Classes and objects

- In Java, all code consists of classes.
- Every object is an instance of some class.
- A source program can have multiple classes in it.
- They follow object - oriented concepts such as abstraction, encapsulation, inheritance, polymorphism.
- For inheritance, ‘extends’ keyword is used n Java.
- Objects are created using ‘new’ operator in Java.

```
class_name   object_name  =  new   constructor_name();
```

Sample program showing object creation in Java

```
public class Main {
  int x = 64;

  public static void main(String ar[]) {
    Main myObj = new Main();
    System.out.println(myObj.x);
  }
}
```

Output

64

Arrays, Constructors, Strings in Java

- Arrays are collection of similar type.
- It is possible to create arrays with or without objects.
- Constructors are called automatically when an object is created.
- If the class name is same and the number and type of parameters are different, then it is referred to as constructor overloading.
- Strings are considered as objects of type String.
- A string object cannot be modified. In case of modification, a new string object is created.
- String functions present in C, C++ are also present in Java. In addition to it, some new functions are also introduced.

Access control specifiers

- private - can be accessed only within the class.
- default - can be accessed by any other class in the same package.
- protected - can be accessed by any other class in the same package or inherited classes from other package.
- public - can be accessed by any class inside or outside the package.

Static variables, methods, block

Static variables

- can be used to refer the common property of all objects
- gets memory only once at the time of class loading

Static methods

- a method declared static can be called without creating an object
- can directly access other static properties in the same class

Static block

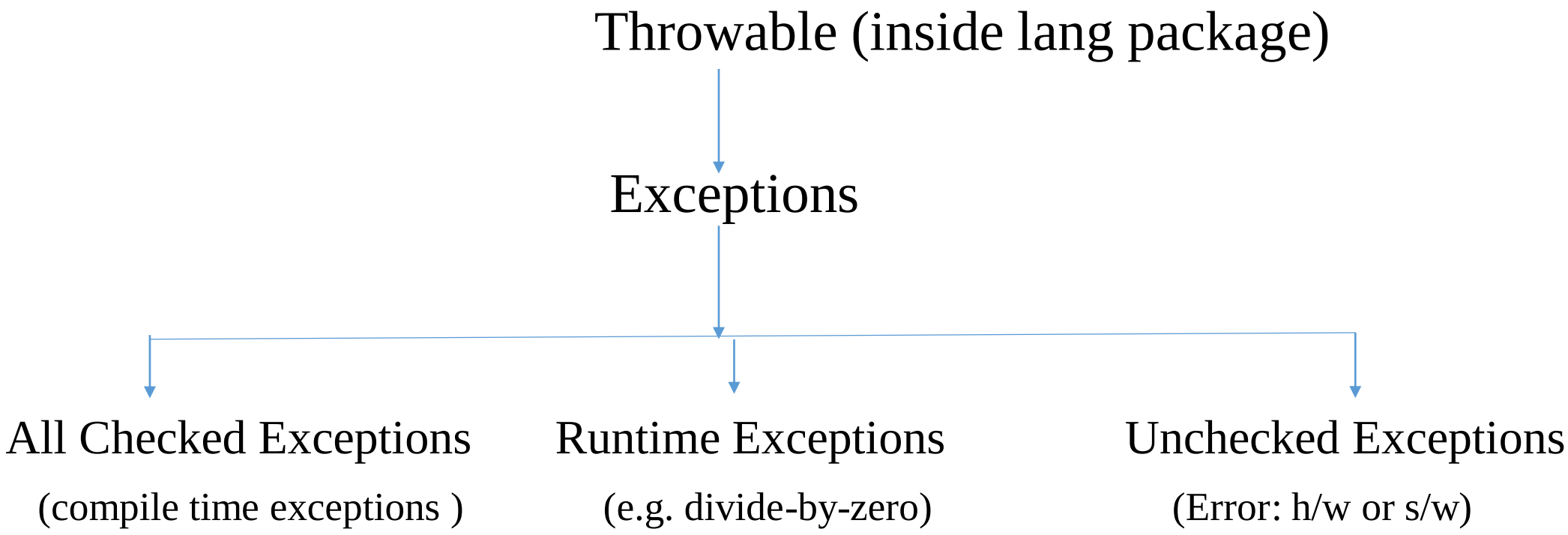
- it is the only block that can be invoked before ‘main()’ method.

Interfaces

- An abstract class is a class which is not used for creating objects but is a basis for making subclasses.
- Interface is similar to abstract classes, but all methods are abstract and all properties are static final.
- It specifies what a class must do but not how.
- Methods do not have any implementation part.
- Interfaces can be implemented by classes and extended by other interfaces.
- All methods in an interface are public and abstract.
- All variables in an interface are public, static and final.

Exceptions in Java

- Event that occurs during the execution of a program that disrupts the normal flow of instructions is said to be an exception.
- When an error occurs, an exception object is created and handed to the runtime system. This procedure is termed as throwing the exception.
- In Java, the following keywords are used in exception handling
    - try, catch, throw, finally, throws


Throwable (inside lang package)
Exceptions
All Checked Exceptions
(compile time exceptions)
Runtime Exceptions
(e.g. divide-by-zero)
Unchecked Exceptions
(Error: h/w or s/w)

- Programs to be monitored for exceptions are contained within the try block.
- The exception is thrown using the ‘throw’ keyword.
- The corresponding catch block handles the exception.
- The try statement can have multiple catch statements each with a different type of exception.
- ‘finally’- creates a bloc of code that s executed after a try/catch block has completed and before the code following the try/catch block. It is executed whether or not n exception is thrown.
- ‘throws’- lists the types of exceptions that a method might throw.

Multithreading in Java

- The ability of an operating system to execute the different parts of a program called threads at the same time is referred to as multithreading.
- A single process may have multiple threads.
- A thread is a light-weight process.
- All threads share the same memory space, and variables and can communicate with each other directly.
- Primary methods used are:

  start(), run()

Lifecycle of a thread

- New state - create an instance of Thread class.
- Ready state - ready for execution , waiting for CPU access.
- Running state - executing stage of a thread.
- Waiting state - waiting for some action to happen.
- Dead state - when thread has finished execution.

Creating a thread(2 ways)

- Create a thread by extending Thread class
- Create a thread by implementing RunnableInterface

Packages

- It is a container for classes.
- Contains group of related types.
- Mainly used to avoid naming conflicts and to control access.
- If package is not mentioned in a program, then default package will be automatically loaded on to the program.
- Packages in Java begin with java or javax and to avoid conflicts with classes they are all lowercase.
- To have a hierarchy of packages, each package name is separated with a dot(.)

1.java.lang package

- All classes of this package are imported automatically into all programs.
- Contains classes and interfaces that are fundamental to all programs.

Object class

- Super class of all classes.
- All classes inherit the methods of Object class.

Wrapper class

- Used to wrap primitive types in a class structure.
- All primitive types have an equivalent class.
- Includes useful constants and static methods, including one to convert back to the primitive type.

Primitive data type	Wrapper class
double	Double
float	Float
long	Long
int	Integer
short	Short
byte	Byte
char	Character
boolean	Boolean

Number class

- Abstract class
- Superclass of Integer, Long, Float, Double, Short, Byte

System class

- Class for JVM and the control and security for Operating System.
- Define the standard input and output.
- Contains input stream class, output stream class and error stream class.

Math class

- Contains all floating point functions used in geometry and trigonometry and general purpose methods.

2. java.util package

- A package that supports a wide range of functionalities.
- It includes

  collections - a group of objects

  collection framework- represents a unified architecture for storing and manipulating group of objects

- Collection framework contains interfaces (List Interface, Set Interface, Queue Interface, Map Interface)