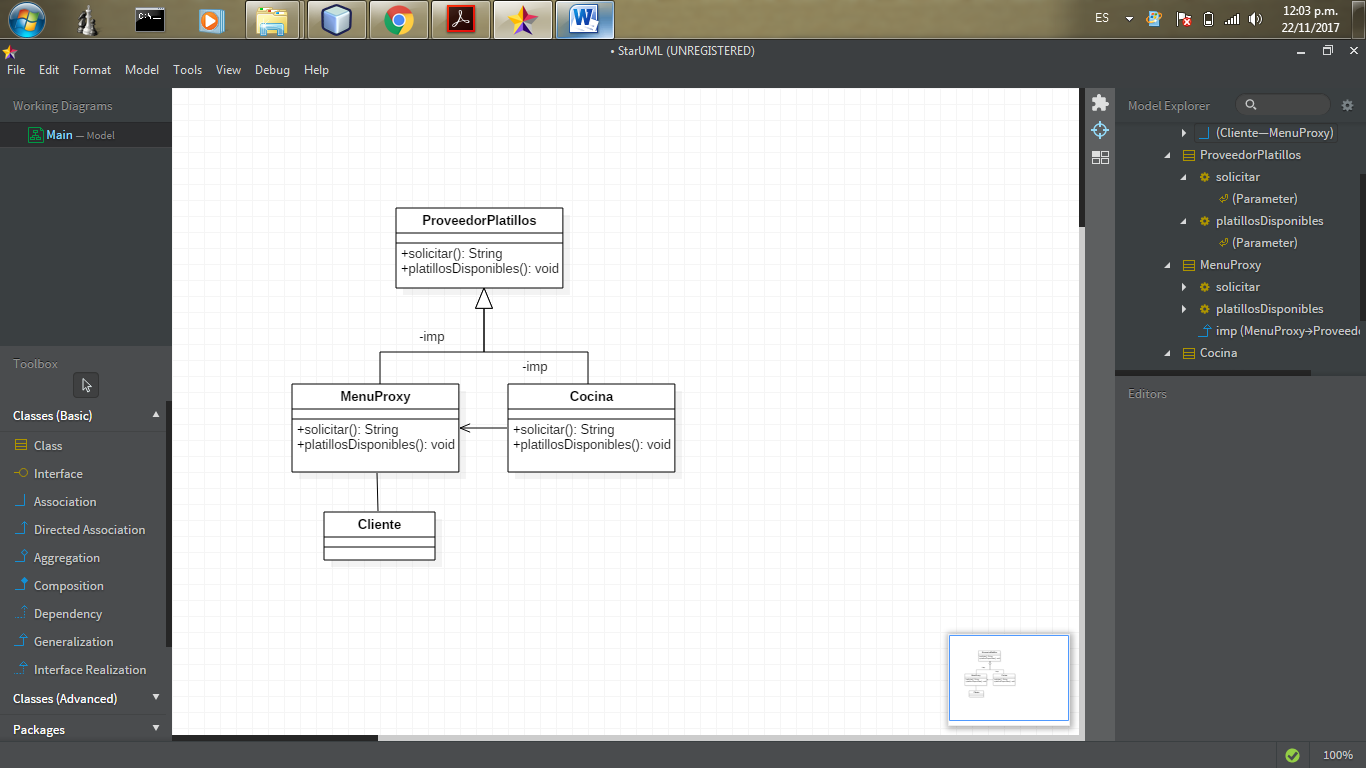
**ESCENARIO:**

Planteamos un escenario que sucedería dentro de un restaurante, la situación problema se presenta para el control de los platillos que están disponibles cada día, evitando el llamado a la cocina sin antes hacer un filtro con los platos disponibles.



public class ProxyMenu {

public static void main(String[] args) {

ProveerPlatillos menu = new MenuProxy();

String opcion, pedido;

boolean salir = false;

String[] platillosCocina = {"lomo de cerdo", "chuletas de cerdo", "carne a la plancha", "carne asada",

"bandeja 2 carnes", "bandeja 3 carnes", "pollo asado", "pollo a la plancha",

"pastel de chocolate", "helado"};

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

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

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

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

for (int x = 0; x < platillosCocina.length; x++) {

System.out.println("(" + (x + 1) + ") " + platillosCocina[x]);

}

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

while (salir == false) {

opcion = JOptionPane.showInputDialog(null, "(1) Solicitar Platillo: \n"

+ "(2) Menu Del Dia \n"

+ "(3) Salir");

try {

switch (opcion) {

case "1":

pedido = JOptionPane.showInputDialog(null, "Digite El Nombre Del Platillo");

System.out.println(menu.solicitar(pedido));

break;

case "2":

menu.platillosDisponibles();

break;

case "3":

salir = true;

break;

default:

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

break;

}

} catch (NumberFormatException ex) {

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

}

}

}

}

public interface ProveerPlatillos {

public String solicitar(String game);

public void platillosDisponibles();

}

public class Cocina implements ProveerPlatillos {

private String[] platillos = {"lomo de cerdo", "chuletas de cerdo", "carne a la plancha", "carne asada",

"bandeja 2 carnes", "bandeja 3 carnes", "pollo asado", "pollo a la plancha",

"pastel de chocolate", "helado"};

public Cocina() {

}

@Override

public String solicitar(String platillo) {

return ("Preparando: " + platillo);

}

@Override

public void platillosDisponibles() {

System.out.println("MENU RESTAURANTE...");

for (int x = 0; x < platillos.length; x++) {

System.out.println("(" + (x + 1) + ") " + platillos[x]);

}

}

}

public class MenuProxy implements ProveerPlatillos {

public Cocina cocina;

private String[] platillosDelDia = {"chuletas de cerdo", "carne asada",

"bandeja 2 carnes", "pollo a la plancha", "helado"};

public MenuProxy() {

cocina = null;

}

public String solicitar(String pedido) {

String platillo;

if (isStored(pedido) == true) {

if (cocina == null) {

cocina = new Cocina();

}

platillo = cocina.solicitar(pedido);

} else {

platillo = ("Platillo NO Disponible El Dia De Hoy.");

}

return platillo;

}

public boolean isStored(String game) {

boolean resultado = false;

for (int x = 0; x < platillosDelDia.length; x++) {

if (game.equals(platillosDelDia[x])) {

resultado = true;

}

}

return resultado;

}

@Override

public void platillosDisponibles() {

for (int x = 0; x < platillosDelDia.length; x++) {

System.out.println("(" + (x + 1) + ") " + platillosDelDia[x]);

}

}

}

RESULTADOS:

