**Clase n°8, ejercicio 1 (guía ejercicio 8)**

**Autor: Vargas Diego**

Ejercicio 1.

import java.io.File;

import java.io.IOException;

import java.nio.charset.StandardCharsets;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Scanner;

public class Ejer\_1\_App {

public static void main(String[] args) {

String ruta\_lista="E:\\Material de java\\JAVA-UTN\\clase 7\\lista\_productos.txt";

leer\_Lista\_Productos(ruta\_lista);

}

public static void leer\_Lista\_Productos(String ruta\_lista){

Scanner ingreso\_datos1= new Scanner(System.in);

ArrayList<Ejer\_1\_Item\_Carrito> productos\_carrito= new ArrayList();

File archivo= new File(ruta\_lista);

try{

Scanner ingreso\_datos= new Scanner(archivo,StandardCharsets.UTF\_8);

int opcion=0;

do{

System.out.println("Bienvenido al carrito de compras, usted puede agregar los items de productos que necesite comprar\n"

+ "a continuacion puede cargarlos manualmente o desde un archivo .txt\n");

System.out.println("-Ingrese 1, si quiere cargar items de productos desde un archivo .txt\n"

+ "-Ingrese 2, si quiere cargar items de productos manualmente\n"

+"-Ingrese 3, si quiere ver los productos cargados en el carrito\n"

+ "-Ingrese 4, para salir de la carga de items al carrito y continuar con el proceso de pago del carrito");

opcion=ingreso\_datos1.nextInt();

switch(opcion){

case 1:

while(ingreso\_datos.hasNextLine()){

String[] partes\_linea= ingreso\_datos.nextLine().split(";");

System.out.println("mostrar datos extraidos de la lista de productos" + " " + Arrays.toString(partes\_linea));

Ejer\_1\_Item\_Carrito item\_carrito= new Ejer\_1\_Item\_Carrito();

item\_carrito.setStock(Integer.parseInt(partes\_linea[0]));

item\_carrito.setPrecio(Integer.parseInt(partes\_linea[1]));

item\_carrito.setNombre\_Producto(String.valueOf(partes\_linea[2]));

productos\_carrito.add(item\_carrito);

}

break;

case 2:

System.out.println("Ingrese la cantidad de items que quiere cargar en el carrito");

int cantidad\_item=ingreso\_datos1.nextInt();

int contador=1;

while(contador<=cantidad\_item){

Ejer\_1\_Item\_Carrito item\_carrito= new Ejer\_1\_Item\_Carrito();

System.out.println("Ingrese la cantidad de productos que quiere agregar en el item");

item\_carrito.setStock(ingreso\_datos1.nextInt());

System.out.println("Ingrese el precio para el item de producto");

item\_carrito.setPrecio(ingreso\_datos1.nextDouble());

System.out.println("Ingrese el nombre del producto");

item\_carrito.setNombre\_Producto(ingreso\_datos1.next());

productos\_carrito.add(item\_carrito);

contador++;

}

break;

case 3:

Ejer\_1\_Carrito carrito= new Ejer\_1\_Carrito();

carrito.item\_Cargado\_Carrito(productos\_carrito);

break;

}

}while(opcion >=1 && opcion <=3);

}catch(IOException ioe){

System.out.println("Lo siento, no se pudo encontrar/acceder al archivo" + ioe.getMessage());

ioe.printStackTrace();

}

System.out.println("a continuacion se procede a verificar si corresponde o no, aplicar un descuento sobre la compra total");

System.out.println("-----------------------------------------------------------------------------------------------------");

Ejer\_1\_Detalle\_Descuento detalle\_descuento= new Ejer\_1\_Detalle\_Descuento(1,40.0,1,"promo 1",0.10f,280.0f);

Descuento\_Porcentaje\_Con\_Tope desc\_t= new Descuento\_Porcentaje\_Con\_Tope();

Ejer\_1\_Carrito carrito= new Ejer\_1\_Carrito(productos\_carrito,detalle\_descuento,desc\_t);

double precio\_final=0.0;

precio\_final=carrito.obtener\_Precio\_Total(detalle\_descuento.verificar\_Aplica\_Descuento(productos\_carrito));

if(precio\_final <=0.0){

System.out.println("Lo siento, hay un error en el sistema o no hay ningun producto cargado con su valor\n"

+ "ya que, el total a pagar del carrito esta en $0 o negativo, por ende, no se puede realizar ninguna compra...");

}else if(precio\_final >=0 && precio\_final < 100){

System.out.println("A continuacion se muestra el valor total a pagar por los productos que se agregaron al carrito sin un descuento incluido\n"

+ "debido a que el total a pagar es menor a $100," + "el total a pagar es de:$" + precio\_final);

}else{

System.out.println("A continuacion se muestra el valor total a pagar por los productos que se agregaron al carrito con un descuento incluido\n"

+ "debido a que el total a pagar es mayor a $100," + "el total a pagar es de:$" + precio\_final);

}

}

}

Ejercicio 2. A y B

import java.io.File;

import java.io.IOException;

import java.nio.charset.StandardCharsets;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Scanner;

public class Ejer\_1\_App {

public static void main(String[] args) {

String ruta\_lista="E:\\Material de java\\JAVA-UTN\\clase 7\\lista\_productos.txt";

leer\_Lista\_Productos(ruta\_lista);

}

public static void leer\_Lista\_Productos(String ruta\_lista){

Scanner ingreso\_datos1= new Scanner(System.in);

ArrayList<Ejer\_1\_Item\_Carrito> productos\_carrito= new ArrayList();

File archivo= new File(ruta\_lista);

try{

Scanner ingreso\_datos= new Scanner(archivo,StandardCharsets.UTF\_8);

int opcion=0;

do{

System.out.println("Bienvenido al carrito de compras, usted puede agregar los items de productos que necesite comprar\n"

+ "a continuacion puede cargarlos manualmente o desde un archivo .txt\n");

System.out.println("-Ingrese 1, si quiere cargar items de productos desde un archivo .txt\n"

+ "-Ingrese 2, si quiere cargar items de productos manualmente\n"

+"-Ingrese 3, si quiere ver los productos cargados en el carrito\n"

+ "-Ingrese 4, para salir de la carga de items al carrito y continuar con el proceso de pago del carrito");

opcion=ingreso\_datos1.nextInt();

switch(opcion){

case 1:

while(ingreso\_datos.hasNextLine()){

String[] partes\_linea= ingreso\_datos.nextLine().split(";");

System.out.println("mostrar datos extraidos de la lista de productos" + " " + Arrays.toString(partes\_linea));

Ejer\_1\_Item\_Carrito item\_carrito= new Ejer\_1\_Item\_Carrito();

item\_carrito.setStock(Integer.parseInt(partes\_linea[0]));

item\_carrito.setPrecio(Integer.parseInt(partes\_linea[1]));

item\_carrito.setNombre\_Producto(String.valueOf(partes\_linea[2]));

productos\_carrito.add(item\_carrito);

}

break;

case 2:

System.out.println("Ingrese la cantidad de items que quiere cargar en el carrito");

int cantidad\_item=ingreso\_datos1.nextInt();

int contador=1;

while(contador<=cantidad\_item){

Ejer\_1\_Item\_Carrito item\_carrito= new Ejer\_1\_Item\_Carrito();

System.out.println("Ingrese la cantidad de productos que quiere agregar en el item");

item\_carrito.setStock(ingreso\_datos1.nextInt());

System.out.println("Ingrese el precio para el item de producto");

item\_carrito.setPrecio(ingreso\_datos1.nextDouble());

System.out.println("Ingrese el nombre del producto");

item\_carrito.setNombre\_Producto(ingreso\_datos1.next());

productos\_carrito.add(item\_carrito);

contador++;

}

break;

case 3:

Ejer\_1\_Carrito carrito= new Ejer\_1\_Carrito();

carrito.item\_Cargado\_Carrito(productos\_carrito);

break;

}

}while(opcion >=1 && opcion <=3);

}catch(IOException ioe){

System.out.println("Lo siento, no se pudo encontrar/acceder al archivo" + ioe.getMessage());

ioe.printStackTrace();

}

System.out.println("a continuacion se procede a verificar si corresponde o no, aplicar un descuento sobre la compra total");

System.out.println("-----------------------------------------------------------------------------------------------------");

Ejer\_1\_Detalle\_Descuento detalle\_descuento= new Ejer\_1\_Detalle\_Descuento(1,40.0,1,"promo 1",0.10f,280.0f);

Descuento\_Porcentaje\_Con\_Tope desc\_t= new Descuento\_Porcentaje\_Con\_Tope();

Ejer\_1\_Carrito carrito= new Ejer\_1\_Carrito(productos\_carrito,detalle\_descuento,desc\_t);

double precio\_final=0.0;

precio\_final=carrito.obtener\_Precio\_Total(detalle\_descuento.verificar\_Aplica\_Descuento(productos\_carrito));

if(precio\_final <=0.0){

System.out.println("Lo siento, hay un error en el sistema o no hay ningun producto cargado con su valor\n"

+ "ya que, el total a pagar del carrito esta en $0 o negativo, por ende, no se puede realizar ninguna compra...");

}else if(precio\_final >=0 && precio\_final < 100){

System.out.println("A continuacion se muestra el valor total a pagar por los productos que se agregaron al carrito sin un descuento incluido\n"

+ "debido a que el total a pagar es menor a $100," + "el total a pagar es de:$" + precio\_final);

}else{

System.out.println("A continuacion se muestra el valor total a pagar por los productos que se agregaron al carrito con un descuento incluido\n"

+ "debido a que el total a pagar es mayor a $100," + "el total a pagar es de:$" + precio\_final);

}

}

}

public class Ejer\_1\_Carrito {

private int id\_nro\_carrito;

private double subtotal;

private double precio\_envio;

private double pago\_total;

private Ejer\_1\_Detalle\_Descuento descuento\_carrito;

private Descuento\_Porcentaje\_Con\_Tope desc\_tope;

private ArrayList<Ejer\_1\_Item\_Carrito>Productos;

public Ejer\_1\_Carrito(ArrayList<Ejer\_1\_Item\_Carrito> productos, Ejer\_1\_Detalle\_Descuento desc\_carrito, Descuento\_Porcentaje\_Con\_Tope desc\_tope){

this.Productos=productos;

this.descuento\_carrito=desc\_carrito;

this.desc\_tope=desc\_tope;

}

public Ejer\_1\_Carrito(){};

public void item\_Cargado\_Carrito(ArrayList<Ejer\_1\_Item\_Carrito>productos\_carrito){

for(Ejer\_1\_Item\_Carrito items: productos\_carrito){

System.out.println("el item cargado es:" + " " + items.getNombre\_Producto() + " " + "con un precio de: $" + items.getPrecio() + "y con una cantidad de:" + items.getStock());

}

System.out.println("la cantidad total de items que hay en el carrito son:" + "" + productos\_carrito.size());

}

public double obtener\_Precio\_Total(boolean aplica\_descuento) {

Scanner ingreso\_datos= new Scanner(System.in);

pago\_total=0;

int opcion=0;

float descuento\_parcial=0.0f;

for(Ejer\_1\_Item\_Carrito item\_carrito :Productos){

pago\_total += (item\_carrito.getStock()\*item\_carrito.getPrecio());

}

try{

if(pago\_total>0.0){

if((aplica\_descuento) && (pago\_total>=250.0)){

do{

System.out.println("Su compra obtiene un descuento(fijo,10% y 5%) por parte del comercio\n"

+ ",los descuentos no son acumulables, a continuación debe elegir el decuento que quiere que se le aplique al carrito....");

System.out.println("Ingrese la opcion 1, para el descuento fijo\n"

+ "Ingrese la opcion 2, para el descuento del 10%\n"

+ "Ingrese la opcion 3, para el descuento del 5%");

opcion=ingreso\_datos.nextInt();

switch(opcion){

case 1:

if(pago\_total>250){

pago\_total= pago\_total - descuento\_carrito.getDescuento\_fijo();

if(pago\_total<0){

throw new RuntimeException("....no se puede aplicar el descuento, ya que, el total a pagar no puede ser negativo");

}else{

System.out.println("Se aplica un descuento fijo de $:" + " " + descuento\_carrito.getDescuento\_fijo() + " "

+ " debido a que el monto total a pagar supera los $250");

}

return pago\_total;

}else{

System.out.println("lo siento, para aplicar este descuento, el total de compra en su carrito debe ser de $250 en adelante\n"

+ " mientras que la compra que desea realizar llega al monto total de:" + "$"+ pago\_total);

}

break;

case 2:

if(pago\_total<250 && pago\_total > 200){

descuento\_parcial= (float) (pago\_total\* descuento\_carrito.getPorcentaje\_descuento());

pago\_total= pago\_total- descuento\_parcial;

System.out.println("Se aplica un descuento del 10%, el cual equivale a $:" + " " + descuento\_parcial + " "

+ " debido a que el monto total a pagar no supera los $250");

return pago\_total;

}else{

System.out.println("lo siento, para aplicar este descuento, el total de compra en su carrito debe ser entre $200 y $250\n"

+ " mientras que la compra que desea realizar llega al monto total de:" + "$"+ pago\_total);

}

break;

case 3:

if(pago\_total<=200 && pago\_total >=100){

descuento\_parcial= (float) (pago\_total \* desc\_tope.getObtener\_Desc\_Fijo());

pago\_total=pago\_total-descuento\_parcial;

System.out.println("Se aplica un descuento del 5%, el cual equivale a $:" + " " + descuento\_parcial + " "

+ " debido a que el monto total a pagar se encuentra entre $200 y $100");

return pago\_total;

}

break;

}

}while(opcion > 0 && opcion <=2 );

}

}else{

System.out.println("Lo siento, tu compra no aplica ningun tipo de descuento");

throw new ArithmeticException("....no corresponde aplicar descuentos a un carrito en $0");

}

}catch(ArithmeticException ae){

System.out.println("lo siento, el valor total a pagar en el carrito de compras es $0, por lo tanto no aplica ningun descuento" + " " + ae.getMessage());

}

catch(RuntimeException re){

System.out.println("lo siento, el valor total a pagar en el carrito de compras no puede quedar en negativo" + " " + re.getMessage());

}

return pago\_total;

}

}

import java.util.ArrayList;

public abstract class Ejer\_1\_Descuento {

private int id\_nro\_descuento;

private double descuento\_total;

public Ejer\_1\_Descuento(int id, double descuento){

this.id\_nro\_descuento=id;

this.descuento\_total=descuento;

}

public Ejer\_1\_Descuento(){}

public abstract boolean verificar\_Aplica\_Descuento(ArrayList<Ejer\_1\_Item\_Carrito> item\_productos);

}

import java.util.ArrayList;

public class Descuento\_Porcentaje\_Con\_Tope extends Ejer\_1\_Descuento {

private static float descuento\_porcentaje\_fijo= 0.05f;

public static float getObtener\_Desc\_Fijo(){

return descuento\_porcentaje\_fijo;

}

public Descuento\_Porcentaje\_Con\_Tope(){};

@Override

public boolean verificar\_Aplica\_Descuento(ArrayList<Ejer\_1\_Item\_Carrito> item\_productos) {

return false;

}

}

import java.util.ArrayList;

/\*\*

\*

\* @author diego vargas

\*/

public class Ejer\_1\_Detalle\_Descuento extends Ejer\_1\_Descuento{

private int id\_detalle\_descuento;

private String nombre\_detalle\_descuento;

private float porcentaje\_descuento;

private float descuento\_fijo;

public Ejer\_1\_Detalle\_Descuento(int id\_nro\_descuento,double descuento\_total, int id\_deta\_desc, String nomb\_desc, float porcentaje\_desc, float desc\_fijo){

super(id\_nro\_descuento,descuento\_total);

this.id\_detalle\_descuento=id\_deta\_desc;

this.nombre\_detalle\_descuento=nomb\_desc;

this.porcentaje\_descuento=porcentaje\_desc;

this.descuento\_fijo=desc\_fijo;

}

public Ejer\_1\_Detalle\_Descuento(){}

public float getPorcentaje\_descuento() {

return porcentaje\_descuento;

}

public float getDescuento\_fijo() {

return descuento\_fijo;

}

@Override

public boolean verificar\_Aplica\_Descuento(ArrayList<Ejer\_1\_Item\_Carrito> item\_productos) {

int cantidad\_productos=0;

for(Ejer\_1\_Item\_Carrito item: item\_productos){

cantidad\_productos += item.getStock();

}

if(cantidad\_productos>3){

return true;

}

return false;

}

}

public class Ejer\_1\_Item\_Carrito extends Ejer\_1\_Producto{

private int id\_item\_carrito;

private String descripcion\_producto;

private int cantidad\_item;

private double precio\_item;

private int cantidad\_x\_item;

private double total\_x\_item;

public Ejer\_1\_Item\_Carrito(int id,int stock,double precio,String nombre\_producto,int id\_item,String descrip, int cantidad,double precio\_item, int cant\_prod, double total){

super(id,stock,precio,nombre\_producto);

this.id\_item\_carrito=id\_item;

this.descripcion\_producto=descrip;

this.cantidad\_item=cantidad;

this.precio\_item=precio\_item;

this.cantidad\_x\_item=cant\_prod;

this.total\_x\_item=total;

}

public Ejer\_1\_Item\_Carrito(){}

public void setId\_item\_carrito(int id\_item\_carrito) {

this.id\_item\_carrito = id\_item\_carrito;

}

public void setCantidad\_X\_Item(int cantidad\_productos) {

this.cantidad\_x\_item = cantidad\_productos;

}

public int getId\_item\_carrito() {

return id\_item\_carrito;

}

public int getCantidad\_X\_Item() {

return cantidad\_x\_item;

}

@Override

public String toString() {

return "Ejer\_1\_Item\_Carrito{" + "stock\_producto= " + " " + super.getStock()+ " " + "id\_item\_carrito=" + id\_item\_carrito + ", descripcion\_producto=" + descripcion\_producto + ", cantidad\_item=" + cantidad\_item + ", precio\_item=" + precio\_item + ", cantidad\_x\_item=" + cantidad\_x\_item + ", total\_x\_item=" + total\_x\_item + '}';

}

public int cantidad\_Total\_X\_Item(){

if(cantidad\_x\_item<=0){

System.out.println("lo siento, no hay productos cargados");

return 0;

}

else{

System.out.println("la cantidad total de productos cargados son:" + " " + cantidad\_x\_item);

return cantidad\_x\_item;

}

}

@Override

public int consultar\_Stock(){

return 0;

}

}

public class Ejer\_1\_Producto {

private int id\_producto;

private int stock;

private double precio;

private String nombre\_producto;

public Ejer\_1\_Producto(int id, int cant, double precio,String desc\_prod){

this.id\_producto=id;

this.stock=cant;

this.precio=precio;

this.nombre\_producto=desc\_prod;

}

public Ejer\_1\_Producto(){};

public int getId\_Producto() {

return id\_producto;

}

public int getStock() {

return stock;

}

public double getPrecio() {

return precio;

}

public String getNombre\_Producto() {

return nombre\_producto;

}

public void setId\_Producto(int id\_producto) {

this.id\_producto = id\_producto;

}

public void setStock(int stock) {

this.stock = stock;

}

public void setPrecio(double precio) {

this.precio = precio;

}

public void setNombre\_Producto(String nombre\_producto) {

this.nombre\_producto = nombre\_producto;

}

public int consultar\_Stock(){

return 0;

}

}

public class Ejer\_1\_Tipo\_Producto extends Ejer\_1\_Producto{

private int id\_tipo\_producto;

private String marca;

private String descripcion\_tipo\_producto;

public Ejer\_1\_Tipo\_Producto(int id\_producto,int stock,double precio,String nombre\_producto,int id\_tipo\_prod,String marca,String descripcion){

super(id\_producto,stock,precio,nombre\_producto);

this.id\_tipo\_producto=id\_tipo\_prod;

this.marca=marca;

this.descripcion\_tipo\_producto=descripcion;

}

public Ejer\_1\_Tipo\_Producto(){};

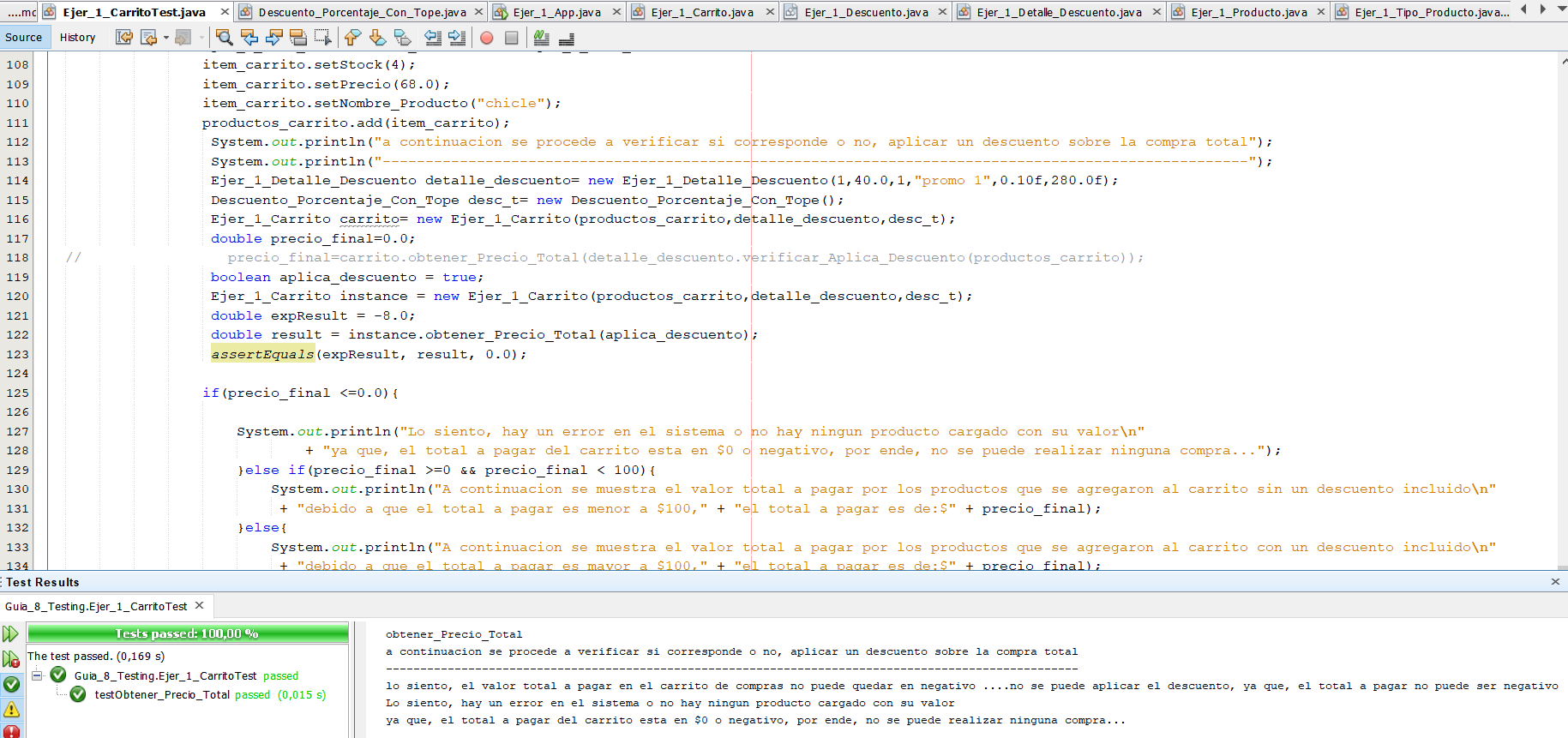
@Override

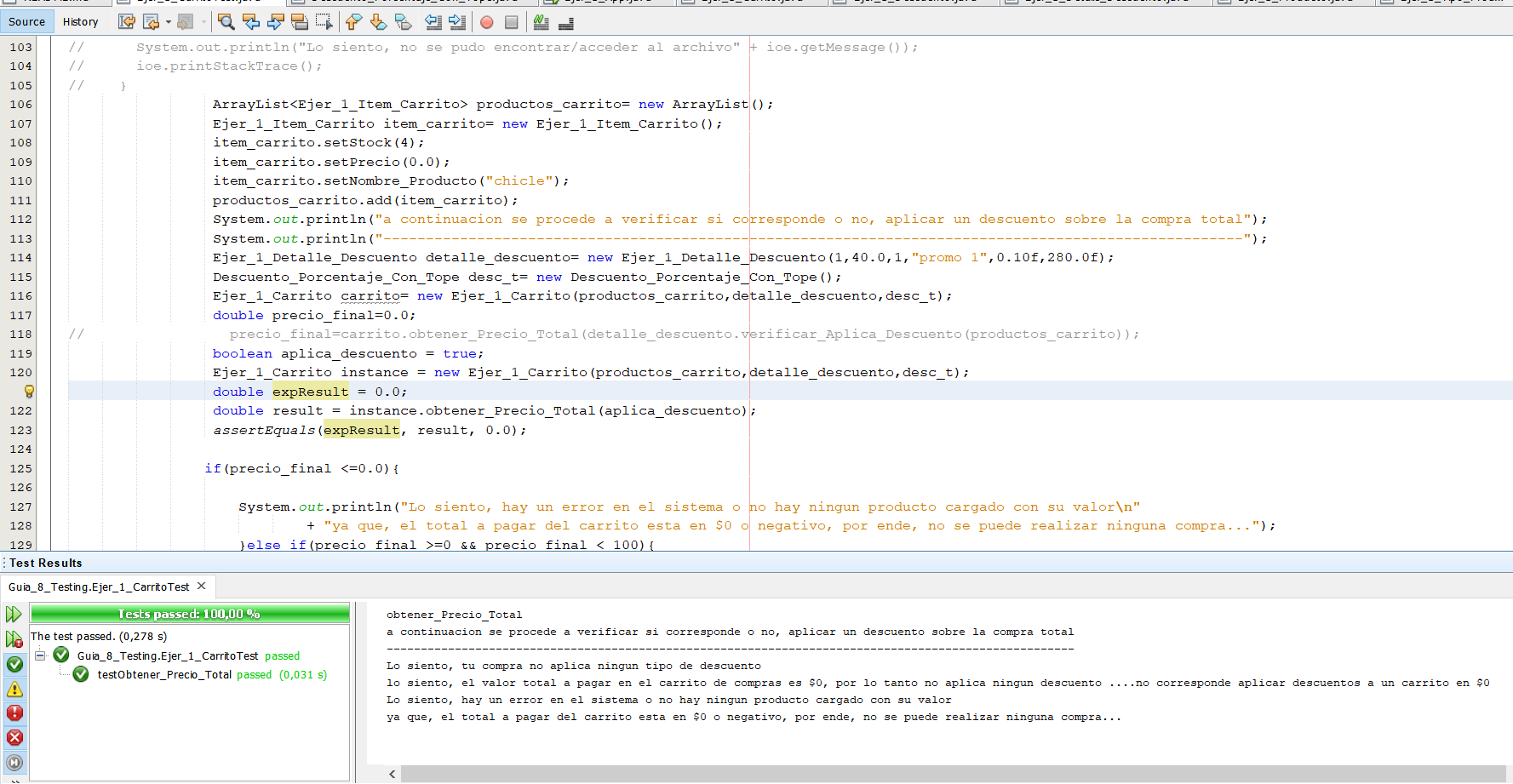
public int consultar\_Stock(){

return 0;

}

}

Ejercicio 2. C



En ambos casos se aplican las excepciones planteadas en el código, y para un caso no se puede aplicar el descuento, ya que, el carrito se encuentra en $0, para el otro caso, no se puede aplicar el descuento, debido a que, el descuento a aplicar es mayor al valor del pago total del carrito y no puede quedar un valor negativo.

Ejercicio 3.

package Guia\_8\_Testing;

import java.util.ArrayList;

import org.junit.Test;

import static org.junit.Assert.\*;

/\*\*

\*

\* @author diego vargas

\*/

public class Ejer\_1\_CarritoTest {

public Ejer\_1\_CarritoTest() {

}

@Test

public void testObtener\_Precio\_Total() {

System.out.println("obtener\_Precio\_Total");

ArrayList<Ejer\_1\_Item\_Carrito> productos\_carrito= new ArrayList();

Ejer\_1\_Item\_Carrito item\_carrito= new Ejer\_1\_Item\_Carrito();

item\_carrito.setStock(4);

item\_carrito.setPrecio(68.0);

item\_carrito.setNombre\_Producto("chicle");

productos\_carrito.add(item\_carrito);

System.out.println("a continuacion se procede a verificar si corresponde o no, aplicar un descuento sobre la compra total");

System.out.println("-----------------------------------------------------------------------------------------------------");

Ejer\_1\_Detalle\_Descuento detalle\_descuento= new Ejer\_1\_Detalle\_Descuento(1,40.0,1,"promo 1",0.10f,280.0f);

Descuento\_Porcentaje\_Con\_Tope desc\_t= new Descuento\_Porcentaje\_Con\_Tope();

Ejer\_1\_Carrito carrito= new Ejer\_1\_Carrito(productos\_carrito,detalle\_descuento,desc\_t);

double precio\_final=0.0;

precio\_final=carrito.obtener\_Precio\_Total(detalle\_descuento.verificar\_Aplica\_Descuento(productos\_carrito));

boolean aplica\_descuento = true;

Ejer\_1\_Carrito instance = new Ejer\_1\_Carrito(productos\_carrito,detalle\_descuento,desc\_t);

double expResult = -8.0;

double result = instance.obtener\_Precio\_Total(aplica\_descuento);

assertEquals(expResult, result, 0.0);

if(precio\_final <=0.0){

System.out.println("Lo siento, hay un error en el sistema o no hay ningun producto cargado con su valor\n"

+ "ya que, el total a pagar del carrito esta en $0 o negativo, por ende, no se puede realizar ninguna compra...");

}else if(precio\_final >=0 && precio\_final < 100){

System.out.println("A continuacion se muestra el valor total a pagar por los productos que se agregaron al carrito sin un descuento incluido\n"

+ "debido a que el total a pagar es menor a $100," + "el total a pagar es de:$" + precio\_final);

}else{

System.out.println("A continuacion se muestra el valor total a pagar por los productos que se agregaron al carrito con un descuento incluido\n"

+ "debido a que el total a pagar es mayor a $100," + "el total a pagar es de:$" + precio\_final);

}

// fail("The test case is a prototype.");

}

}

public double obtener\_Precio\_Total(boolean aplica\_descuento) throws ArithmeticException{

Scanner ingreso\_datos= new Scanner(System.in);

pago\_total=0;

int opcion=0;

float descuento\_parcial=0.0f;

for(Ejer\_1\_Item\_Carrito item\_carrito :Productos){

pago\_total += (item\_carrito.getStock()\*item\_carrito.getPrecio());

}

try{

if(pago\_total>0.0){

if((aplica\_descuento) && (pago\_total>=250.0)){

if(pago\_total>250){

pago\_total= pago\_total - descuento\_carrito.getDescuento\_fijo();

if(pago\_total<0){

throw new RuntimeException("....no se puede aplicar el descuento, ya que, el total a pagar no puede ser negativo");

}else{

System.out.println("Se aplica un descuento fijo de $:" + " " + descuento\_carrito.getDescuento\_fijo() + " "

+ " debido a que el monto total a pagar supera los $250");

}

return pago\_total;

}else{

System.out.println("lo siento, para aplicar este descuento, el total de compra en su carrito debe ser de $250 en adelante\n"

+ " mientras que la compra que desea realizar llega al monto total de:" + "$"+ pago\_total);

}

}

}else{

System.out.println("Lo siento, tu compra no aplica ningun tipo de descuento");

throw new ArithmeticException("....no corresponde aplicar descuentos a un carrito en $0");

}

}catch(ArithmeticException ae){

System.out.println("lo siento, el valor total a pagar en el carrito de compras es $0, por lo tanto no aplica ningun descuento" + " " + ae.getMessage());

}

catch(RuntimeException re){

System.out.println("lo siento, el valor total a pagar en el carrito de compras no puede quedar en negativo" + " " + re.getMessage());

}

return pago\_total;

}

}

import java.util.ArrayList;

import org.junit.Test;

import static org.junit.Assert.\*;

public class Ejer\_1\_CarritoTest {

public Ejer\_1\_CarritoTest() {

}

@Test

public void testObtener\_Precio\_Total() {

System.out.println("obtener\_Precio\_Total");

ArrayList<Ejer\_1\_Item\_Carrito> productos\_carrito= new ArrayList();

Ejer\_1\_Item\_Carrito item\_carrito= new Ejer\_1\_Item\_Carrito();

item\_carrito.setStock(4);

item\_carrito.setPrecio(0.0);

item\_carrito.setNombre\_Producto("chicle");

productos\_carrito.add(item\_carrito);

System.out.println("a continuacion se procede a verificar si corresponde o no, aplicar un descuento sobre la compra total");

System.out.println("-----------------------------------------------------------------------------------------------------");

Ejer\_1\_Detalle\_Descuento detalle\_descuento= new Ejer\_1\_Detalle\_Descuento(1,40.0,1,"promo 1",0.10f,40.0f);

Descuento\_Porcentaje\_Con\_Tope desc\_t= new Descuento\_Porcentaje\_Con\_Tope();

Ejer\_1\_Carrito carrito= new Ejer\_1\_Carrito(productos\_carrito,detalle\_descuento,desc\_t);

double precio\_final=0.0;

// precio\_final=carrito.obtener\_Precio\_Total(detalle\_descuento.verificar\_Aplica\_Descuento(productos\_carrito));

boolean aplica\_descuento = true;

Ejer\_1\_Carrito instance = new Ejer\_1\_Carrito(productos\_carrito,detalle\_descuento,desc\_t);

double expResult = 0.0;

double result = instance.obtener\_Precio\_Total(aplica\_descuento);

assertEquals(expResult, result, 0.0);

if(precio\_final <=0.0){

System.out.println("Lo siento, hay un error en el sistema o no hay ningun producto cargado con su valor\n"

+ "ya que, el total a pagar del carrito esta en $0 o negativo, por ende, no se puede realizar ninguna compra...");

}else if(precio\_final >=0 && precio\_final < 100){

System.out.println("A continuacion se muestra el valor total a pagar por los productos que se agregaron al carrito sin un descuento incluido\n"

+ "debido a que el total a pagar es menor a $100," + "el total a pagar es de:$" + precio\_final);

}else{

System.out.println("A continuacion se muestra el valor total a pagar por los productos que se agregaron al carrito con un descuento incluido\n"

+ "debido a que el total a pagar es mayor a $100," + "el total a pagar es de:$" + precio\_final);

}

}

}

**Enlace GitHub:**