

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Ciencias Químicas e Ingeniería

Programas de Ingeniero en Computación e Ingeniero en Software y Tecnologías Emergentes

INFORMACIÓN DE LA MATERIA

Nombre de la materia y clave: Lenguajes de Programación Orientado a Objetos (40006).

Grupo y periodo: 341 (2022-2)

Profesor: Manuel Castañón Puga.

INFORMACIÓN DE LA ACTIVIDAD

Nombre de la actividad: Práctica de laboratorio 5.1.1: Implementación de la persistencia de objetos.

Lugar y fecha: A 25 de noviembre de 2022 en el Edificio 6E, Salón 204.

Carácter de la actividad: En Equipo

Participante(es):

Deysi Belen Rufino Ramos

Francisco Javier Hernandez Ornelas

Diego Andres Gonzalez Beltran

REPORTE DE ACTIVIDADES

Objetivo de la actividad:

En esta práctica se tiene como objetivo adaptar el diagrama de Célula utilizando el diseño de Entity-Control-Boundary. Los controllers que son los objetos que son funcionan como mediador entre los boundaries y entities. Estos orquestan la ejecución los comandos que vengan del boundary. Los boundaries son los objetos que tienen una interfaz con el actor del sistema, por ejemplo interfaces, gateways, proxies, etc. Entity consiste en los objetos que representan los datos del sistema como cliente, transacción, carrito, etc.

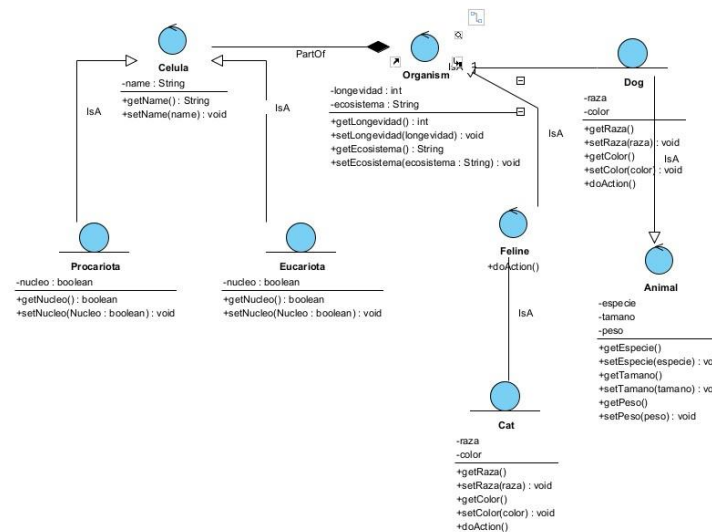


Figura 1. Diagrama Célula actualizado con el diseño ECB

En el código en Java se utilizan las clases `ObjectOutputStream` para almacenar los atributos del objeto a guardar en un archivo con la ayuda de `FileOutputStream` para crear dicho archivo. En `ObjectInputStream` se busca el dicho archivo con `FileInputStream` y recuperamos los datos guardados en la ejecución del programa previa (figura 2).

```
try{
    ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(name: "data.log"));
    oos.writeObject(dog);
    oos.writeObject(proc);
    oos.writeObject(euca);
    oos.writeObject(cat);
    oos.close();
} catch (Exception e){
    e.printStackTrace();
}

try{
    ObjectInputStream ois = new ObjectInputStream(new FileInputStream(name: "data.log"));
    Dog dog2 = (Dog) ois.readObject();
    Procariota proc2 = (Procariota) ois.readObject();
    Eucariota euca2 = (Eucariota) ois.readObject();
    Feline cat2 = (Feline) ois.readObject();
    System.out.println(dog2.toString()+proc2.toString()+euca2.toString());
    System.out.println(cat2.toString());
    cat2.doAction();
    ois.close();
} catch (Exception e){
    e.printStackTrace();
}
```

Figura 2. Persistencia de objetos, guardando y recuperando dog y entre otros objetos.

Ahora el programa imprimé la información correspondiente de lo que se guardó en el archivo, a esto se le llama persistencia de objetos (figura 3).

```
Color: Cafe y blanco
Especie: Canino
Tamano: 50
Peso: 45
Raza: Beagle
Longevidad: 15 años
Ecosistema: Terrestre
Name: Procariota
Existe nucleo? false
Name: Eucariota
Existe nucleo? true

Color: Caf💎
Especie: Felino
Tamano: 50
Peso: 25
Raza: Persa
Longevidad: 14 años
Ecosistema: Terrestre

Meow!
```

Figura 3. Prueba en Java

Con C#, es diferente y se necesita utilizar dos métodos en la clase que se quiere guardar, `GetObjectData` y un constructor de los que se quiere deserializar (figura 3).

```

0 references
public void GetObjectData(SerializationInfo info, StreamingContext context){
    info.AddValue("longevidad", longevidad);
    info.AddValue("ecosistema", ecosistema);
}

0 references
public Organism(SerializationInfo info, StreamingContext context){
    longevidad = (int)info.GetValue("longevidad", typeof(int));
    ecosistema = (String)info.GetValue("ecosistema", typeof(String));
}

```

Figura 3. Constructor y getObjectdata

```

Stream stream = File.Open("Dog.log", FileMode.Create);
BinaryFormatter bf = new BinaryFormatter();
bf.Serialize(stream, dog);
stream.Close();
dog = null;

stream = File.Open("Dog.log", FileMode.Open);
bf = new BinaryFormatter();
dog = (Dog)bf.Deserialize(stream);
stream.Close();
Console.WriteLine(dog.toString());

```

Figura 4. Código en C# para guardar y recuperar

```

Raza: Beagle
Color: Cafe y blanco
Especie: Perro
Tamano: 50
Peso: 45
Longevidad: 15 años
Ecosistema: Terrestre
Name: Procariota
Existe nucleo? False
Name: Eucariota
Existe nucleo? True

Meow!

```

Figura 5. Prueba en C#

Enlace del repositorio de GitHub:

https://github.com/LPOO-FCQI-UABC/Practica_5.1.1_Equipo-B

RESUMEN/REFLEXIÓN/CONCLUSIÓN

En esta práctica se utilizaron clases importadas de serialización para aplicar el concepto de persistencia de objetos en Java y C#.

Aprendimos la funcionalidad de la serialización de objetos además de como diseñar un diagrama con diseño Entity-Control-Boundary.

Concluimos que la persistencia de objetos es un concepto que es bastante importante porque las aplicaciones hoy en día guardan información de usuarios al momento de que no esté activo el programa que ejecutó el usuario. Básicamente, cualquier aplicación que requiera de guardar estados del programa o información requiere aplicar persistencia.

Doy fe de que toda la información dada es completa y correcta.

Deysi Belen Rufino Ramos
Francisco Javier Hernandez Ornelas
Diego Andres Gonzalez Beltran