1] Implement the program using map

import java.util.HashMap;

public class map{

static void findelement(HashMap<String,String>Bowl,String fruit){

if(Bowl.containsKey(fruit))

System.out.println("The "+fruit+" is "+Bowl.get(fruit));

else

System.out.println("not found");

}

public static void main(String arg[]){

HashMap<String,String> Bowl=new HashMap<String,String>();

Bowl.put("Apple","Red");

Bowl.put("Berry","Blue");

Bowl.put("Orange","Orange");

Bowl.put("Banana","Yellow");

System.out.println(Bowl);

String color=Bowl.get("Banana");

System.out.println(color);

findelement(Bowl,"Banana");

}

}

Output:

{Apple=Red, Berry=Blue, Orange=Orange, Banana=Yellow}

Yellow

The Banana is Yellow

=== Code Execution Successful ===

2] Create a program using linked list for stack

import java.util.LinkedList;

public class Stack{

public static void main(String args[]){

LinkedList<String> letter=new LinkedList<String>();

letter.push("a");

letter.push("b");

letter.push("c");

letter.push("d");

System.out.println("Linked list : "+letter);

System.out.println("Stack Size: " +letter.size());

if(!letter.isEmpty()){

System.out.println("poped element: "+letter.pop());

}

}

}

Output:

Linked list : [d, c, b, a]

Stack Size: 4

poped element: d

=== Code Execution Successful ===

3] Create a program using linked list for queue

import java.util.LinkedList;

public class Queue{

public static void main(String args[]){

LinkedList<String> letter=new LinkedList<String>();

letter.add("a");

letter.add("b");

letter.add("c");

letter.add("d");

System.out.println("Linked list : "+letter);

System.out.println("Queue Size: "+letter.size());

if(!letter.isEmpty()){

System.out.println(letter.removeFirst());

}

System.out.println("Linked list : "+letter);

}

}

Output:

Linked list : [a, b, c, d]

Queue Size: 4

a

Linked list : [b, c, d]=== Code Execution Successful ===

4] Create mobile class with field price and so on take any one field and sort the class

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

class Mobile implements Comparable<Mobile>{

private String brand;

private String model;

private double price;

public Mobile(String brand, String model, double price){

this.brand = brand;

this.model = model;

this.price = price;

}

public String getBrand(){

return brand;

}

public String getModel(){

return model;

}

public double getPrice(){

return price;

}

public String toString(){

return "brand='" + brand + '\'' +

", model='" + model + '\'' +

", price=" + price;

}

public int compareTo(Mobile other){

return Double.compare(this.price, other.price);

}

}

public class Main{

public static void main(String[] args){

List<Mobile> mobiles = new ArrayList<>();

mobiles.add(new Mobile("Apple", "iPhone 12", 999.99));

mobiles.add(new Mobile("Samsung", "Galaxy S21", 899.99));

mobiles.add(new Mobile("Google", "Pixel 6", 699.99));

mobiles.add(new Mobile("OnePlus", "9 Pro", 799.99));

System.out.println("Unsorted List:");

for (Mobile mobile : mobiles){

System.out.println(mobile);

}

Collections.sort(mobiles);

System.out.println("\nSorted List:");

for (Mobile mobile : mobiles){

System.out.println(mobile);

}

}

}

Output:

Unsorted List:

brand='Apple', model='iPhone 12', price=999.99

brand='Samsung', model='Galaxy S21', price=899.99

brand='Google', model='Pixel 6', price=699.99

brand='OnePlus', model='9 Pro', price=799.99

Sorted List:

brand='Google', model='Pixel 6', price=699.99

brand='OnePlus', model='9 Pro', price=799.99

brand='Samsung', model='Galaxy S21', price=899.99

brand='Apple', model='iPhone 12', price=999.99

=== Code Execution Successful ===