**package TestProduct;**

import java.text.\*;

import javax.swing.\*;

import java.util.ArrayList;

import java.text.SimpleDateFormat;

import java.util.Date;

import javax.swing.JOptionPane;

**public class TestProduct{**

// Get the date

static SimpleDateFormat Date\_Format = new SimpleDateFormat("MM/dd/yyyy");

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

// Declare variables

String Address\_street, Address\_city, Address\_state, Address\_zip, Company\_Name, Product\_Name, DateOfManufacture;

int Product\_Quantity;

double Product\_Price;

Product product;

Manufacturer manufacturer;

Address Full\_Address;

Database Product\_Data = new Database();

Database Deleted\_Product\_Data = new Database();

Date current;

boolean isDone = false;

int subMenu;

// loop until, the program is done

while (!isDone){

// prompt the user for the menu

int selectedMenu = GetData.get\_Int(

"Welcome to Inventory\n"

+ "\t1. Create\n"

+ "\t2. Update (Quantity/Price)\n"

+ "\t3. Remove\n"

+ "\t4. Search product\n"

+ "\t5. Inventory Report\n"

+ "\t6. List of Deleted Products\n"

+ "\t7. Quit");

// Switch for menu

switch (selectedMenu){

case 1://add items

// Get user input regarding product

Product\_Name = GetData.get\_String("Enter product name: ");

Product\_Quantity = GetData.get\_Int("Enter the quantity of product: ");

Product\_Price = GetData.get\_Double("Enter the unit price of product(in $): ");

Company\_Name = GetData.get\_String("Enter the name of the manufacturer company: ");

Address\_street = GetData.get\_String("Enter street address of company: ");

Address\_city = GetData.get\_String("Enter city address of company: ");

Address\_state = GetData.get\_String("Enter state address of company: ");

Address\_zip = GetData.get\_String("Enter ZIP code of company: ");

DateOfManufacture = GetData.get\_String("Enter date of manufacturing product in the form of(MM/dd/yyyy): ");

Full\_Address = new Address(Address\_street, Address\_city, Address\_state, Address\_zip);

manufacturer = new Manufacturer(Company\_Name, Full\_Address);

current = Date\_Format.parse(DateOfManufacture);

product = new Product(Product\_Name, Product\_Quantity, Product\_Price, current, manufacturer);

// add the product to the data base

Product\_Data.add(product);

break;

case 2://Update product data

// Get what to update

Product\_Name = GetData.get\_String("Enter name of the product to update: ");

subMenu = GetData.get\_Int("Enter 1 or 2: \n"

+ "1. Update quantity\n"

+ "2. Update price\n");

// search product

Product\_Data.search(Product\_Name);

if (!Product\_Data.inList()){//if product is not found

JOptionPane.showMessageDialog(null, "Product does not exists.");

}else{//if product is found

switch (subMenu){

case 1: //Update product Product\_Quantity

int menuOpt = GetData.get\_Int("Enter 1 or 2: \n"

+ "1. Add quantity\n"

+ "2. Remove quantity\n");

int addRDeductQuant;

// Condition to check for the valid input

if (menuOpt < 1 || menuOpt > 2){//Invalid input

JOptionPane.showMessageDialog(null, "Please choose the correct choice");

}

else{//Valid input

switch (menuOpt){

case 1://Add Product\_Quantity

addRDeductQuant = GetData.get\_Int("Enter quantity to add: ");

// Update the product in the database

product = Product\_Data.getProduct();

product.update\_Product\_Quantity(addRDeductQuant);

JOptionPane.showMessageDialog(null,"Record updated successfully.");

break;

case 2://Remove Product\_Quantity

addRDeductQuant = GetData.get\_Int("Enter quantity to remove: ");

// Update the product in the database

product = Product\_Data.getProduct();

if (product.get\_Product\_Quantity()-addRDeductQuant<0){

JOptionPane.showMessageDialog(null,"Entered quantity exceeds quantity in the inventory.");

}else{

product.update\_Product\_Quantity(addRDeductQuant \* -1);

JOptionPane.showMessageDialog(null,"Record updated successfully.");

}

break;

}

}

break;

case 2://Update the price

// Get new price

double priceToUpdate = GetData.get\_Double("Enter the unit price of product to update: ");

// Price validation

if (priceToUpdate < 0){

JOptionPane.showMessageDialog(null,"Price of the product is not updated.");

}else{

// Update the database

product = Product\_Data.getProduct();

product.update\_Product\_Price(priceToUpdate);

JOptionPane.showMessageDialog(null,"Price updated successfully.");

}

break;

default:

JOptionPane.showMessageDialog(null,"Invalid option");

}

}

break;

case 3://Delete product

// Get product name

Product\_Name = GetData.get\_String("Enter product name: ");

// Get product from database

Product\_Data.search(Product\_Name);

if (Product\_Data.inList()){//if product exists

int index = Product\_Data.getIndex();

// Add the product to the deleted database

Deleted\_Product\_Data.add(Product\_Data.getProduct());

// delete the product from the product database

Product\_Data.delete(index);

JOptionPane.showMessageDialog(null, "Product deleted successfully.");

}else JOptionPane.showMessageDialog(null, "Product not found.");//if product does not exists

break;

case 4://Search product

Product\_Name = GetData.get\_String("Enter product name: ");

// Search database for the product

Product\_Data.search(Product\_Name);

if (Product\_Data.inList()){ //If product exists

Display\_Single\_Product(Product\_Data.getProduct(),JOptionPane.INFORMATION\_MESSAGE);

}else JOptionPane.showMessageDialog(null, "Product not found."); //If product does not exists

break;

case 5: //Inventory report

if (Product\_Data.getList() != null){ // If inventory has some records

Display\_Product\_Inventory(Product\_Data,JOptionPane.INFORMATION\_MESSAGE);

}else JOptionPane.showMessageDialog(null,"No products present."); //If no record exists

break;

case 6: //Deleted products

if (Deleted\_Product\_Data.getList() != null){ //If deleted database has some records

Display\_Product\_Inventory(Deleted\_Product\_Data,JOptionPane.INFORMATION\_MESSAGE);

}else JOptionPane.showMessageDialog(null,"No products present.");//If no record exists

break;

case 7://Exit

isDone = true;

break;

default: //Other than 1-7 input

JOptionPane.showMessageDialog(null,"Unable to process the option selected from the menu list.");

}

}

}

// Get product info into a specific format

public static String ProductInfo\_Format(Product info){

String ProdInfo = String.format("%30s", info.get\_Product\_Name());

ProdInfo += String.format("%30s", Date\_Format.format(info.get\_Product\_Date()));

ProdInfo += String.format("%30s", info.get\_Manufacturer().get\_Company\_Name());

return ProdInfo;

}

// Get deleted products list

public static void Deleted\_Prod\_Display(Database productDB,int Type\_Message){

String inventResult = null;

ArrayList<Product> prodList = productDB.getList();

inventResult += String.format("%30s %30s %30s", "Product","Purchase Date", "Manufacturer");

for (int i = 0; i < productDB.size(); i++){

inventResult += ProductInfo\_Format(prodList.get(i)) + "\n";

}

JTextArea text = new JTextArea(inventResult, 10, 50);

JScrollPane pane = new JScrollPane(text);

JOptionPane.showMessageDialog(null, pane, "Deleted Inventory Details", Type\_Message);

}

// Display inventory

public static void Display\_Product\_Inventory(Database productDB, int Type\_Message){

String inventResult = "";

ArrayList<Product> prodList = productDB.getList();

inventResult += String.format("%-30s \t%s %10s %15s %20s %15s\n","Product", "Purchase Date", "Quantity", "Price($)", "Manufacturer", "State");

for (int i = 0; i < productDB.size(); i++){

inventResult += prodList.get(i).get\_Product\_Infomation() + "\n";

}

JTextArea text = new JTextArea(inventResult, 10, 60);

JScrollPane pane = new JScrollPane(text);

JOptionPane.showMessageDialog(null, pane, "Inventory Details", Type\_Message);

}

// Single product info

public static void Display\_Single\_Product(Product product, int Type\_Message){

String Product\_Info = "Product Name: " + product.get\_Product\_Name() + "\n";

Product\_Info += String.format("Unit Price: $%.2f", product.get\_Product\_Price()) + "\n";

Product\_Info += "Quantity: " + product.get\_Product\_Quantity() + "\n";

JTextArea text = new JTextArea(Product\_Info, 10, 30);

JScrollPane pane = new JScrollPane(text);

JOptionPane.showMessageDialog(null, pane, product.get\_Product\_Name() + " Details", Type\_Message);

}

}

// Product database class

**class Database{**

private ArrayList<Product> list;

private Product prod;

private int index;

private boolean found;

public Database(){

list = new ArrayList<Product>();

prod = null;

index = 0;

found = false;

}

// Product search method

public void search(String key){

found = false;

int i = 0;

while (!found && i < list.size()){

Product b = list.get(i);

if (b.get\_Product\_Name().equalsIgnoreCase(key)){

prod = b;

found = true;

index = i;

}

else i++;

}

}

// Add the product to the database

public void add(Product newProduct){

list.add(newProduct);

}

// Remove the product from the database

public Product delete(int i){

return list.remove(i);

}

// Get index of current product

public int getIndex(){

return index;

}

// If product in the database

public boolean inList(){

return found;

}

// Current product

public Product getProduct(){

return prod;

}

// List size

public int size(){

return list.size();

}

// If list is empty

public boolean isEmpty(){

return list.isEmpty();

}

// Get product list

public ArrayList<Product> getList(){

return list;

}

}

// Product info class

**class Product{**

SimpleDateFormat Date\_Format = new SimpleDateFormat("MM/dd/yyyy");

Manufacturer manufacturer;

String Product\_Name;

int Product\_Quantity;

double Product\_Price;

Date Product\_Date;

public Product(){

this.Product\_Name = "";

this.Product\_Quantity = 0;

this.Product\_Price = 0.0;

this.Product\_Date = null;

this.manufacturer = null;

}

// Customized constructor

public Product(String prod\_Name, int prod\_Quantity, double prod\_Price, Date product\_Created, Manufacturer manufact){

this.Product\_Name = prod\_Name;

this.Product\_Quantity = prod\_Quantity;

this.Product\_Price = prod\_Price;

this.Product\_Date = product\_Created;

this.manufacturer = manufact;

}

// Get product date

public Date get\_Product\_Date(){

return Product\_Date;

}

// Add product date

public void set\_Product\_Date(Date productCreated){

this.Product\_Date = productCreated;

}

// Get manufacturer

public Manufacturer get\_Manufacturer(){

return manufacturer;

}

// Add Manufacturer

public void set\_Manufacturer(Manufacturer manufacture){

this.manufacturer = manufacture;

}

// Get product name

public String get\_Product\_Name(){

return Product\_Name;

}

// Set product name

public void set\_Product\_Name(String prodName){

this.Product\_Name = prodName;

}

// Get product quantity

public int get\_Product\_Quantity(){

return Product\_Quantity;

}

// Set Product Quantity

public void set\_Product\_Quantity(int quantity){

this.Product\_Quantity = quantity;

}

// Get product price

public double get\_Product\_Price(){

return Product\_Price;

}

// Set product price

public void set\_Product\_Price(double unitPrice){

this.Product\_Price = unitPrice;

}

// Update product quantity

public void update\_Product\_Quantity(int quantity\_upDate){

Product\_Quantity += quantity\_upDate;

}

// Updtae product price

public void update\_Product\_Price(double price\_upDate){

this.Product\_Price = price\_upDate;

}

// Get product info

public String get\_Product\_Infomation(){

String result = "";

result += String.format("%-30s", Product\_Name);

String dateForm = Date\_Format.format(Product\_Date);

result += String.format("\t %s", dateForm);

result += String.format("%10d", Product\_Quantity);

result += String.format("\t%15.2f", Product\_Price);

result += String.format("\t%15s", manufacturer.get\_Company\_Name());

result += String.format("\t%20s", manufacturer.get\_Company\_Address().get\_State());

return result;

}

}

// Product manufacturer class

**class Manufacturer{**

private String companyName;

private Address companyAddress;

public Manufacturer(){

this.companyName = "";

this.companyAddress = null;

}

// Customized constructor

public Manufacturer(String compName, Address address){

this.companyName = compName;

this.companyAddress = address;

}

// Get manufacturer company name

public String get\_Company\_Name(){

return companyName;

}

// Set manufacturer company name

public void set\_Company\_Name(String companyName){

this.companyName = companyName;

}

// Get manufacturer address

public Address get\_Company\_Address(){

return companyAddress;

}

// Set manufacturer address

public void set\_Company\_Address(Address address){

this.companyAddress = address;

}

}

// Manufacturer address class

**class Address{**

private String street, city, state, zip;

// Customized constructor

public Address(String str, String city, String st, String zip){

street = str;

this.city = city;

state = st;

this.zip = zip;

}

// Get manufacturer street address

public String get\_Street(){

return street;

}

// Get manufacturer City address

public String get\_City(){

return city;

}

// Get manufacturer state address

public String get\_State(){

return state;

}

// Get manufacturer zipcode

public String get\_Zip(){

return zip;

}

}

// Retrieve data class

**class GetData{**

public static double get\_Double(String s){

return Double.parseDouble(get\_Word(s));

}

public static int get\_Int(String s){

return Integer.parseInt(get\_Word(s));

}

public static String get\_Word(String s){

return JOptionPane.showInputDialog(s);

}

public static String get\_String(String s){

return JOptionPane.showInputDialog(s);

}

}