Interface

An **OBJECT** is a basic unit of Object-Oriented Programming and represents real life entities. A typical Java program creates many objects, which as you know, interact by invoking methods.

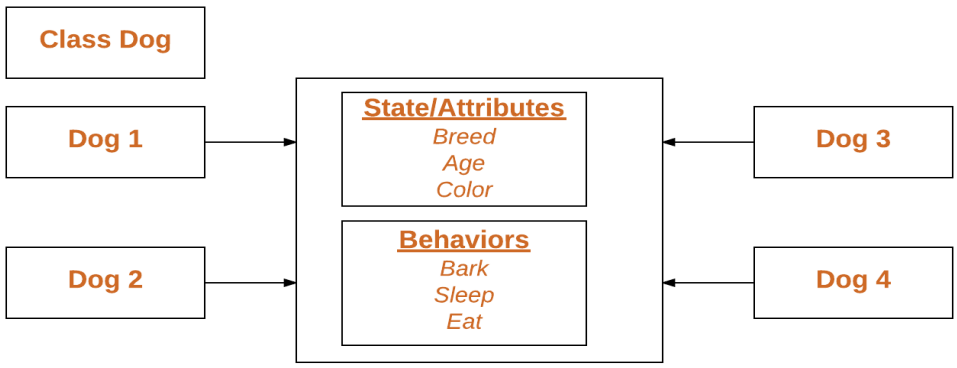


An object consists of :

1. **State**: It is represented by attributes of an object. It also reflects the properties of an object.
2. **Behavior**: It is represented by methods of an object. It also reflects the response of an object with other objects.
3. **Identity**: It gives a unique name to an object and enables one object to interact with other objects.



Declaring Objects (Also called instantiating a class)



Code Example for Practice : https://ide.geeksforgeeks.org/yK1v1b9KLa

Ways to create object of a class

1. Using new keyword
2. Using Class.forName(String className) method
3. Using clone() method
4. Deserialization

Topics to discuss :

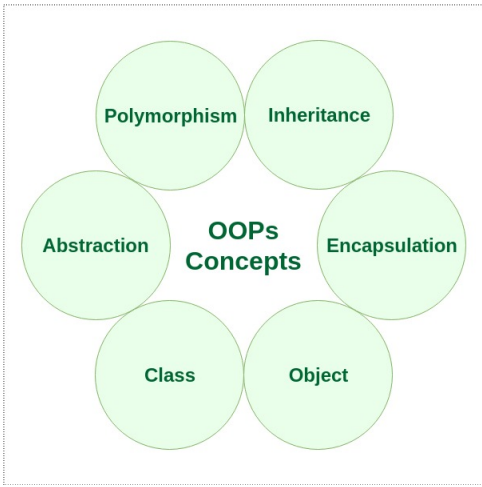
1. Static keyword in Java (https://www.geeksforgeeks.org/static-keyword-java/)

2. Final keyword in Java (https://www.geeksforgeeks.org/final-keyword-in-java/)

Access Modifier : Defines **access type** of the method i.e. from where it can be accessed in your application. In Java, there 4 type of the access specifiers.

- **public:** accessible in all class in your application.
- **protected:** accessible within the package in which it is defined and in its **subclass(es)(including subclasses declared outside the package)**
- **private:** accessible only within the class in which it is defined.
- **default (declared/defined without using any modifier):** accessible within same class and package within which its class is defined.
-

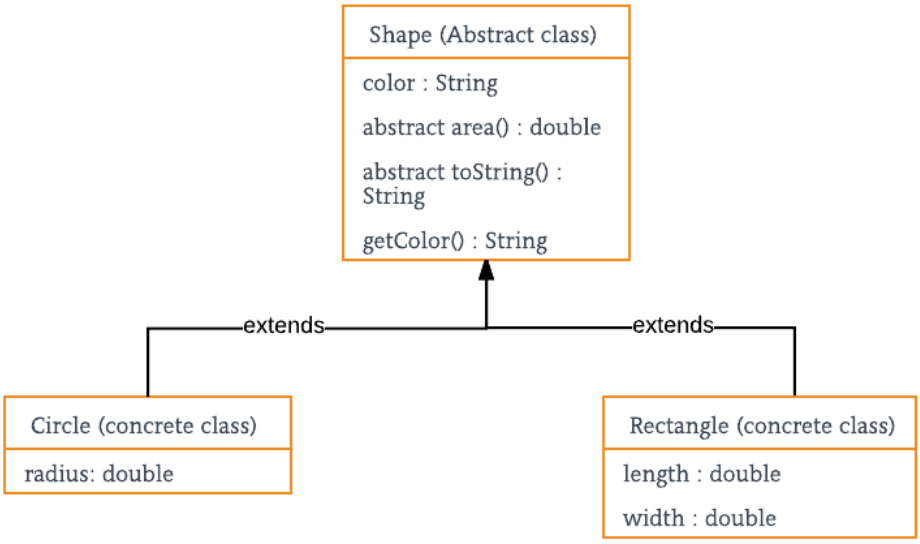
Object Oriented Programming (OOPs) Concept in Java



Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism etc in programming. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

Let us now discuss 4 pillars of OOPS:

Pillar 1: Abstraction



Data Abstraction is the property by virtue of which only the essential details are displayed to the user. The trivial or the non-essentials units are not displayed to the user. Ex: A car is viewed as a car rather than its individual components.

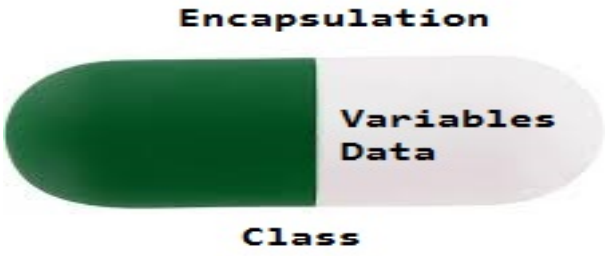
Code Example : https://ide.geeksforgeeks.org/btCYEwU3Mj

Point to discuss : **Abstract classes and Abstract methods**

Pillar 2: Encapsulation

Encapsulation is defined as the wrapping up of data under a single unit. It is the mechanism that binds together code and the data it manipulates. Another way to think about encapsulation is, it is a protective shield that prevents the data from being accessed by the code outside this shield.

Code Example : https://ide.geeksforgeeks.org/wh3Iqv4eHx



Pillar 3: Inheritence

Inheritance is an important pillar of OOP(Object-Oriented Programming). It is the mechanism in java by which one class is allowed to inherit the features(fields and methods) of another class.

Types of Inheritance in Java

1. Single Inheritance : https://ide.geeksforgeeks.org/iXpFHkh4rz
2. Multilevel Inheritance : https://ide.geeksforgeeks.org/tbBVCp8wQn
3. Hierarchical Inheritance : https://ide.geeksforgeeks.org/EXeCyyqZgY
4. Multiple Inheritance : https://ide.geeksforgeeks.org/W161yoraTO
5. Hybrid Inheritance(Through Interfaces)

Pillar 4: Polymorphism

The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form.

Types of polymorphism :

In Java polymorphism is mainly divided into two types:

- Compile-time Polymorphism : https://ide.geeksforgeeks.org/cijtDcYUT2
- Runtime Polymorphism : https://ide.geeksforgeeks.org/4cgdWD1eJX

Exceptions in Java

Exception Handling in Java is one of the effective means to handle the runtime errors so that the regular flow of the application can be preserved. Java Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IOException, SQLException, RemoteException, etc.

What is an Exception?

An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e at run time, that disrupts the normal flow of the program's instructions.

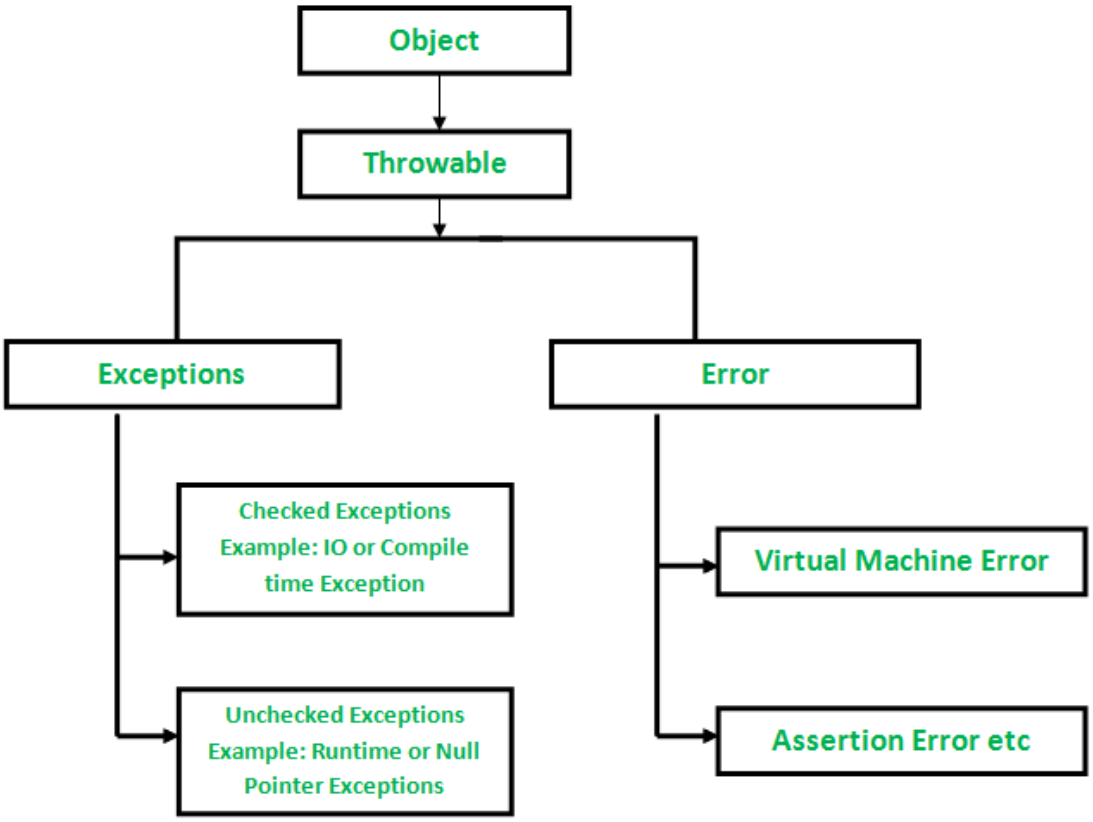
What is an Error?

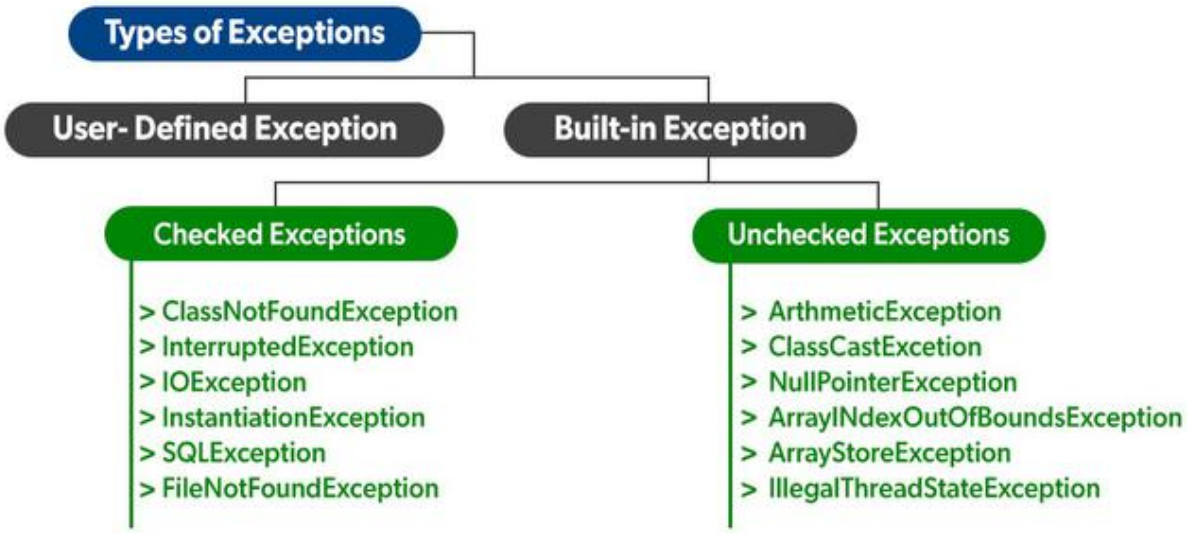
Errors represent irrecoverable conditions such as Java virtual machine (JVM) running out of memory, memory leaks, stack overflow errors, library incompatibility, infinite recursion, etc.

Errors are usually beyond the control of the programmer and we should not try to handle errors.

Error vs Exception

- **Error:** An Error indicates a serious problem that a reasonable application should not try to catch.
- **Exception:** Exception indicates conditions that a reasonable application might try to catch.





Exceptions can be Categorized into 2 Ways:

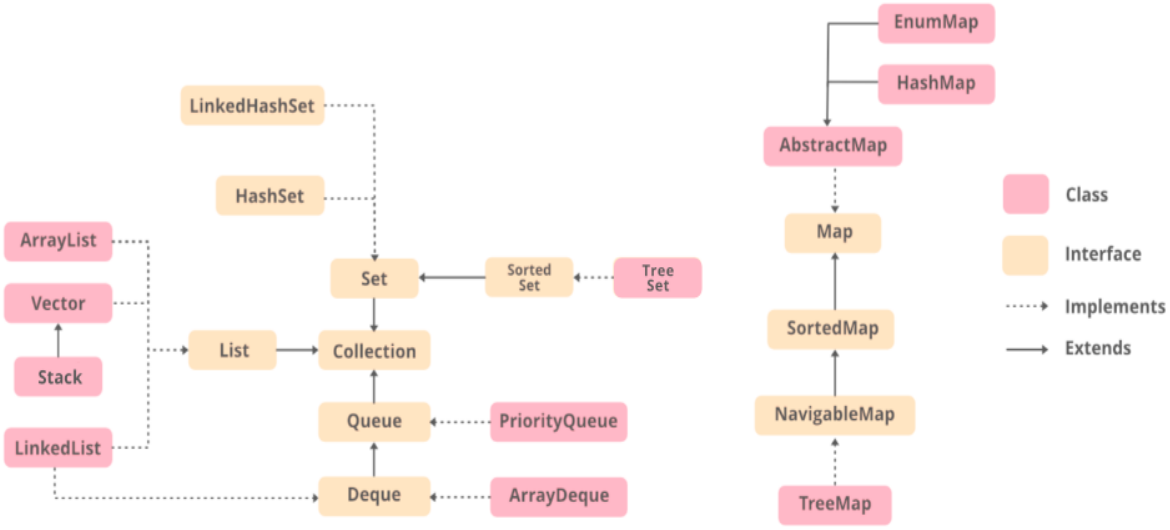
1. Built-in Exceptions
    - Checked Exception
    - Unchecked Exception

2. User-Defined Exceptions

Points to discuss :

- Need of try-catch clause(Customized Exception Handling)
  https://ide.geeksforgeeks.org/wWRSZ6gZVj
- throw, throws https://www.geeksforgeeks.org/throw-throws-java/
- finally https://www.geeksforgeeks.org/final-keyword-in-java

Collections in Java



POINTS TO DISCUSS :

1. Iterable Interface
2. Collection Interface
3. List Interface
4. ArrayList (https://ide.geeksforgeeks.org/VAojWFIbfu)
5. LinkedList (https://ide.geeksforgeeks.org/v4HVcUORsc)
6. Set Interface
7. HashSet (https://ide.geeksforgeeks.org/KgfnJhwz8j)
8. Map Interface
9. HashMap (https://ide.geeksforgeeks.org/L3WyK25UI5)

Tip : To see all functions inside your current class , press Ctrl + F12 in Windows or Cmd + Fn + F12 in Mac .
https://betterprogramming.pub/intellij-keyboard-shortcuts-to-swear-by-7638c0efcc76

Tip : To kill processes running on specific port :
lsof -i tcp:<port>
[~]$ kill -9 <pid>

https://stackoverflow.com/questions/5813614/what-is-difference-between-errors-and-exceptions