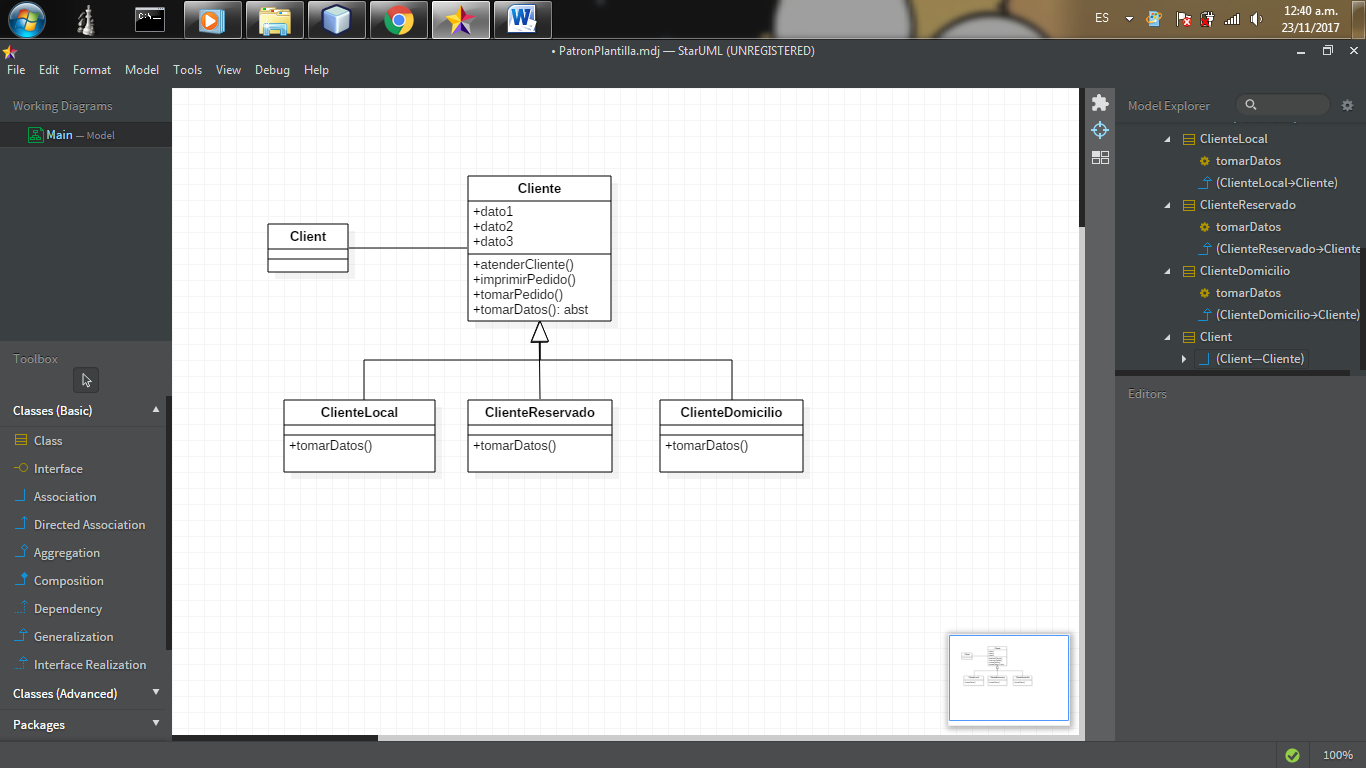
**ESCENARIO:**

Planteamos un escenario que sucedería dentro de un restaurante, la situación problema se presenta para el manejo de los datos en los diferentes tipos de clientes, sin desligarlos de la idea de que todos son clientes y deben ser atendidos.



public class PlantillaCliente {

public static void main(String[] args) {

Cliente cliente;

String opcion;

boolean salir = false;

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

System.out.println("BIENVENIDO AL RESTAURANTE");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

while (salir == false) {

opcion = JOptionPane.showInputDialog(null, "...REGISTRO DE CLIENTES..."

+ "\n(1)Cliente Local "

+ "\n(2)Cliente Domicilio "

+ "\n(3)Cliente Reserva "

+ "\n(4)Salir");

try {

switch (opcion) {

case "1":

cliente = new ClienteLocal();

cliente.atenderCliente();

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

break;

case "2":

cliente = new ClienteDomicilio();

cliente.atenderCliente();

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

break;

case "3":

cliente = new ClienteReservado();

cliente.atenderCliente();

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

break;

case "4":

salir = true;

break;

default:

System.out.println("Opcion NO VALIDA");

break;

}

} catch (NumberFormatException ex) {

System.out.println("No Es Un Valor Valido");

}

}

}

}

public abstract class Cliente {

private Vector<String> platillos = new Vector<String>();

private int articulos = 0;

public String dato1;

public String dato2;

public String dato3;

protected abstract void tomarDatos();

public void atenderCliente() {

tomarDatos();

tomarPedido();

imprimirPedido();

}

protected void tomarPedido() {

String entrada;

boolean terminado = false;

String opcion;

while (terminado == false) {

try {

opcion = JOptionPane.showInputDialog(null, "(1)Pedir Producto "

+ "\n(2)Pedido Terminado");

switch (opcion) {

case "1":

entrada = JOptionPane.showInputDialog(null, "Digite Su Pedido: ");

platillos.addElement(entrada);

articulos++;

break;

case "2":

terminado = true;

break;

}

} catch (NumberFormatException ex) {

System.out.println("No Es Un Valor Valido");

}

}

}

protected void imprimirDatos() {

System.out.println(dato1);

System.out.println(dato2);

System.out.println(dato3);

}

protected void imprimirPedido() {

for (int i = 0; i < articulos; i++) {

System.out.println((1 + i) + ": " + platillos.get(i));

}

}

}

public class ClienteLocal extends Cliente {

@Override

protected void tomarDatos() {

this.dato1 = JOptionPane.showInputDialog(null, "Nombre: ");

this.dato2 = JOptionPane.showInputDialog(null, "Mesa: ");

this.dato3 = JOptionPane.showInputDialog(null, "Mesero: ");

}

}

public class ClienteDomicilio extends Cliente {

@Override

protected void tomarDatos() {

this.dato1 = JOptionPane.showInputDialog(null, "Nombre: ");

this.dato2 = JOptionPane.showInputDialog(null, "Direccion: ");

this.dato3 = JOptionPane.showInputDialog(null, "Telefono: ");

}

}

public class ClienteReservado extends Cliente {

@Override

protected void tomarDatos() {

this.dato1 = JOptionPane.showInputDialog(null, "Nombre: ");

this.dato2 = JOptionPane.showInputDialog(null, "Fecha Reserva: ");

this.dato3 = JOptionPane.showInputDialog(null, "Tipo Reserva: ");

}

}

RESULTADOS:

