Problem 1 : Accelerate the Car

class Car{

private Integer year;

private String make;

private Double speed;

Car (int year,String make,double speed){

this.year=year;

this.make=make;

this.speed=speed;

}

public int getyear(){

return this.year;

}

public String getmake(){

return this.make;

}

public double getspeed(){

return this.speed;

}

void accelerate(){

speed=speed+1;

System.out.println("The new speed is"+" "+speed);

}

}

class RaceTrack{

public static void main (String [] args){

Car c1 = new Car(2010,"Porsche",25.0);

System.out.println (c1.getyear());

System.out.println (c1.getmake());

System.out.println (c1.getspeed());

c1.accelerate();

}

}

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Problem 2 : Inventory Management

import java.util.\*;

class Item{

Integer item\_id;

String item\_name;

Item(int item\_id,String item\_name ){

this.item\_name=item\_name;

this.item\_id=item\_id;

}

Item(){}

void setitem\_id(int item\_id){

this.item\_id=item\_id;

}

void setitem\_name(String item\_name){

this.item\_name=item\_name;

}

public String toString(){

return this.item\_id+" "+this.item\_name;

}

@Override

public boolean equals(Object o){

if(o instanceof Item){

Item temp= (Item) o;

if(this.item\_id.equals(temp.item\_id) && (this.item\_name.equals(temp.item\_name) )){

return true;

}

}

return false;

}

@Override

public int hashCode(){

//System.out.println("hashCode invoked!!");

int prime = 13;

int val = 1;

val = val\*prime + this.item\_id.hashCode();

val = val\*prime + this.item\_name.hashCode();

return val;

//return 40;

}

}

class namesort implements Comparator<Item>{

public int compare(Item I1,Item I2){

return I1.item\_name.compareTo(I2.item\_name);

}

}

class idsort implements Comparator<Item>{

public int compare(Item I1,Item I2){

return I1.item\_id-(I2.item\_id);

}

}

class InventoryAns{

static Item I=new Item();

static ArrayList<Item> list=new ArrayList<> ();

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

int choice;

Item I1=new Item(1,"A");

Item I2=new Item(3,"R");

Item I3=new Item(2,"Z");

Item I4=new Item(4,"H");

Item I5=new Item(10,"M");

list.add(I1);

list.add(I2);

list.add(I3);

list.add(I4);

list.add(I5);

do{

System.out.println("Enter your choice ==");

System.out.println("1) Add Item.\n2) Display complete inventory in sorted order of item names as well as item\_id.\n3) Remove Item.\n4) Exit");

choice=sc.nextInt();

switch(choice){

case 1:

System.out.println("Enter your details as follows");

System.out.println("Enter Item you want add");

//int n=sc.nextInt();

for(int i=1;i<=1;i++){

System.out.println("Enter id of item ");

int d=sc.nextInt();

I.setitem\_id(d);

System.out.println("Enter name of item ");

sc.nextLine();

String ss=sc.nextLine();

I.setitem\_name(ss);

if (!list.contains(I)){

list.add(I);}

}

System.out.println("Added items as follows");

System.out.println(list);

break;

case 2:

System.out.println("before sorting");

System.out.println(list);

System.out.println("Sorting by id");

idsort n2=new idsort();

Collections.sort(list,n2);

System.out.println(list);

System.out.println("Sorting by name");

namesort n1=new namesort();

Collections.sort(list,n1);

System.out.println(list);

break;

case 3:

System.out.println("List as follows");

System.out.println(list);

System.out.println("Enter index od item which you want to remove index start from 0");

int re=sc.nextInt();

list.remove(re);

System.out.println("List after removal");

System.out.println(list);

break;

case 4:

System.out.println("Thank you");

break;

}

}while(choice!=4);

}

}

Q2 output

Text

Description automatically generated