Product.java

import java.util.\*;

import java.text.SimpleDateFormat;

import java.lang.\*;

public class product

{

protected String Description;

protected double ReccPrice;

protected int unit =1;

protected int Weight;

protected static final long packInmillies = -2209024800000L;

protected String unitType;

protected SimpleDateFormat packing = new SimpleDateFormat("dd-MM-yyyy hh:mm:ss");

protected SimpleDateFormat expiry = new SimpleDateFormat("dd-MM-yyyy");

protected StringBuilder ID = new StringBuilder();

protected Date packingDate = new Date(packInmillies);

protected Date expiryDate;

public product()

{

}

public product( String Description, double ReccPrice, int unit, int Weight)

{

this.Description = Description;

this.ReccPrice = ReccPrice;

this.unit = unit;

this.Weight = Weight;

}

public void comparePrice(double input)

{

if (ReccPrice > input)

System.out.println("More than reccomended price");

else if (ReccPrice < input)

System.out.println("Cheaper than reccomended price");

else

System.out.println("Same price as reccomended");

}

public void editProductInformation( String Description, int unit, double price, int weight) {

this.Description = Description;

this.unit = unit;

this.ReccPrice = price;

this.Weight = weight;

}

public void verifyExpireProduct(Date input)

{

if (expiryDate.after(input)) {

System.out.println("This item has not yet expired");

}

if (expiryDate.before(input)) {

System.out.println("This item is expired");

}

if (expiryDate.equals(input)) {

System.out.println("The given date is the expiry date");

}

}

public String getInformationForSavingToFile()

{

return ID.toString() + ", " + Description + ", " + ReccPrice + ", " + unit + ", " + Weight + ", " + packingDate + ", " + expiryDate;

}

public String toString()

{

return "ID: " + ID.toString() + "\nDescription: " + Description + "\nReccomended price: " + ReccPrice + "\nUnit: " + unit + "\nWeight: " + Weight + "\nPacking date: " + packingDate + "\nExpire Date: " + expiry.format(expiryDate);

}

public String getAttributeDescriptionForSavingTofile()

{

return "ID, Description, Recomended Price, Unit, Weight, Unit type, Packing Date, Expire Date";

}

public static ArrayList<product> sortByPrice( ArrayList<product> list1)

{

ArrayList<product> sorted = list1;

Comparator<product> c = new Comparator<product>()

{

public int compare(product a, product b)

{

return Double.compare(a.getPrice(), b.getPrice());

}

};

Collections.sort(sorted, c);

return sorted;

}

public static ArrayList<product> findProductFromDescription(ArrayList<product> shoppingCart, String input) {

ArrayList<product> find = new ArrayList<product>();

for (product product : shoppingCart) {

if(product.getDescription().contains(input)==true)

{

find.add(product);

System.out.println(product.getClass().getName()+" matches the description of " + input);

}

else

{

}

}

return find;

}

public static String allProductToString(ArrayList<product> cart)

{

StringBuilder Allinfo = new StringBuilder();

for (product i : cart)

{

Allinfo.append(i.toString());

Allinfo.append(System.getProperty("line.separator"));

Allinfo.append(System.getProperty("line.separator"));

}

return Allinfo.toString();

}

public void verifyUnitType(String input)

{

if (input.equals(unitType))

{

System.out.println("The type input matches with the item");

}

else

{

System.out.println("The type input doesn't matches with the item");

}

}

public double getPrice()

{

return ReccPrice;

}

public String getDescription()

{

return Description;

}

}

Packed.java

import java.util.\*;

import java.text.SimpleDateFormat;

import java.lang.\*;

import java.text.ParseException;

public class packed extends product{

private static final long expireInmillies = 32503683661000L;

protected Dimension Dimension = new Dimension(0,0,0);

protected Nutrition Nutrition;

protected UnitTypeLimit UnitType;

//Overloaded with expiry Date

public packed(String IDinput, String Description, double ReccPrice, int unit, int Weight, Dimension dimension, UnitTypeLimit unitType, String inputPacking, Nutrition nutrition)

{

this(IDinput, Description, ReccPrice, unit, Weight, dimension, unitType, inputPacking, "01-01-3000", nutrition);

}

public packed(String IDinput, String Description, double ReccPrice, int unit, int Weight, Dimension dimension, UnitTypeLimit unitType, String inputPacking, String inputExpiry, Nutrition nutrition)

{

super(Description, ReccPrice, unit, Weight);

expiryDate = new Date(expireInmillies);

ID.append("2");

this.Description = Description;

this.Dimension = dimension;

this.ReccPrice = ReccPrice;

this.unit = unit;

this.Weight = Weight;

this.UnitType = unitType;

this.Nutrition = nutrition;

try {

packingDate = packing.parse(inputPacking);

expiryDate = expiry.parse(inputExpiry);

}

catch (ParseException e) {

System.out.println("wow dog");

}

if (IDinput.length()==6) {

ID.append(IDinput);

}

else

{

System.out.println("the ID number is either too short or too long please try again");

}

}

@Override

public void verifyUnitType(String input)

{

if (input == UnitType.returnType())

{

System.out.println("The type input matches with the item");

}

else

{

System.out.println("The type input doesn't matches with the item");

}

}

@Override

public String getInformationForSavingToFile()

{

return super.getInformationForSavingToFile() + ", " + UnitType + ", " + Dimension.getWidth() + ", " + Dimension.getHeight() + ", " + Dimension.getDepth() + ", " + Nutrition.getInformationForSavingToFile();

}

@Override

public String toString()

{

return super.toString() + "\nUnit Type" + UnitType + "\nDimension" + Dimension +"\nNutrition" + Nutrition;

}

@Override

public String getAttributeDescriptionForSavingTofile()

{

return super.getAttributeDescriptionForSavingTofile() + ", UnitType, Width, Height, Depth, Quantity, Engergy, Protein, Fat, Sugar, Salt";

}

public int getVolumn()

{

return Dimension.getHeight() \* Dimension.getWidth() \* Dimension.getDepth();

}

}

class Dimension

{

private int height = 0;

private int width = 0;

private int depth = 0;

public Dimension(int height, int width, int depth)

{

this.height = height;

this.width = width;

this.depth = depth;

}

public Dimension()

{

}

public String toString()

{

return ("\nwidth: " + width + "\nheight: " + height + "\ndepth: " + depth);

}

public int getHeight()

{

return height;

}

public int getWidth()

{

return width;

}

public int getDepth()

{

return depth;

}

}

class Nutrition

{

private int Quantity;

private int Engergy;

private int Protein;

private int Fat;

private int Sugar;

private int Salt;

private Dimension Dimension = new Dimension();

public Nutrition(String full)

{

String[] address = full.split("/",6);

this.Quantity = Integer.parseInt(address[0]);

this.Engergy = Integer.parseInt(address[1]);

this.Protein = Integer.parseInt(address[2]);

this.Fat = Integer.parseInt(address[3]);

this.Sugar = Integer.parseInt(address[4]);

this.Salt = Integer.parseInt(address[5]);

}

public String toString()

{

return ( "\nQuantity: "+ Quantity + "\nEngergy: " + Engergy + "\nProtein: " + Protein + "\nFat: " + Fat + "\nSugar: " + Sugar + "\nSalt: " + Salt);

}

public String getInformationForSavingToFile()

{

return(Quantity + ", " + Engergy + ", " + Protein + ", " + Fat + ", " + Sugar + ", " + Salt);

}

}

enum UnitTypeLimit {

Box, Bottle, Pack;

public String returnType()

{

return this.toString();

}

};

PackedMeat.java

public class packedMeat extends packed{

private String meatType;

private int grade;

//Overloaded with Dimension, packin date, expiryDate

public packedMeat (String IDinput, String Description, double ReccPrice, int unit, int Weight, UnitTypeLimit unitType, Nutrition nutrition, String meatType, int grade)

{

this(IDinput, Description, ReccPrice, unit, Weight, new Dimension(0,0,0), unitType, "01-01-1900 00:00:01", "01-01-3000", nutrition, meatType, grade);

}

public packedMeat (String IDinput, String Description, double ReccPrice, int unit, int Weight, Dimension dimension, UnitTypeLimit unitType, String inputPacking, String inputExpiry, Nutrition nutrition, String meatType, int grade)

{

super( IDinput, Description, ReccPrice, unit, Weight, dimension, unitType, inputPacking, inputExpiry, nutrition);

this.meatType = meatType;

this.grade = grade;

}

@Override

public String toString()

{

return super.toString() + "\nMeat Type: " + meatType + "\nGrade: " + grade;

}

@Override

public String getAttributeDescriptionForSavingTofile()

{

return super.getAttributeDescriptionForSavingTofile() + ", Meat type, Grade";

}

@Override

public String getInformationForSavingToFile()

{

return super.getInformationForSavingToFile() + ", " + meatType + ", " + grade;

}

}

PackedMRE.java

public class packedMRE extends packed{

private String Producer;

private String Code;

//Overloaded with Dimension, packing Date and expiry Date

public packedMRE (String IDinput, String Description, double ReccPrice, int unit, int Weight, UnitTypeLimit unitType, Nutrition nutrition, String Producer, String code)

{

this(IDinput, Description, ReccPrice, unit, Weight, new Dimension(0,0,0), unitType, "01-01-1900 00:00:01", "01-01-3000", nutrition, Producer, code);

}

public packedMRE(String IDinput, String Description, double ReccPrice, int unit, int Weight, Dimension dimension, UnitTypeLimit unitType, String inputPacking, String inputExpiry, Nutrition nutrition, String Producer, String code)

{

super( IDinput, Description, ReccPrice, unit, Weight, dimension, unitType, inputPacking, inputExpiry, nutrition);

this.Producer = Producer;

this.Code = code;

}

@Override

public String getInformationForSavingToFile()

{

return super.getInformationForSavingToFile() + ", " + Producer + ", " + Code;

}

@Override

public String getAttributeDescriptionForSavingTofile()

{

return super.getAttributeDescriptionForSavingTofile() + ", Producer, Code";

}

@Override

public String toString()

{

return super.toString() + "\nProducer: " + Producer + "\nCode: " + Code;

}

}

Fresh.java

import java.util.\*;

import java.text.SimpleDateFormat;

import java.lang.\*;

import java.text.ParseException;

public class fresh extends product{

//Overloaded with Packing Date

public fresh(String IDinput, String Description, double ReccPrice, int unit, int Weight, String unitType, String inputExpiry)

{

this(IDinput, Description, ReccPrice, unit, Weight, unitType, "01-01-1900 01:01:01", inputExpiry);

}

public fresh(String IDinput, String Description, double ReccPrice, int unit, int Weight, String unitType, String inputPacking, String inputExpiry)

{

super(Description, ReccPrice, unit, Weight);

expiryDate = new Date();

ID.append("1");

this.Description = Description;

this.ReccPrice = ReccPrice;

this.unit = unit;

this.Weight = Weight;

this.unitType = unitType;

try {

packingDate = packing.parse(inputPacking);

expiryDate = expiry.parse(inputExpiry);

}

catch (ParseException e) {

System.out.println("wow dog");

}

if (IDinput.length()==6) {

ID.append(IDinput);

}

else

{

System.out.println("the ID number is either too short or too long please try again");

}

}

public String getInformationForSavingToFile()

{

return super.getInformationForSavingToFile() + ", " + unitType;

}

public String getAttributeDescriptionForSavingTofile()

{

return super.getAttributeDescriptionForSavingTofile()+ ", Unit Type";

}

@Override

public String toString()

{

return super.toString() + "\nUnit Type" + unitType;

}

}

Fresh.java

import java.util.\*;

import java.text.SimpleDateFormat;

import java.lang.\*;

import java.text.ParseException;

public class freshCake extends fresh{

private int CakeDiameter;

private String BakerName;

//Overloaded with unit and packing Date

public freshCake(String IDinput, String Description, double ReccPrice, int Weight, String unitType, String inputExpiry, int CakeDiameter, String BakerName)

{

this(IDinput, Description, ReccPrice, 1, Weight, unitType, "01-01-1900 00:00:01", inputExpiry, CakeDiameter, BakerName);

}

public freshCake(String IDinput, String Description, double ReccPrice, int unit, int Weight, String unitType, String inputPacking, String inputExpiry, int CakeDiameter, String BakerName)

{

super(IDinput, Description, ReccPrice, unit, Weight, unitType, inputPacking, inputExpiry);

this.CakeDiameter = CakeDiameter;

this.BakerName = BakerName;

}

@Override

public String getInformationForSavingToFile()

{

return super.getInformationForSavingToFile() + ", " + CakeDiameter + ", " + BakerName;

}

@Override

public String getAttributeDescriptionForSavingTofile()

{

return super.getAttributeDescriptionForSavingTofile() + ", Cake Diameter, BakerName";

}

@Override

public String toString()

{

return super.toString() + "\nCake Diameter: " + CakeDiameter + "\nBakerName " + BakerName;

}

}

FreshSpice.java

import java.util.\*;

import java.text.SimpleDateFormat;

import java.lang.\*;

import java.text.ParseException;

public class freshSpice extends fresh{

private String jar;

private String components;

//Overloaded with unit and packing date

public freshSpice(String IDinput, String Description, double ReccPrice, int Weight, String unitType, String inputExpiry, String jar, String components)

{

this(IDinput, Description, ReccPrice, 1, Weight, unitType, "01-01-1900 00:00:01", inputExpiry, jar, components);

}

public freshSpice(String IDinput, String Description, double ReccPrice, int unit, int Weight, String unitType, String inputPacking, String inputExpiry, String jar, String components)

{

super(IDinput, Description, ReccPrice, unit, Weight, unitType, inputPacking, inputExpiry);

this.jar = jar;

this.components = components;

}

@Override

public String getInformationForSavingToFile()

{

return super.getInformationForSavingToFile() + ", " + jar + ", " + components;

}

@Override

public String getAttributeDescriptionForSavingTofile()

{

return super.getAttributeDescriptionForSavingTofile() + ", jar, components";

}

@Override

public String toString()

{

return super.toString() + "\njar: " + jar + "\ncomponents: " + components;

}

}

Household.java

import java.util.\*;

import java.text.SimpleDateFormat;

import java.lang.\*;

import java.text.ParseException;

public class household extends product{

private static final long expireInmillies = 32503683661000L;

protected Dimension Dimension = new Dimension(0,0,0);

//Overloaded with expiry Date

public household(String IDinput, String Description, double ReccPrice, int unit, int Weight, Dimension dimension, String unitType, String inputPacking)

{

this(IDinput, Description, ReccPrice, unit, Weight, dimension, unitType, inputPacking, "01-01-3000");

}

public household(String IDinput, String Description, double ReccPrice, int unit, int Weight, Dimension dimension, String unitType, String inputPacking, String inputExpiry)

{

super(Description, ReccPrice, unit, Weight);

packingDate = new Date(-30610200000000L);

expiryDate = new Date(expireInmillies);

ID.append("3");

this.Description = Description;

this.Dimension = dimension;

this.ReccPrice = ReccPrice;

this.unit = unit;

this.Weight = Weight;

this.unitType = unitType;

try {

packingDate = packing.parse(inputPacking);

expiryDate = expiry.parse(inputExpiry);

}

catch (ParseException e) {

System.out.println("wow dog");

}

if (IDinput.length()==6) {

ID.append(IDinput);

}

else

{

System.out.println("the ID number is either too short or too long please try again");

}

}

public String getInformationForSavingToFile()

{

return super.getInformationForSavingToFile() + ", " + unitType + ", " + Dimension.getWidth() + ", " + Dimension.getHeight() + ", " + Dimension.getDepth();

}

public String toString()

{

return super.toString() + "\nUnit Type" + unitType + "\nDimension: " + Dimension;

}

public String getAttributeDescriptionForSavingTofile()

{

return super.getAttributeDescriptionForSavingTofile() + ", Unit Type, Width, Height, Depth,";

}

public int getVolumn()

{

return Dimension.getHeight() \* Dimension.getWidth() \* Dimension.getDepth();

}

}

HouseHoldKitchen.java

public class householdKitchenWare extends household{

private String Manufacturer;

private String Series;

//Overloaded with ExpiryDate

public householdKitchenWare(String IDinput, String Description, double ReccPrice, int unit, int Weight, Dimension dimension, String unitType, String inputPacking, String Manufacturer, String Series)

{

this(IDinput, Description, ReccPrice, unit, Weight, dimension, unitType, inputPacking, "01-01-3000", Manufacturer, Series );

}

public householdKitchenWare(String IDinput, String Description, double ReccPrice, int unit, int Weight, Dimension dimension, String unitType, String inputPacking, String inputExpiry, String Manufacturer, String Series)

{

super(IDinput, Description, ReccPrice, unit, Weight, dimension, unitType, inputPacking, inputExpiry);

this.Manufacturer = Manufacturer;

this.Series = Series;

}

public String getInformationForSavingToFile()

{

return super.getInformationForSavingToFile() + ", " + Manufacturer + ", " + Series;

}

public String toString()

{

return super.toString() + "\nManufacturer: " + Manufacturer + "\nSeries: " + Series;

}

public String getAttributeDescriptionForSavingTofile()

{

return super.getAttributeDescriptionForSavingTofile() + ", Manufacturer, Series";

}

}

HouseholdStationery.java

public class householdStationery extends household{

private String inkType;

private String penType;

//Overloaded with expiry Date

public householdStationery(String IDinput, String Description, double ReccPrice, int unit, int Weight, Dimension dimension, String unitType, String inputPacking, String inkType, String penType)

{

this(IDinput, Description, ReccPrice, unit, Weight, dimension, unitType, inputPacking, "01-01-3000", inkType, penType);

}

public householdStationery(String IDinput, String Description, double ReccPrice, int unit, int Weight, Dimension dimension, String unitType, String inputPacking, String inputExpiry, String inkType, String penType)

{

super(IDinput, Description, ReccPrice, unit, Weight, dimension, unitType, inputPacking, inputExpiry);

this.inkType = inkType;

this.penType = penType;

}

public String getInformationForSavingToFile()

{

return super.getInformationForSavingToFile() + ", " + inkType + ", " + penType;

}

public String toString()

{

return super.toString() + "\ninkType: " + inkType + "\npenType: " + penType;

}

public String getAttributeDescriptionForSavingTofile()

{

return super.getAttributeDescriptionForSavingTofile() + ", inkType, penType";

}

}

Test.java

import java.lang.reflect.Field;

import java.text.ParseException;

import java.util.\*;

import java.lang.\*;

import java.io.\*;

import java.text.\*;

public class test

{

public static void main (String[] args)

{

//create objects

packedMRE Spam = new packedMRE("123456", "it's spam", 4.45, 1, 300, new Dimension(10,10,10), UnitTypeLimit.Bottle, "14-12-1900 09:09:09", "14-12-2019" , new Nutrition("100/122/200/300/400/400"),"American Airforce","X5632");

packedMeat Wagyu9 = new packedMeat("123457", "grade9 wagyu beef", 4.45, 1, 300, new Dimension(30,30,30), UnitTypeLimit.Box, "14-12-1900 09:09:09", "14-12-2010" , new Nutrition("100/122/200/300/400/400"),"Japanese Wagyu",9);

freshSpice Weistersauce = new freshSpice("234567", "german sauce", 2.25, 20, 50, "Can", "11-11-2000 08:08:00", "11-11-2011", "mason jar", "pepper, barbeque sauce");

freshCake PoundCake = new freshCake("234567", "german cakes", 2.25, 20, 50, "Cake", "11-11-2000 08:08:00", "11-11-2011", 10, "Andrew Osborn");

householdStationery Bently\_pen = new householdStationery("234567", "Bently electric pen", 80, 1, 40, new Dimension(1,30,50), "case", "11-09-2018 11:15:00", "09-12-2025", "blue ink", "electric pen");

householdKitchenWare Yoshihiro\_knife = new householdKitchenWare("234567", "Fish knife", 80, 1, 200, new Dimension(4,35,30), "unit", "11-11-2000 08:08:00", "11-11-2011", "Petty", "chief knife");

//create array list for objects

ArrayList<product> shoppingCart = new ArrayList<product>();

shoppingCart.add(Spam);

//add objects to arraylist

shoppingCart.add(Wagyu9);

shoppingCart.add(Weistersauce);

shoppingCart.add(PoundCake);

shoppingCart.add(Bently\_pen);

shoppingCart.add(Yoshihiro\_knife);

//Core level functions

//VerifyUnitType on 3 categories

//Spam.verifyUnitType("Bottle");//returns equal

//Weistersauce.verifyUnitType("mason jar");// returns equal

//Bently\_pen.verifyUnitType("Bottle");//returns not equal

//getAttributeDescriptionForSavingTofile

//System.out.println(Spam.getAttributeDescriptionForSavingTofile());

//System.out.println(Weistersauce.getAttributeDescriptionForSavingTofile());

//System.out.println(Bently\_pen.getAttributeDescriptionForSavingTofile());

//getInformationForSavingToFile

//System.out.println(Spam.getInformationForSavingToFile());

//System.out.println(Weistersauce.getInformationForSavingToFile());

//ystem.out.println(Bently\_pen.getInformationForSavingToFile());

//Standard level functions

//verifyExpireProduct on 3 categories using date as input

//Spam.verifyExpireProduct(parseDate("14-11-2019"));// not expire yet

//Weistersauce.verifyExpireProduct(parseDate("11-11-2011"));// same date as expire date

//Bently\_pen.verifyExpireProduct(parseDate("11-12-2026"));//expired

//getVolumn

//System.out.println(Spam.getVolumn());

//System.out.println(Bently\_pen.getVolumn());

//editProductInformation

//Spam.editProductInformation("it's not spam", 20, 34.5, 10);

//System.out.println(Spam);

//comparePrice

//Spam.comparePrice(1);

//Weistersauce.comparePrice(80.43);

//Yoshihiro\_knife.comparePrice(80);

//Advanced level functions

//findProductFromDescription print out lines and also return an arrarylist

//product.findProductFromDescription(shoppingCart, "german");

//sortByPrice sort the arraylist by price

//product.sortByPrice(shoppingCart);

//allProductToString return one single string with all the items in it using toString() of the items

System.out.println(product.allProductToString(shoppingCart));

}

public static Date parseDate(String date) {

try {

return new SimpleDateFormat("dd-MM-yyyy").parse(date);

} catch (ParseException e) {

return null;

}

}

}

