

15.04.2022

## Java Day 5 - Assignment

### Scanner class :-

Scanner class is used to get the input from the user at run time.

It is present in package `java.util` (utility)

### Scanner Methods

Method	Description
<code>nextBoolean()</code>	- Read a Boolean value from user.
<code>nextByte()</code>	- Reads a byte value from the user.
<code>nextShort()</code>	- Reads a short value from the user.
<code>nextInt()</code>	- Reads a int value from user.
<code>nextLong()</code>	- Reads a long value from user.

~~nextFloat~~ nextFloat() - Reads a float value from the user

nextDouble() - Reads a double value from the user

nextLine() - Reads a String Value from the user.

next() - Reads a String Value from the user.

### SYNTAX:

```
Scanner myScanner = new Scanner  
                        (System.in);
```

eg.,

```
Scanner myObj = new Scanner(System.in);
```

### Method calling :

```
myName. ScannerMethods();
```

### Ex program:

```
next Int :  
nextLine();  
next();
```

```
Package org.scanner;  
import java.util.Scanner;
```

```
public class ScannerExample {  
    public static void main (String []  
                                args)  
    { Scanner s = new Scanner(System.in);  
      System.out.println ("Enter your name");  
      String data = s.next();  
      System.out.println ("Continue your name:" +  
                          data);  
    }
```



## oops concept:

oops (Object-Oriented Programming system)

Everything considered to be an object.

Object-Oriented programming is a methodology or paradigm to design a program using classes and objects. It simplifies software development and maintenance by providing some concepts:

- object
- class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation.

### OBJECT::

Any entity that has state and behaviour is known as an object.

object is an instance of class.  
super class of Java.

### CLASS::

class is a blueprint (or) template collection of objects is called class.

## Inheritance :

Inheritance is a mechanism in which one class acquires the property of another class. we can access one class property from another class.

when one object acquires all the properties and behaviors of a parent object. It is known as inheritance.

- It provides code reusability.

Memory wastage is low

It is used to achieve runtime

Polymorphism

### Important terminology :

super class / parent class / base class  
=== subclass (child class)  
derived class.

### Superclass :

The class whose features are inherited is known as superclass (or a base class or parent class).

### Subclass :

The class that inherits the other class is known as a subclass or a derived class, extended class, or child class.

The subclasses can add its own fields and methods in addition to the superclass fields and methods.



## Reusability :

Inheritance supports the concept of reusability i.e., when we want to create a new class and there is already a class that includes some of the code that we want we can derive our new class from existing class.

## SYNTAX:-

```
class derived-class extends  
    base-class  
{ // methods and fields  
}
```

## TYPES OF INHERITANCE:

Single Inheritance

Multilevel Inheritance

Hierarchical Inheritance

Multiple Inheritance

Hybrid Inheritance.

## SINGLE INHERITANCE:-

when a class inherits another class, it is known as Single Inheritance.

Ex Program:

```
package org.inheritance;  
public class Father  
{ int tot-car = 10;  
  public void house() {  
    System.out.println("Fathers house  
    method");  
  }  
  public void car() {  
    System.out.println("Fathers  
    car method");  
  }  
}
```

```
Package org.inheritance;  
Public class Son extends Father {  
  public void bike()  
  { System.out.println("Sons bike  
  method");  
  }  
  public static void main (String []  
  args)  
  { Son s = new Son();  
    s.bike();  
    s.car();  
    s.house();  
    System.out.println(s.tot-car);  
  }  
}
```

### Multilevel Inheritance:

when there is a chain of inheritance, It is known as multilevel inheritance.



Ex-program:

```
package org.inheritance;  
public class student  
{  
    public void studentNames()  
    {  
        System.out.println("12 students");  
    }  
    public void smallTask()  
    {  
        System.out.println("small Task");  
    }  
    public void weeklyTask()  
    {  
        System.out.println("weekly Task");  
    }  
}
```

---

Package org.inheritance;

```
public class Trainer extends Student  
{  
    public void taskValidation()  
    {  
        System.out.println  
            ("Task Validation");  
    }  
    public void Evaluation()  
    {  
        System.out.println  
            ("Evaluation");  
    }  
    public static void main  
        (String [] args)  
    {  
        Trainer t = new Trainer();  
        t.studentNames();  
    }  
}
```

```
t. weeklyTask();  
t. taskValidation();
```

```
} }
```

```
package org.inheritance;  
public class Aiite extends Trainer {  
    public void mainEvaluation()  
    {  
        System.out.println("Aiites  
        Main Evaluation");  
    }  
    public static void main (String  
        [] args)  
    {  
        Aiite a = new Aiite();  
        a.mainEvaluation();  
        a.SmallTask();  
        a.Evaluation();  
    }  
}
```

Hierarchical Inheritance:

when two (or) more classes  
inherits a single class, It is known  
as hierarchical inheritance.

Ex Program:

```
package org.inheritance;  
public class InterviewPreparation  
{  
    int tot_stu = 100;  
    public void javaQuestion()  
    {  
        System.out.println
```



C"only Java interview Question";

```
}  
public void seleniumQuestion()  
{ System.out.println("only Java interview  
Questions");  
}
```

```
package org.inheritance;  
public class JavaPreparation  
extends InterviewPreparation  
{ int jstudent = 50;  
public void javaStudent()  
{ System.out.println("Java class Student");  
}
```

```
}  
public static void main  
(String [] args)  
{ JavaPreparation jp = new  
JavaPreparation();  
jp.javaStudent();  
jp.javaQuestion();  
}
```

---

```
package org.inheritance;  
public class SeleniumPreparation  
extends InterviewPreparation  
{ int selstudent = 35;
```

```

public void SeleniumStudent()
{
    System.out.println("Java class student");
}

public static void main (String [] args)
{
    SeleniumPreparation sp = new SeleniumPreparation();
    sp.SeleniumStudent();
    sp.SeleniumQuestion();
    System.out.println(sp.getTotal());
    System.out.println(sp.getStudent());
}
}

```

### MULTIPLE INHERITANCE:

One class can inherit properties of more than one parent class.

To reduce the complexity and simplify the language, multiple inheritance is not supported in java.

### HYBRID.

Hybrid Inheritance in Java is a combination of two or more types of inheritances.



~~07~~ 15-04-2023

Non-Access Modifier:

final → value cannot be changed

Static

Abstract

Scanner - predefined class

+ - concatenation

Scanner s = new Scanner(System.in);

s.nextInt(); → Integer

s.next(); → String

s.nextLine(); → String

Small task:

nextInt(), get runtime input in  
all data types.

app:

Object oriented programming System

Instance : Code reusability

Single Inheritance - class inherit  
another class

## Hierarchical Inheritance

