Java™

ORACLE

KIRANJIT
MTECH
ANDROID & IPHONE APPS

Point of Discussion

- Packages
- Abstract classes
- Interface

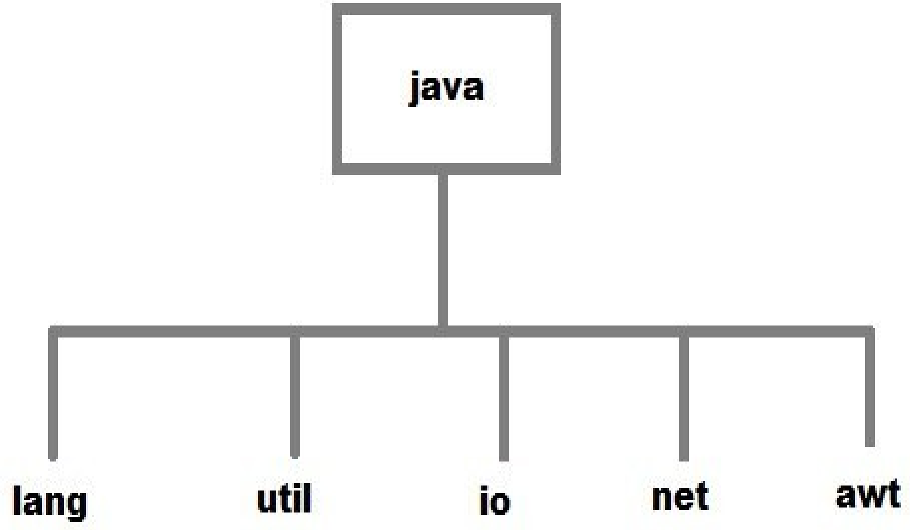
Packages

Package can be defined as agroup of similar types of classes,interface, enumeration and sub-packages.Using packages it is easier to locate the related clases.

- Two type:-

1.Built-in:-java.lang , java.util etc

2.User-defined



abstract clas

- If a class contain any abstract method then the class is declared as abstract class.An abstract class is never instantiated.It is used to provide abstraction.Although it does not provide 100% abstractionbecause it can also have concrete method.

Why abstract class?

Abstraction is an important feature in OOPs. It means hiding complexity.Abstract class is used to provide abstraction.Athough it does not provide 100% abstraction because it can have concrete method.

```
abstract class Vehicle{
public abstract void engine();
}
Public class Car extends Vehicle{
Public void engine()
{
System.out.println(“Car engine”);
}
Public static void main(String args[]){
Vehicle v=new Car();
v.engine();
}
}
```

Points to ponder:

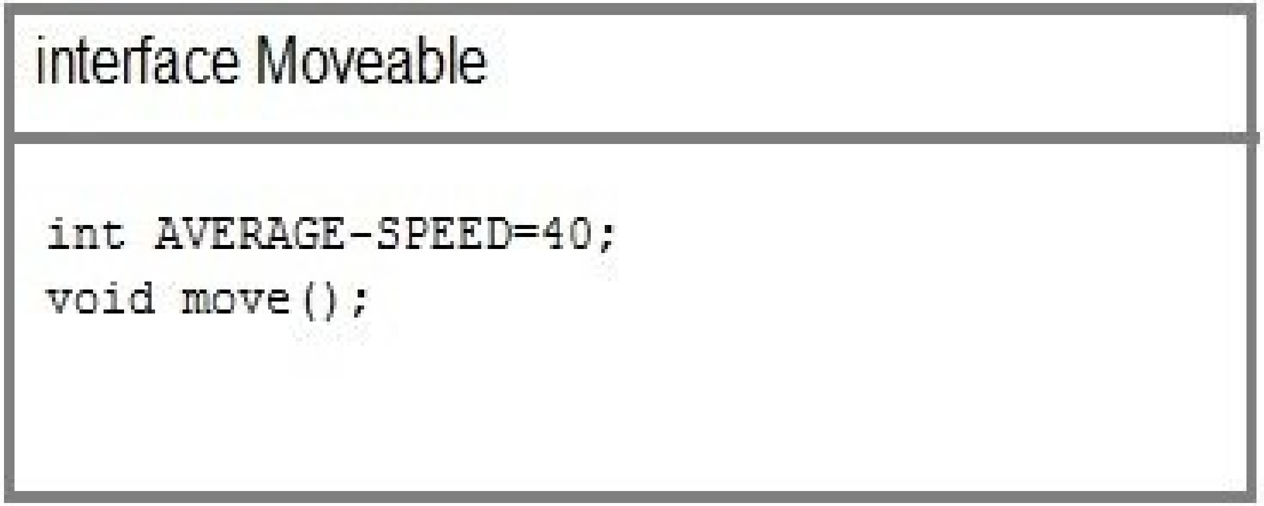
- Abstract classes are not interface. They are different, we will study this when we will study Interface.
- An abstract class must have an abstract method.
- An abstract classes can have constructors,Member variables and normal methods.
- When we extend Abstract class with abstract method,we must define the abstract method in the child class or make the child class abstract.

Interface.........

Interface is a pure abstract class.They are syntactically similar to classes, but you cannot create instance of an **Interface** and their methods are declared without any body. Interface is used to achieve complete **abstraction** in Java. When you create an interface it defines what a class can do without saying anything about how the class will do it.

example

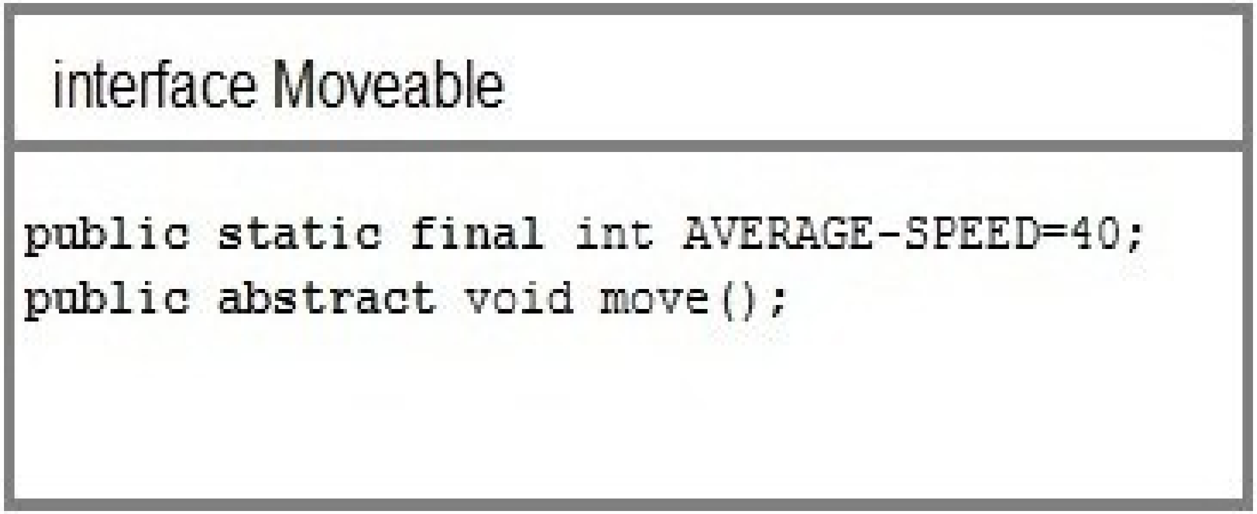
```
interface Moveable
{
 int AVERAGE-SPEED=40;
 void move();
}
```



interface Moveable

```
int AVERAGE-SPEED=40;
void move();
```

what you declare



interface Moveable

```
public static final int AVERAGE-SPEED=40;
public abstract void move();
```

what the compiler sees

• Rules for using Interface

1. Methods inside Interface must not be static, final, native or strictfp.
2. All variables declared inside interface are implicitly public static final variables(constants).
3. All methods declared inside Java Interfaces are implicitly public and abstract, even if you don't use public or abstract keyword.
4. Interface can extend one or more other interface.
5. Interface cannot implement a class.
6. Interface can be nested inside another interface.

```
interface Moveable
{
 int AVG-SPEED = 40;
 void move();
}

class Vehicle implements Moveable
{
 public void move()
 {
  System .out. print in ("Average speed is"+AVG-SPEED");
 }
 public static void main (String[] arg)
 {
  Vehicle vc = new Vehicle();
  vc.move();
 }
}
```

Multiple inheritance

```
interface Moveable
{
 boolean isMoveable();
}

interface Rollable
{
 boolean isRollable
}

class Tyre implements Moveable, Rollable
{
 int width;

 boolean isMoveable()
 {
  return true;
 }

 boolean isRollable()
 {
  return true;
 }
 public static void main(String args[])
 {
  Tyre tr=new Tyre();
  System.out.println(tr.isMoveable());
  System.out.println(tr.isRollable());
 }
}
```

Output:

```
true
true
```

• Interface extends other Interface

Classes implements interfaces, but an interface extends other interface.

```
interface NewsPaper
{
 news();
}

interface Magazine extends NewsPaper
{
 colorful();
}
```

Abstract class	Interface
Abstract class is a class which contain one or more abstract methods, which has to be implemented by its sub classes.	Interface is a Java Object containing method declaration but no implementation. The classes which implement the Interfaces must provide the method definition for all the methods.
Abstract class is a Class prefix with an abstract keyword followed by Class definition.	Interface is a pure abstract class which starts with interface keyword.
Abstract class can also contain concrete methods.	Whereas, Interface contains all abstract methods and final variable declarations.
Abstract classes are useful in a situation that Some general methods should be implemented and specialization behavior should be implemented by child classes.	Interfaces are useful in a situation that all properties should be implemented.