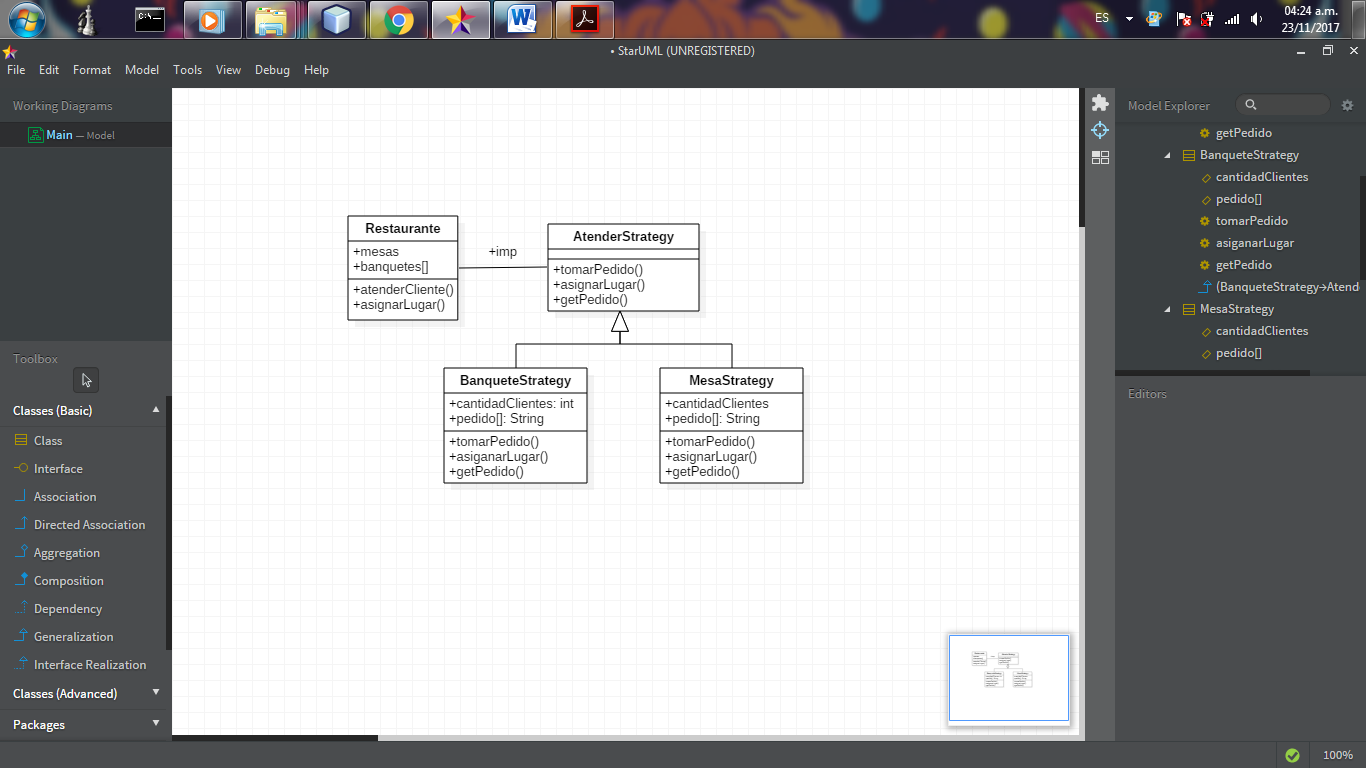
**ESCENARIO:**

Planteamos un escenario que sucedería dentro de un restaurante, la situación problema se presenta en la atención de los dos tipos de espacio que ofrece el restaurante, mesa y banquete, este último teniendo una restricción en la variedad de platos, el restaurante es quien tendrá el contexto y decidirá con qué estrategia atender.



public class StrategyRestaurante {

public static void main(String[] args) {

Restaurante restaurante = new Restaurante();

String estrategia;

int nMesas = 0, nBanquetes = 0, nClientes;

boolean salir = false;

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

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

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

while (salir == false) {

estrategia = JOptionPane.showInputDialog(null, "Lugar De Ubicacion:\n"

+ "(1)Mesa\n"

+ "(2)Banquete (Max 5 Platillos Diferentes) \n"

+ "(3)Salir");

nClientes = Integer.parseInt(JOptionPane.showInputDialog(null, "Cantidad De Personas: "));

switch (estrategia) {

case "1":

restaurante.asignarLugar("M", nMesas, nClientes);

restaurante.atenderCliente("M", nMesas);

nMesas++;

break;

case "2":

restaurante.asignarLugar("B", nBanquetes , nClientes);

restaurante.atenderCliente("B", nBanquetes);

nBanquetes++;

break;

case "3":

salir = true;

break;

default:

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

break;

}

}

}

}

public class Restaurante {

public AtenderStrategy[] mesas;

public AtenderStrategy[] banquetes;

public int nMesas = 0;

public int nBanquetes = 0;

public Restaurante() {

mesas = new MesaStrategy[10];

banquetes = new BanqueteStrategy[10];

}

public void atenderCliente(String tipo, int numeroLugar) {

if (null != tipo) switch (tipo) {

case "B":

banquetes[numeroLugar].tomarPedido();

break;

case "M":

mesas[numeroLugar].tomarPedido();

break;

}

}

public void asignarLugar(String tipo, int posicion, int nClientes) {

if (tipo == "B") {

if (nBanquetes < 10) {

banquetes[nBanquetes] = new BanqueteStrategy(nClientes);

banquetes[nBanquetes].asignarLugar(posicion);

nBanquetes++;

} else {

JOptionPane.showMessageDialog(null, "LUGARES PARA BANQUETE LLENOS...");

}

} else if (tipo == "M") {

if (nMesas < 10) {

mesas[nMesas] = new MesaStrategy(nClientes);

mesas[nMesas].asignarLugar(posicion);

nMesas++;

} else {

JOptionPane.showMessageDialog(null, "MESAS LLENAS...");

}

}

}

}

public interface AtenderStrategy {

public void tomarPedido();

public void asignarLugar(int nLugar);

public Vector<String> getPedido();

}

public class BanqueteStrategy implements AtenderStrategy {

private int cantidadClientes;

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

public int maxPlatillos = 5;

public int numBanquete;

public BanqueteStrategy(int numeroClientes) {

this.cantidadClientes = numeroClientes;

}

@Override

public void tomarPedido() {

String platillo;

for (int p = 0; p < maxPlatillos; p++) {

platillo = JOptionPane.showInputDialog(null, "Platillo Seleccionado: ");

pedido.addElement(platillo);

System.out.println(platillo);

}

}

@Override

public void asignarLugar(int nLugar) {

this.numBanquete = nLugar;

}

@Override

public Vector<String> getPedido() {

return pedido;

}

}

public class MesaStrategy implements AtenderStrategy {

private int cantidadClientes;

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

public int numPlatillos=0;

public int numMesa;

public MesaStrategy(int numeroClientes) {

this.cantidadClientes = numeroClientes;

}

@Override

public void tomarPedido() {

String platillo;

boolean terminado = false;

String opcion;

while (terminado == false) {

try {

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

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

switch (opcion) {

case "1":

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

pedido.addElement(platillo);

System.out.println(platillo);

numPlatillos++;

break;

case "2":

terminado = true;

break;

}

} catch (NumberFormatException ex) {

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

}

}

}

@Override

public void asignarLugar(int nLugar) {

this.numMesa = nLugar;

}

@Override

public Vector<String> getPedido() {

return pedido;

}

}

RESULTADOS:

