

PMR: This project was simpler because it was based of the previous projects. I used the previous project as a guideline for this project. It was also helpful that there was a screenshot at the end of the assignment as a guideline for what to do.

CAR

public class Car extends Vehicle {

public Car(String name, double cost){

super(name, cost);

}

}

INVENTORYDEMO

import java.util.ArrayList;

public class InventoryDemo {

public static ArrayList<Product> p = new ArrayList<Product>();

public static void main(String[] args){

Car c1 = new Car("Jaguar", 1000000.00);

Car c2= new Car("Neon", 17000.00);

Tool t1 = new Tool("JigSaw", 149.18);

Car c3 = new Car("Jaguar", 110000.00);

Car c4 = new Car("Neon", 17500.00);

Car c5 = new Car("Neon", 17875.32);

Truck tr1 = new Truck("RAM", 35700.00);

Tool t2 = new Tool("CircularSaw", 200.00);

Tool t3 = new Tool("CircularSaw", 150.00);

p.add(c1);

p.add(c2);

p.add(t1);

p.add(c3);

p.add(c4);

p.add(c5);

p.add(tr1);

p.add(t2);

p.add(t3);

System.out.println("Name Cost");

takeInventory("JigSaw");

takeInventory("Neon");

takeInventory("Jaguar");

takeInventory("RAM");

takeInventory("CircularSaw");

if (t1.compareTo(t2) == 1){

System.out.println("The first saw is more expensive");

}

else if (t1.compareTo(t2) == 0){

System.out.println("The saws are the same price.");

}

else{

System.out.println("The second saw is more expensive");

}

}

public static void takeInventory(String name){

int counter = 0;

double totalCost = 0.0;

for(Product pr : p){

if(pr.getName().equalsIgnoreCase(name)){

counter++;

totalCost += pr.getCost();

}

}

System.out.println(name + ": Quantity = " +counter + ", Total Cost = " + totalCost);

}

}

PRODUCT

public interface Product {

public abstract String getName();

public abstract double getCost();

}

TOOL

public class Tool implements Product, Comparable<Tool> {

private String name;

private double cost;

public Tool(String name, double cost){

this.name = name;

this.cost = cost;

}

public String getName(){

return name;

}

public double getCost(){

return cost;

}

public int compareTo(Tool obj){

if(cost > obj.cost){

return 1;

}

else if(cost == obj.cost){

return 0;

}

else return -1;

}

}

TRUCK

public class Truck extends Vehicle {

public Truck(String name, double cost){

super(name, cost);

}

}

VEHICLE

public abstract class Vehicle implements Product {

private String name;

private double cost;

public Vehicle(String name, double cost){

this.name = name;

this.cost = cost;

}

public String getName(){

return name;

}

public double getCost(){

return cost;

}

}