TAREA # 4

Nombre: Juan David Jiménez Romero

Clase main:

**package** lab2;

**import** java.util.Scanner;

**public** **class** Main {

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

// **TODO** Auto-generated method stub

Restaurante restaurante = **new** Restaurante();

plato plato = **new** plato();

String a;

**double** b;

**int** opc,c;

Scanner scan = **new** Scanner(System.***in***);

//Menu de opciones que aparece en pantalla al iniciar el programa

System.***out***.println("===Menu de opciones===");

System.***out***.println("1. Agregar Pedido a la Cocina");

System.***out***.println("2. Servir Pedido en el Comedor");

System.***out***.println("3. Mostrar Pedidos en el Comedor");

System.***out***.println("4. Eliminar Pedido de la Cocina");

System.***out***.println("5.Salir");

opc = scan.nextInt();

**while**(opc!=5) {

**switch**(opc) {

//caso 1 donde se agrega el pedido a la cocina

**case** 1:

System.***out***.println("Ingrese los datos del que pidio el producto");

System.***out***.println("Numero del pedido: ");

c = scan.nextInt();

System.***out***.println("Ingrese el nombre del cliente: ");

a = scan.next();

Pedido pedido = **new** Pedido(c, a);

System.***out***.println("Ingrese los datos del producto");

System.***out***.println("Menu Actual: ");

System.***out***.println("--Encebollado-$2,00");

System.***out***.println("---Chaulafan--$2,50");

System.***out***.println("Ingrese el nobre del producto: ");

a = scan.next();

System.***out***.println("Ingrese el valor del platillo: ");

b = scan.nextDouble();

System.***out***.println("Ingrese la cantidad que desea del platillo: ");

c = scan.nextInt();

pedido.agregarProducto(**new** Producto(a, b, c));

**for**(**int** i= 0; i< c;i++) {

**if** (a.equals("Encebollado")) {

**try** {

// Intenta detener el programa actual durante 3 segundos (3000 milisegundos)

Thread.*sleep*(3000);

System.***out***.println("Ingresar");

} **catch** (InterruptedException e) {

// Maneja la excepción si ocurre algún problema durante la pausa del hilo

e.printStackTrace();

}

}**else** **if**(a.equals("Chaulafan")) {

**try** {

// Intenta detener el programa actual durante 3 segundos (3000 milisegundos)

Thread.*sleep*(3000);

} **catch** (InterruptedException e) {

// Maneja la excepción si ocurre algún problema durante la pausa del hilo

e.printStackTrace();

}

}

}

restaurante.agregarPedido(pedido);

**break**;

//case 2 donde se enrega el plato al comedor

**case** 2:

restaurante.entregarPedidoPreparado();

restaurante.prepararPedidoActual();

System.***out***.println("Se entregaron los productos exitosamente");

**break**;

//caso 3 donde se muestra la lista de los pedidos

**case** 3:

System.***out***.println("Esta es la lista de los Pedidos: ");

restaurante.mostrarPedidos();

**break**;

**case** 4:

//caso donde se elimina un pedido especifico de los pedidos

plato.eliminarPedidoCocina();

System.***out***.println("Se elimino el pedido correctamente");

**break**;

**default**:

//aqui es donde se muestra un mensaje en caso de no elejir una opcion valida

System.***out***.println("Opcion no valida");

**break**;

}

//aqui se vuelve a mostrar el menu de opciones en cuando se acabe de realizar una funcion

System.***out***.println("===Menu de opciones===");

System.***out***.println("1. Agregar Pedido a la Cocina");

System.***out***.println("2. Servir Pedido en el Comedor");

System.***out***.println("3. Mostrar Pedidos en el Comedor");

System.***out***.println("4. Eliminar Pedido de la Cocina");

System.***out***.println("5.Salir");

opc = scan.nextInt();

}

//mensaje que sale al pulsar 5 (opcion para salir)

System.***out***.println("Gracias por ingresar");

}

}

Clase Producto:

**package** lab2;

**import** java.util.Queue;

**public** **class** Producto {

//atributos de la clase producto

String nombre;

**double** precio;

**int** cantidad;

//Constructor que inicializa los valores de nombre, precio, cantidad

**public** Producto(String nombre, **double** precio, **int** cantidad) {

**this**.nombre = nombre;

**this**.precio = precio;

**this**.cantidad = cantidad;

}

}

Clase pedido:

**package** lab2;

**import** java.util.LinkedList;

**import** java.util.Queue;

**public** **class** Pedido {

//atributos de la clase pedido

**int** numeroPedido;

String nombreCliente;

Queue<Producto> productos = **new** LinkedList<>();

//constructor de la clase pedidos donde se inicializa los valores de numeroPedido y nombreCliente

**public** Pedido(**int** numeroPedido, String nombreCliente) {

**this**.numeroPedido = numeroPedido;

**this**.nombreCliente = nombreCliente;

}

//Metodo que sirve para agregar los productos

**public** **void** agregarProducto(Producto producto) {

productos.add(producto);

}

**public** **void** Tiempoespera(String a, **int** c){

}

}

Clase Restaurante:

**package** lab2;

**import** java.util.Queue;

**import** java.util.Stack;

**import** java.util.LinkedList;

**import** java.util.Scanner;

**public** **class** Restaurante {

//creacion de la lista pedidosPendientes y la cola pedidosPreparados

Scanner scan = **new** Scanner(System.***in***);

**int** a;

Stack<Pedido> pedidosPendientes = **new** Stack<>();

Queue<Pedido> pedidosPreparados = **new** LinkedList<>();

//metodo agregarPedido que como dice su nombre agrega el pedido a la lista

**public** **void** agregarPedido(Pedido pedido) {

pedidosPendientes.push(pedido);

}

//metodo prepararPedidoActual donde el valor que esta en pedidosPendientes se guarda en un pedido y lo elimina de la lista pedidoPendientes

//y se añade a la cola pedidosPreparados el pedido que se elimino de pedidosPendientes

**public** **void** prepararPedidoActual() {

Pedido pedido = pedidosPendientes.pop();

pedidosPreparados.add(pedido);

}

//Metodo que borra de la cola el primer pedidoPreparado

**public** **void** entregarPedidoPreparado() {

pedidosPreparados.poll();

}

//metodo que muestra los pedidos

**public** **void** mostrarPedidos() {

**for**(**int** i=0;i<pedidosPreparados.size();i++)

pedidosPreparados.poll();

}

}

Clase plato:

**package** lab2;

**public** **class** plato {

//se declaran los atributos de la clase plato

String nombre;

String tipo;

**double** precio;

Pedido pedido= **new** Pedido(0,**null**);

//metodo que elimina un pedido de la cocina

**public** **void** eliminarPedidoCocina() {

pedido.productos.poll();

}

//metodo que muestra los platos que ya se estan hechos

**public** **void** mostrarPlatos() {

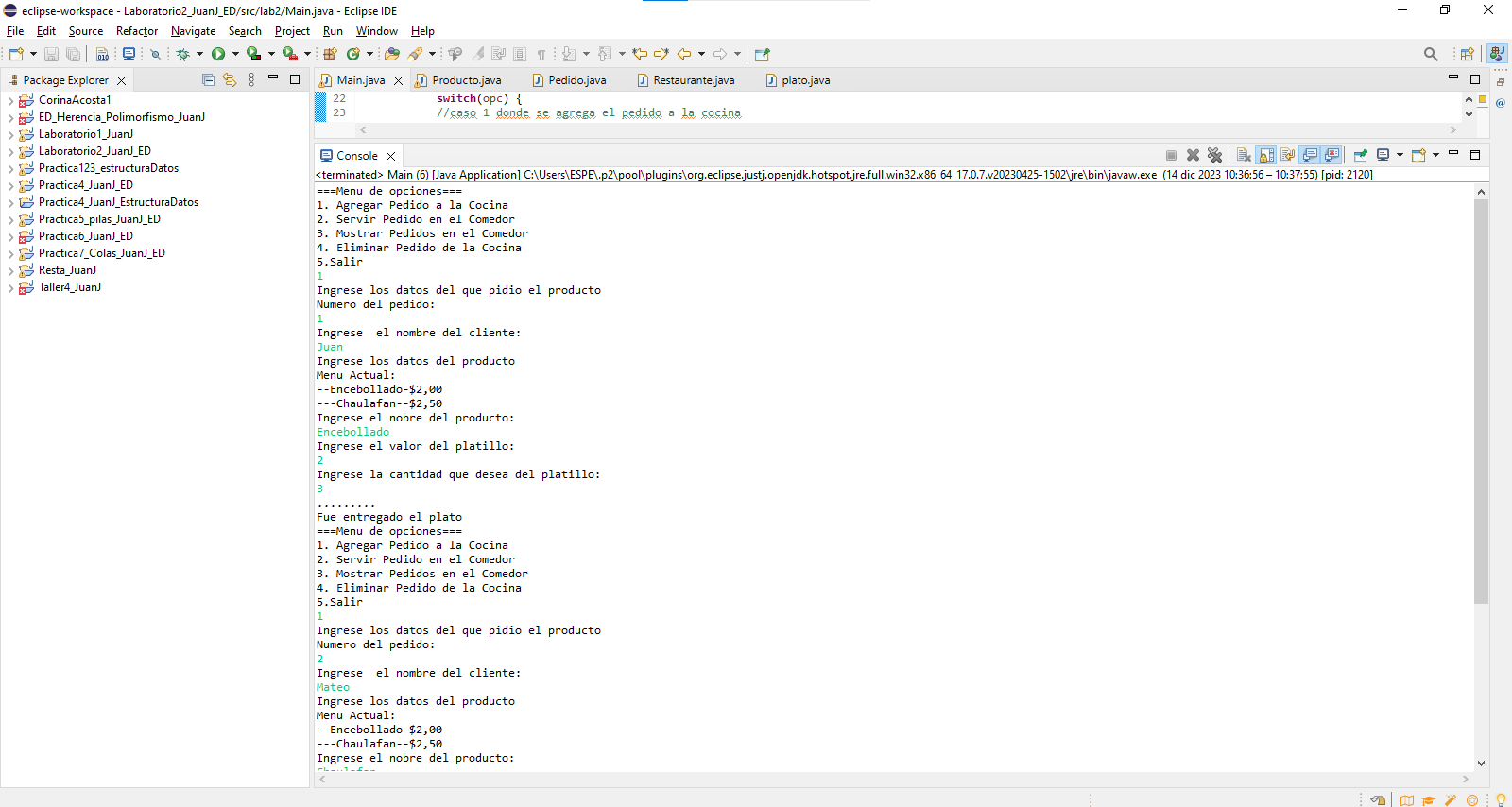
**while**(!(pedido.productos.peek()==**null**)) {

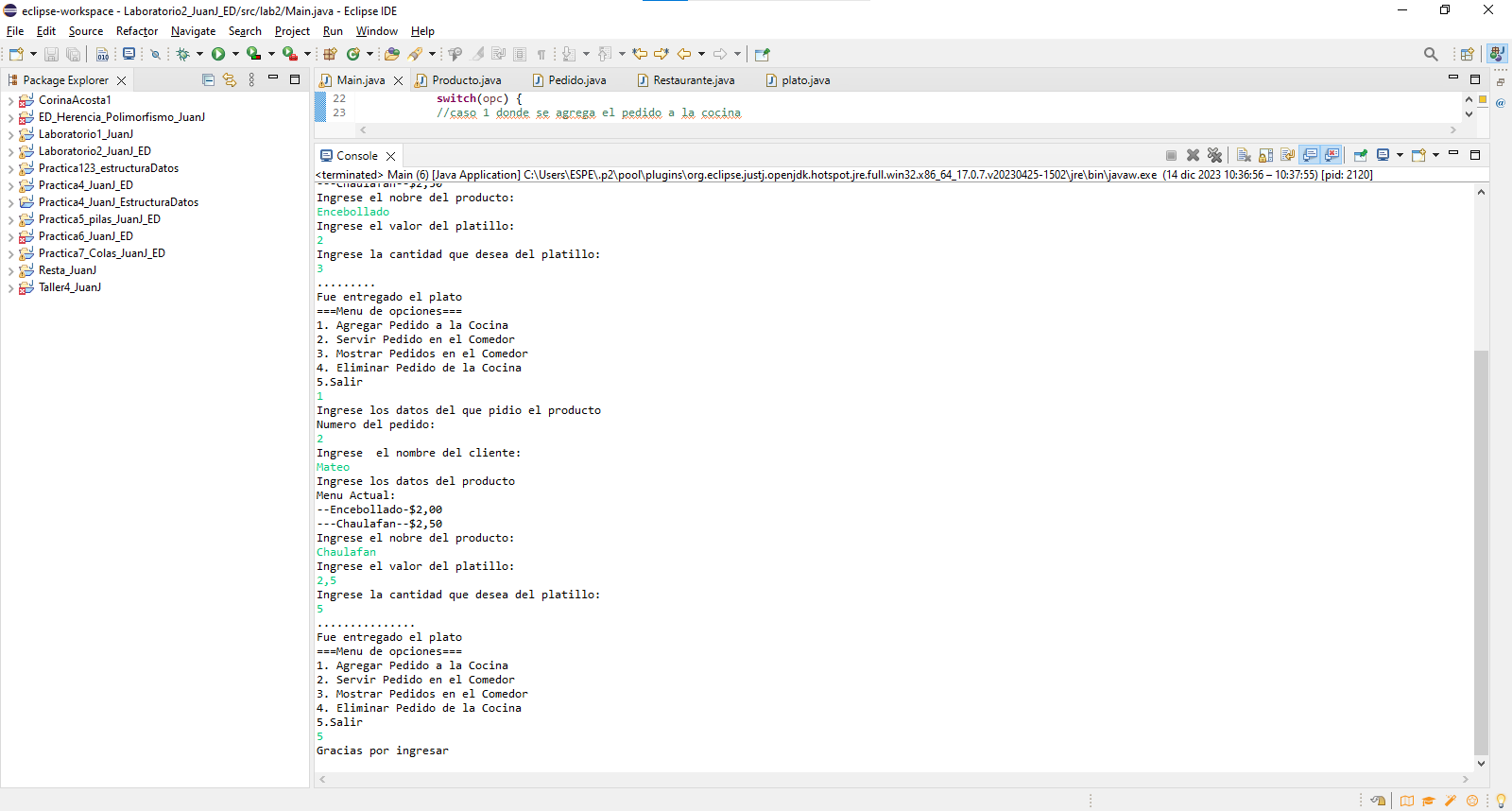
pedido.productos.poll();

}

}

}

Ejecucion:  
  


  
En la actividad actual se realizó que cuando el plato este en realización se demore cierto tiempo dependiendo el plato que sea el cual podíamos usar el método ThreadSleep para hacer que esto ocurra, por ende hice que el programa al momento de entrar a la cocina este se tarde cierto tiempo para dependiendo el nombre del plato para que este se tarde cierto tiempo por lo cual si se ingresa un plato del menú especificado y además hice que dependiendo la cantidad de platos que haya pedido este se tarde más tiempo respectivamente al tiempo que se tarde en hacer 1 \* la cantidad de platos que desea.