Name: Lutika Kolhe

Core Java Assignment 5

**Assignments on Generics**

1. Use Hashset to hold Employee Objects. Upon running the application, the details of the employees added to the Hashset should be displayed.

Employee <<class>>

|-id

|-name

|-salary

|-department

|-displayDetails(optional)

Feel free to add properties and methods to Employee Class.

Note: If we try to store any object other than Employee Object in Hashset, we should not be

Allowed to.

Ans- I created 2 Class here first is Employe.java and another is Hash.java

Class 1= Employe.java

**package** Generic;

**public** **class** Employe

{

**int** Id,salary;

String name,department;

**public** Employe(**int** Id,**int** salary,String name,String department)

{

**this**.Id=Id;

**this**.salary=salary;

**this**.name=name;

**this**.department=department;

}

**public** **int** getId()

{

**return** Id;

}

**public** **void** setId(**int** Id)

{

**this**.Id = Id;

}

**public** **int** getSalary()

{

**return** salary;

}

**public** **void** setSalary(**int** salary) {

**this**.salary = salary;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getDepartment() {

**return** department;

}

**public** **void** setDepartment(String department) {

**this**.department = department;

}

**public** **void** displayDetails()

{

System.***out***.println("Id :"+getId()+" \nSalary: "+getSalary()+"\nName: "

+getName()+"\nDepartment: "+getDepartment());

}

}

Class 1= Hash.java

**package** Generic;

**import** java.util.\*;

**public** **class** Hash {

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

{

HashSet<Employe> set=**new** HashSet<>();

Employe emp=**new** Employe (3 ,80000 , "Lutika kolhe", "Java Developer");

set.add(emp);

emp.displayDetails();

}

}

OUTPUT:

Id :3

Salary: 80000

Name: Lutika kolhe

Department: Java Developer

1. Write an application to hold 10 random int values are keys and 10 random double values as values for a Hashmap. Print the data store in the Hashmap. Note : Keys can only be int and values double.

**package** Generic;

**import** java.util.HashMap;

**public** **class** HoldNum

{

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

{

HashMap<Integer,Double> map=**new** HashMap<>();

map.put(41, 37.9);

map.put(51, 28.7);

map.put(62, 65.7);

map.put(30, 5.3);

map.put(10, 33.8);

map.put(70, 5.9);

map.put(68, 96.8);

map.put(66, 24.5);

map.put(48, 85.5);

map.put(95, 87.6);

System.***out***.println(" Size of the values is :" + map.size());

System.***out***.println(map);

}

}

**OUTPUT:**

Size of the values is :10

{48=85.5, 66=24.5, 51=28.7, 68=96.8, 70=5.9, 41=37.9, 10=33.8, 62=65.7, 30=5.3, 95=87.6}

1. Write a generic method to exchange the postions of two different elements in an array.

**package** Generic;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.Collections;

**import** java.util.List;

**public** **class** Exchange {

**public** **static** **final** <T> **void** swap (T[] a, **int** i, **int** j) {

T t = a[i];

a[i] = a[j];

a[j] = t;

}

**public** **static** **final** <T> **void** swap (List<T> b, **int** i, **int** j) {

Collections.<T>*swap*(b, i, j);

}

**private** **static** **void** swap1() {

Integer [] a = {10, 20};

*swap*(a, 0, 1);

System.***out***.println("a:"+Arrays.*toString*(a));

List<Integer> b = **new** ArrayList<Integer>(Arrays.*asList*(a));

*swap*(b, 0, 1);

System.***out***.println("b:"+b);

}

**public** **static** **void** main(String...args)

{

*swap1*();

}

}

**OUTPUT:**

a:[10, 20]

b:[20, 10]

1. Design a class named Pair which has two properties. The name of the first property is key and that of the second property is value. When desiging the class take case of the following scenarios:
2. Create an Object of pair class to store String value for the properties and String value for the property value. Restriction Apart from String type no other types should be acceptable as Key or value input.

e.g.

myObj.setKey(“1”);

myObj.setValue(“Hello”);

**package** Generic;

**import** java.util.HashMap;

**public** **class** Pair

{

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

{

HashMap<String,String> map=**new** HashMap<>();

map.put("Hello Lutika kolhe", "1");

map.put("How is your day", "It's Fine");

System.***out***.println(map);

}

}

OUTPUT:

{Hello Lutika kolhe=1, How is your day=It's Fine}

1. Create an object of the class pair to store String value for the property key and java.util.Date as value for the property value

myObj.setKey(“Today is”);

myObj.setValues(new java.util.Date());

Note: In scenario a, no data apart from String should be used for key value, in scenario b, no data apart from String for key and java.util.Date should be allowed.

**package** Generic;

**import** java.util.HashMap;

**import** java.util.\*;

**public** **class** Pair

{

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

{

HashMap<String,Date> map=**new** HashMap<>();

map.put("Today is ", **new** java.util.Date());

System.***out***.println(map);

}

}

**OUTPUT:**

{Today is =Mon Jan 17 14:51:45 IST 2022}