Unidad

3

DIPLOMATURA EN PROGRAMACION JAVA

Ejercicios

Tecnológica Nacional – Derechos Reservados

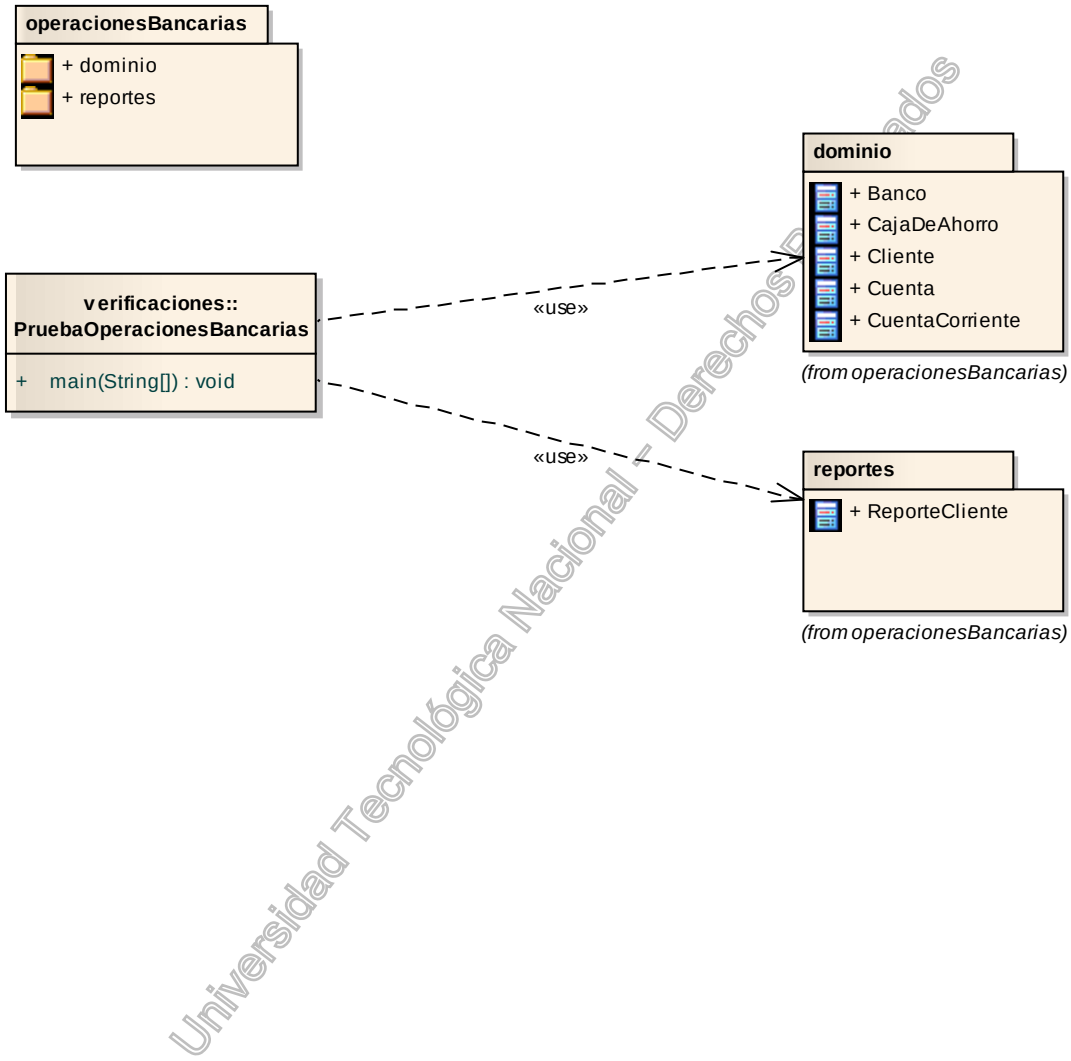
Capítulo 5

Clases: Conceptos Avanzados

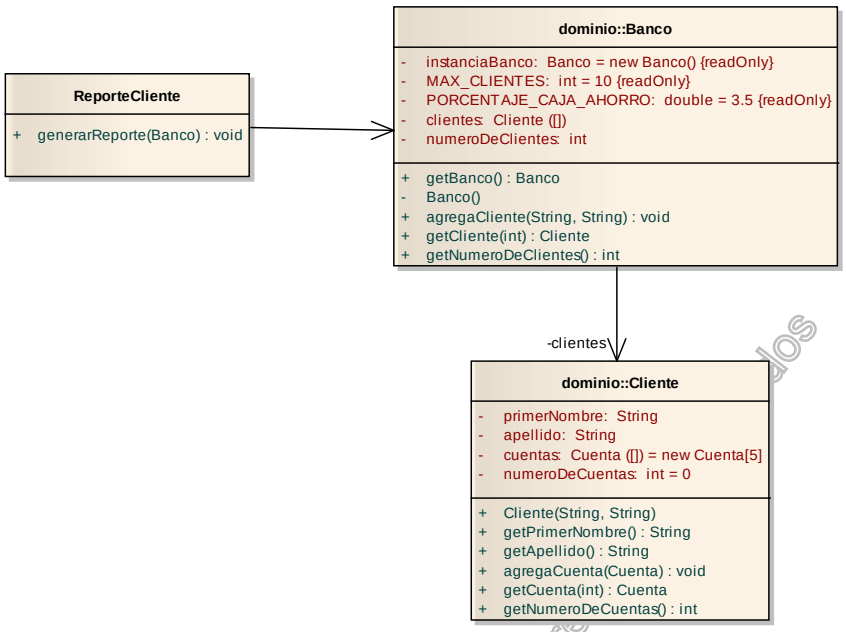
Capítulo 5

Ejercicio 1

En este ejercicio se modificará la clase Banco para implementar el patrón de diseño Singleton, según se muestra en los diagramas UML.



Universidad Tecnológica Nacional - Derechos Reservados



En este punto el proyecto necesita una jerarquía de paquetes más compleja. Por lo tanto, las clases del dominio (principales) necesitan estar en el paquete operacionesBancarias.dominio, razón por la cual se necesita cambiar la declaración de esos archivos.

Realizar los siguientes pasos:

1. Abrir el espacio de trabajo del directorio Capítulo 5\Ejercicio 1. Los errores que figuran luego de importar los archivos se deben a las modificaciones que deberán realizarse.
2. Modificar la declaración package en todas las clases para que estas pertenezcan ahora al paquete operacionesBancarias.dominio. Recuerde que la ubicación del código fuente (los .java) no tiene que estar necesariamente en el mismo directorio al que irán las clases compiladas (los .class), y que la declaración package es para los archivos de este último tipo.

Universidad Tecnológica Nacional – Derechos

Modificar la clase Banco para implementar el patrón de diseño Singleton.

3. Modificar la clase Banco para crear un método estático público llamado getBanco() que retorne una instancia de la clase Banco.
4. Esta instancia única deberá ser un atributo estático con un modificador de acceso (visibilidad) privada y constante. De igual modo, el constructor de Banco deberá ser privado.
5. Modificar el atributo MAX_CLIENTES para que sea constante.

Modificar la clase CajaDeAhorro

6. Modificar el atributo porcentajeInteres y cambiarlo por PORCENTAJE_CAJA_AHORRO para que sea constante.

Modificar la clase ReporteCliente

En el proyecto del ejercicio anterior, se creaba un reporte de clientes que estaba embebido dentro del método main() de la clase PruebaOperacionesBancarias. En este ejercicio, ese código se puso dentro de la clase ReporteCliente en el paquete operacionesBancarias.reportes. Se deberá modificar esta clase para utilizar el objeto del tipo Banco que es un Singleton para que lo utilice esta clase.

7. Buscar la línea de código marcada como un bloque de comentario /***…***/
8. Modificar esta línea para que reciba un objeto de tipo Banco que es un Singleton.
9. Compilar y ejecutar el programa
10. Verificar que la salida sea la siguiente:

```
Creando al cliente Juan Perez
Creando al cliente Pedro García.
Creando al cliente Oscar Toma.
Creando a la cliente Maria Soley.

Recuperando al cliente Juan Perez y su caja de ahorro.
Retira 150.00: true
Deposita 22.50: true
Retira 47.62: true
Retira 400.00: false

Recuperando al cliente Pedro García y su cuenta corriente.
Retira 150.00: true
Deposita 22.50: true
Retira 47.62: true
Retira 400.00: false

Recuperando al cliente Oscar Toma y su cuenta corriente.
Retira 150.00: true
Deposita 22.50: true
Retira 47.62: true
Retira 400.00: true

Recuperando a la cliente Maria Soley y su caja de ahorro unida a su
esposo Oscar.
Deposita 150.00: true
Retira 750.00: true
```

Universidad Tecnológica Nacional – Derechos Reservados

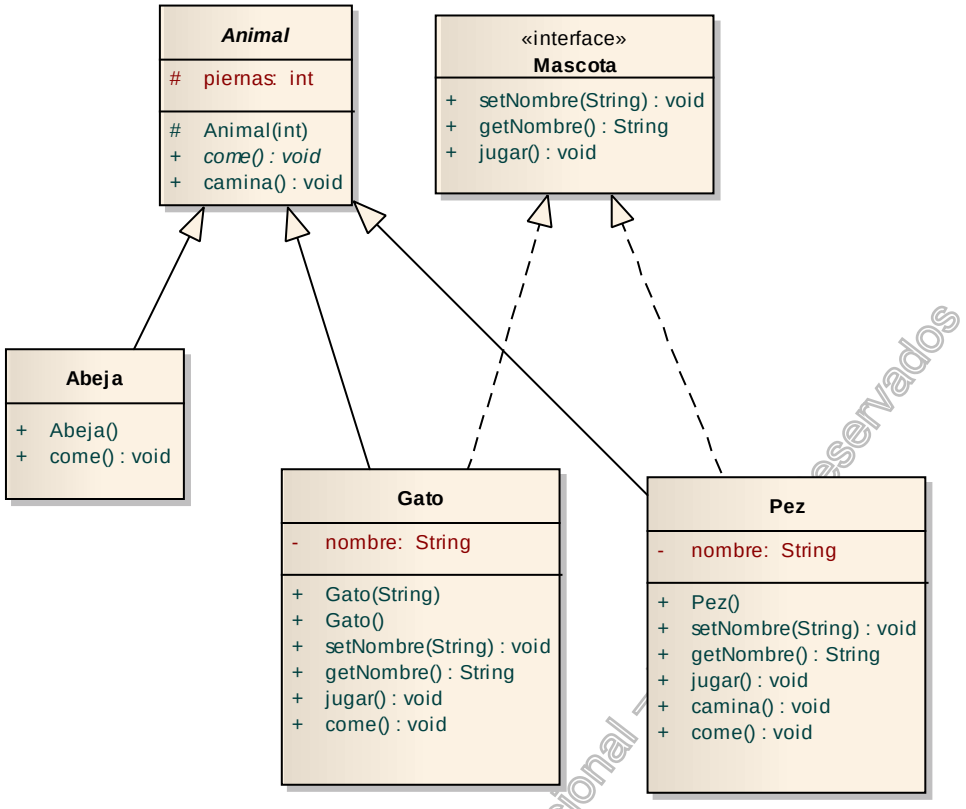
```
                    REPORTE DE CLIENTES
                    ===================

Cliente: Pérez, Juan
    Caja de Ahorro: el balance actual es $222,50
    Cuenta Corriente: el balance actual es $200,00

Cliente: García, Pedro
    Cuenta Corriente: el balance actual es $222,50
    Caja de Ahorro: el balance actual es $380,00

Cliente: Toma, Oscar
    Cuenta Corriente: el balance actual es $0,00
    Caja de Ahorro: el balance actual es $100,00

Cliente: Soley, Maria
    Caja de Ahorro: el balance actual es $100,00
```

Ejercicio 2

En este ejercicio se creará una jerarquía de animales la cual tiene una clase abstracta como base de su cadena de herencia. Varias de las clases “que son” animales implementarán una interfaz llamada Mascota.

En el gráfico UML se muestra el resultado final de la jerarquía a establecer

Universidad Tecnológica Nacional – Derechos Reservados



Universidad Tecnológica Nacional – Todos los derechos reservados

Realizar los siguientes pasos:

1. Crear un espacio de trabajo en el directorio Capítulo 5 y llamarlo Ejercicio 2
2. Crear un proyecto Java y llamarlo Ejercicio 2 también
3. En este ejercicio crear las clases en el paquete “animal”.
4. Crear una clase Animal, la cual es una superclase abstracta de todos los animales
5. Declarar un atributo entero con visibilidad protegida llamado patas, el cual almacena el número de pata de cada animal.
6. Definir un constructor protegido que inicialice el atributo patas
7. Declarar un método abstracto llamado come()
8. Declarar un método concreto (que tenga un cuerpo de sentencias aunque este vacío) llamado camina() que muestre por pantalla un String que indique si el animal camina y con cuantas patas

Crear la clase Abeja

9. La clase Abeja hereda de Animal
10. Definir el constructor por defecto (el que no recibe parámetros) que llamará al constructor de la superclase que especifique que todas las abejas tienen 6 patas.
11. Implementar el método come() (sobrescribirlo y hacerlo concreto).

Crear la interfaz Mascota

Como se especifica en el diagrama UML. Asegurarse que ninguno de los métodos tiene un cuerpo de sentencias.

Crear la clase Gato

12. Esta hereda de Animal e implementa la interfaz Mascota
13. Esta clase deberá incluir un atributo de tipo String para almacenar el nombre de la mascota.
14. Definir un constructor que reciba un parámetro de tipo String y especifique el nombre del gato. Este constructor deberá llamar también al de la superclase para especificar que todos los gatos tienen cuatro patas.
15. Definir otro constructor que no reciba parámetros. Este constructor deberá llamar al anterior (utilizar la palabra reservada this) y pasar un String vacío como argumento
16. Implementar los métodos de la interfaz Mascota.
17. Implementar el método come()

Crear la clase Pez

18. Sobrescribir los métodos de la clase Animal para especificar que un pez no puede caminar y no tiene patas.

Crear un programa VerificarAnimales

19. En el método main() crear y manipular instancias de las clases creadas anteriormente. Un ejemplo podría ser:

```
Pez f = new Pez();
Gato c = new Gato("Manchas");
Animal a = new Pez();
Animal e = new Abeja();
Mascota p = new Gato();
```

20. Experimentar con:
    A. Llamar los métodos de cada objeto
    B. Modificar los tipo en las asignaciones (casting)
    C. Utilizar polimorfismo para llamar a los métodos
    D. Utilizar la palabra reservada super para llamar a los métodos de la superclase.

Universidad Tecnológica Nacional – Derechos Reservados