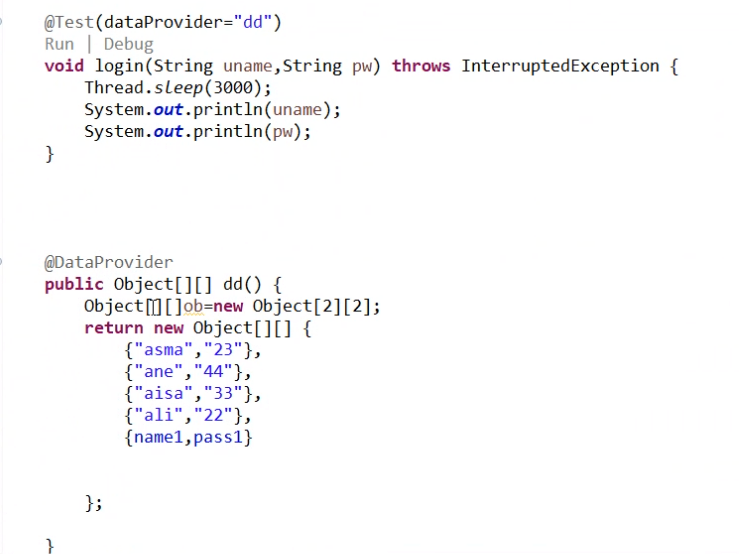
\





=

====================

<suite name=*"suite."* parallel = *"test"*>

<test>

<classes>

<class name = *"packagename.classname"*>

<method>

<exclude> name= "packagename.dd"

</exclude>

<!-- -comment -->>

</method>

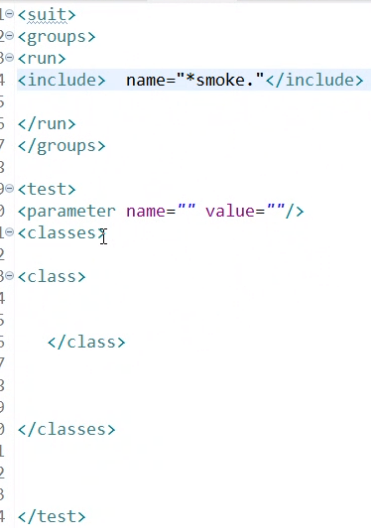
</class>

</classes>

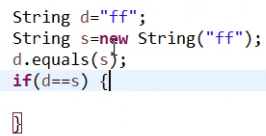
</test>

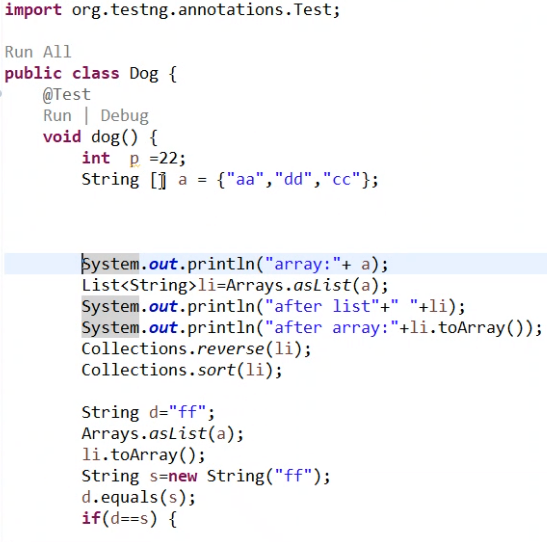
</suite>

<!-- - --!>



Class -package name and Class name

 **32)**



**Underlying data structure of**

**Arraylist – growable array**

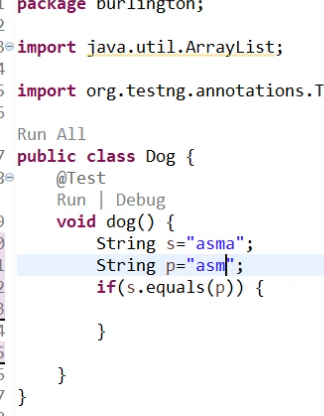
**Linked list – dubulely linked list**

**Hash set – hash map**

**Ve ,HT properties , Enumerationm,Dictionary , Stack**

**Map does not extends collection interface -**

Upcast and downcastinh







**Call by value and Call by reference—**

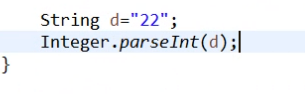
**Call by value- Calling a method with passing parameter as value is known as Call by Value.**

**Call by reference-**

**a method with passing parameter as reference is known as Call By reference.**

**Hash table s subclass – properties class**

**Variable declare – int value – to string we use parseInt() example of Wrappeer class**



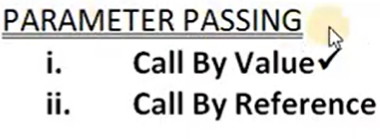
**Main classs- no default / pvt/- but it will be abstract/ final/ public [PAAF]**

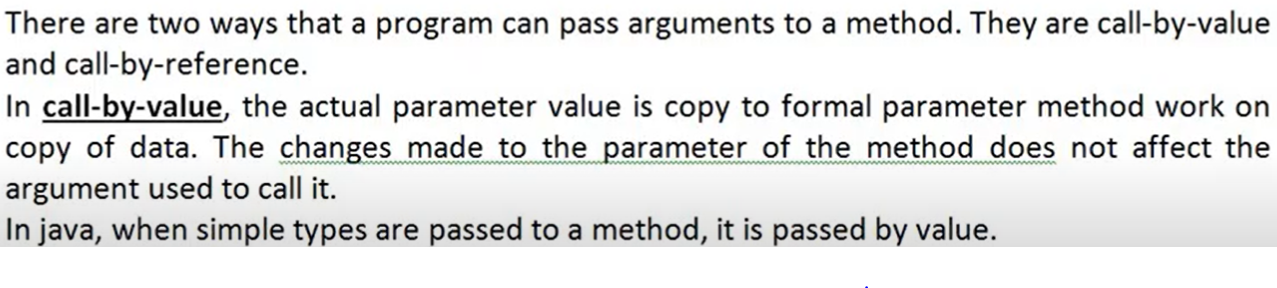
**Can u make customize immutable class -like string is immutable class**

**Class -final – so we can not extends the class .that’s why this class customized class**

**Public final class and variable will be private/ final [nobody can not access it but here we can not use getter and setter method], We can not extends but we can create object of that class. At a time all things can not take by the sub classes. And here we can use**







**package** callbyvalueandreference;

**public** **class** CallByValue {

**public** **static** **void** main(String[] args) {

CallByValue obj = **new** CallByValue();

**int** x =30;

**int** y = 20; // This is Actual parametery

System.***out***.println("X value is : "+ x);

System.***out***.println("Y value is : "+ y);

obj.callValue(x, y); // these are passing to a and b.

System.***out***.println("X value is : "+ x); // this x value will be passed to a .

System.***out***.println("Y value is : "+ y); //// this x value will be passed to b .

}

// First Part ---> Class name callValue

**void** callValue(**int** a, **int** b) { // formal parameter

a = 15; // here a= 15 but actual parameter x = 30 pass to a here.but here to a= 15;

b = 31; // here b= 31 but actual parameter y = 20 pass to b here.but here to b= 31;

// here Actual parameter is passing the value to formal parameter and formal parameter is converted.

// but Actual value will never be changes. here a and b can be changed but x and y will not be changed.

}

//they are receiving from x and y.

// Actual parameters value will be copied to formal/ function parameter.

//So formal parameter has been changes from [a = 15 to 30 ] and [b= 31 to 20].

// Formal parameter will receive the value from the actual parameter's.

// obj.callValue(x, y); -- here call the function/ method and value passing .that's why this is called call by value.

}

/\*

Q- How to achieve call by value and call by ref in java?

Note-

1) Call by reference is possible in java

2)obj.intSum(10, 20); So 10 will be given to a and 20 will be given to b. So the value of a= 10 and value of b= 20.

it means the Copy of 10 and 20 will be given to a and b.

So if you are passing like this int x = 15 and int y = 25 and we are passing like this --

int x = 15;

int y= 25;

obj.intSum(x, y);

So x will be given to a and y will be given to b. but only one copy[photocopy]/duplicate copy

will be given to a and b.

\*/

