



**INSTITUTO POLITECNICO NACIONAL**

ESCUELA SUPERIOR DE CÓMPUTO

**PROGRAMACIÓN ORIENTADA A OBJETOS**

**Fonda**

Práctica No. 3

**Alumno:**

Nava Hernández Jeovanny Martin

**Boleta:** 2010630345

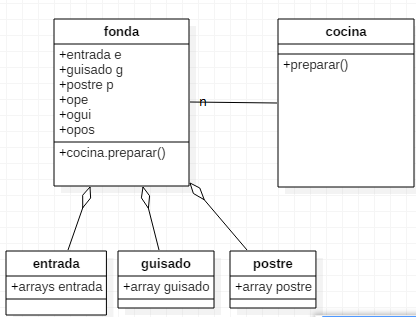
**Grupo:** 2CM7

**Hora límite:** Clase de lab.

**Descripción del problema general:**

Simular una fonda que permita crear un menú

**Diseño**

****

**Codificación**

Entrada.java

1. public class entrada{
2. String[] entrada={"Pasta a la bolognesa", "Arroz a la mexicana","Cebiche"};
3. public String getentrada(int i){
4. return entrada[i-1];
5. }
6. }

Guisado.java

1. public class guisado{
2. String[] guisado={"Chicharron en salsa verde","Tacos al pastor","Papas con longaniza", "Cochinita Pibil"};
3. public String getguisado(int i){
4. return guisado[i-1];
5. }
6. }

Postre.java

1. public class postre{
2. String[] postre={"Helado","Fresas con crema","Malteada de chocolate","Tarta de fresa"};
3. public String getpostre(int i){
4. return postre[i-1];
5. }
6. }

Fonda.java

1. import javax.swing.\*;
2. public class fonda{
3. public static void main(String[] args) {
4. int regresar=0;
5. entrada e=new entrada();
6. guisado g=new guisado();
7. postre p=new postre();
8. JOptionPane.showMessageDialog(null,"Bienvenido a la fonda Capitan Morgan ");
9. JOptionPane.showMessageDialog(null,"Elige una opcion de cada menu ");
10. do{
11. String oen = JOptionPane.showInputDialog("Eliga su entrada\n1.-Pasta \n 2.-Arroz\n 3 .-Cebiche \n");
12. int ope=Integer.parseInt(oen);
13. String gui = JOptionPane.showInputDialog("Elija su guisado\n1.-Chicharron\n 2.-Tacos al pastor\n 3.-Papas \n4.-Cochinita");
14. int ogui=Integer.parseInt(gui);
15. String pos = JOptionPane.showInputDialog("Elija su postre\n1.-Helado\n 2.-Fresas\n 3.-Malteada \n 4.-Tarta de Fresa");
16. int opos=Integer.parseInt(pos);
17. JOptionPane.showMessageDialog(null,"Su menu es:\n"+cocina.preparar(g,ogui)+"\n"+cocina.preparar(e,ope)+"\n"+cocina.preparar(p,opos));
18. String r=JOptionPane.showInputDialog(null, "Deseas ordenar otra cosa\n1.-Volver a ordenar \n2.-Salir");
19. regresar=Integer.parseInt(r);
20. }while(regresar==1);
21. }
22. }

Cocina.java

1. public class cocina{
2. public static String preparar(entrada e, int i){
3. return e.getentrada(i);
4. }
5. public static String preparar(guisado g, int i){
6. return g.getguisado(i);
7. }
   * 1. public static String preparar(postre p, int i){
8. return p.getpostre(i);
   * 1. }
9. }

**Preguntas**

1. **¿Cuál es la idea que sigue la sobrecarga de métodos?**

Permitir que con una sola función o método podamos hacer que funcione para varios casos

1. **¿Con base a qué elementos y condiciones es elegida y ejecutada alguna versión de un método sobrecargado?**

Tipo de retorno, tipo de parámetro y nombre

1. **¿Cuáles son las alternativas para declarar una referencia a un objeto Arreglo en Java?**

private TipoPrimitivoUObjeto [ ] nombreDelArray;

nombreDelArray = new TipoPrimitivoUObjeto [numero];

private int [ ] miArrayDeNumeros = { 2, -3, 4, 7, -10 }; 

1. **En el caso de los métodos static ¿Cuáles son las formas de invocación?**

NombreDeLaClase.nombreDelMétodo (parámetros si los hay);

NombreDelObjeto.nombreDelMétodo (parámetros si los hay);

**Conclusión:**

Los arreglos en java nos proporcionan algunas ventajas y desventajas, entre sus ventajas están las operación y acceso más eficiente y una desventaja podría ser que no se pueden redimensionar.

