

Unidad

4

DIPLOMATURA EN PROGRAMACION .NET

Ejercicios

Universidad Tecnológica Nacional - Derechos Reservados

Capítulo 7

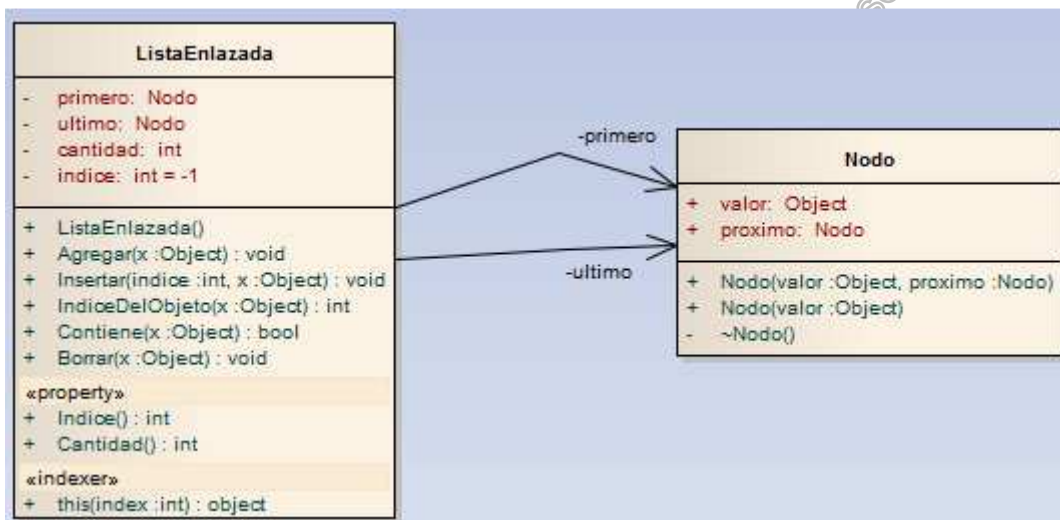
Genéricos

Ejercicio 1

En este ejercicio se va a transformar el código fuente que pertenece a un programa para que utilice genéricos. Las clases fueron diseñadas originalmente para trabajar con cualquier tipo de objeto y se tomó la decisión de diseño de realizarlas en base a Object. Se debe modificar las clases que intervienen en el programa para asegurar los tipos y que se puedan generar objetos utilizando cualquier tipo de dato.

El programa presenta un algoritmo de lista simplemente enlazada donde cada nodo que la conforma es del tipo de la clase Nodo. La clase contenedora se llama ListaEnlazada y administra las operaciones típicas de la lista en la gestión de sus correspondientes nodos.

El siguiente diagrama UML muestra el diseño original:



Se recomienda ejecutar el programa previamente a realizar las modificaciones para comprender su funcionamiento. Para abrir la solución llamada Capítulo7 en el directorio donde estén instalados los ejercicios.

Modificar la clase Nodo

La clase nodo deberá definir un parámetro genérico y gestionarlo como el valor que almacenará cada nodo. Realizar las siguientes acciones modificando la clase para cumplir los siguientes objetivos:

1. Definir un parámetro genérico para la clase.
2. Definir un atributo privado, que sea del tipo del parámetro genérico y que inicialmente asuma el valor por defecto para el tipo que se pase como argumento
3. Definir un atributo del tipo `Nodo` que tenga un parámetro genérico para guardar la referencia al próximo nodo

4. Modificar los argumentos que recibe el constructor que define dos parámetros para que uno sea el valor a almacenar en forma de un tipo genérico y la referencia del segundo argumento consistente con la nueva declaración genérica de la clase Nodo.
5. Modificar el constructor que recibe un solo argumento para que sea consistente con la nueva declaración de la clase Nodo
6. Definir dentro del constructor una asignación al valor por defecto para el tipo genérico que representa al valor almacenado por el nodo y una referencia nula en la variable que apunta al próximo Nodo de la lista.

Modificar la clase ListaEnlazada

Las modificaciones realizadas en la clase Nodo exigen que se maneje el parámetro genérico en la clase ListaEnlazada para que ésta gestione adecuadamente las tareas administrativas de cada uno de los elementos que contiene.

7. Agregar un parámetro genérico a la clase ListaEnlazada
8. Modificar la definición de los atributos de referencia al primer y al último elemento de la lista.
9. Modificar el método Agregar para que maneje el parámetro genérico
10. Modificar el método Insertar para que maneje el parámetro genérico
11. Modificar el método IndiceDelObjeto para que maneje el parámetro genérico
12. Modificar el método Contiene para que maneje el parámetro genérico
13. Modificar el método Borrar para que maneje el parámetro genérico
14. Modificar el indexador para que maneje el parámetro genérico. Tener en cuenta que la declaración y uso de un indexador no se ve afectada por los valores que retorna, *que pueden ser de cualquier tipo*.

Modificar la clase Program

14. Declarar adecuadamente la clase ListaEnlazada para que reciba como argumento del parámetro genérico una cadena que es el tipo de objeto que se almacenará en los nodos.
15. Cambiar la definición del parámetro del método estático MostrarLista para que reciba un argumento definido acorde las modificaciones que se hicieron para manejar genéricos en ListaEnlazada.

Ejecución del programa

Verificar que la salida sea la misma que se obtenía antes de realizar las modificaciones y que es igual a la que se muestra a continuación:

```
Error: Valor de índice fuera de rango
-----
Uno
Dos
Tres
-----
Uno
Tres
-----
```

```
Uno
-----
Uno
Fin
-----
Uno
Otro
Fin
-----
Primero
Uno
Otro
Fin
-----
Primero
El Medio
Uno
Otro
Fin
Contiene 'El Medio' True
El indice de 'El Medio' es 1
Contiene 'Dos' False
El indice de 'Dos' es -1
Contiene 'Otro' True
El indice de 'Otro' es 3
```

Ejercicio 2

Con el fin de mejorar el algoritmo realizado, el área de diseño le solicita que genere una interfaz en base a la funcionalidad en común que tiene el algoritmo de lista enlazada genérica realizada en el ejercicio anterior. La interfaz deberá ser genérica, llamarse ILista y manejar los siguientes elementos:

1. El método Agregar
2. El método Insertar
3. El método IndiceDelObjeto
4. El método Contiene
5. El método Borrar
6. La propiedad Cantidad
7. La propiedad Indice
8. El indexador definido

Además, debe implementar la interfaz creada en la clase ListaEnlazada de manera que los métodos modificados en el ejercicio anterior sean ahora los que se describen de la interfaz.

También debe modificar el método estático MostrarLista para que ahora reciba como argumento la interfaz creada y que el programa se ejecute correctamente con todas estas modificaciones.

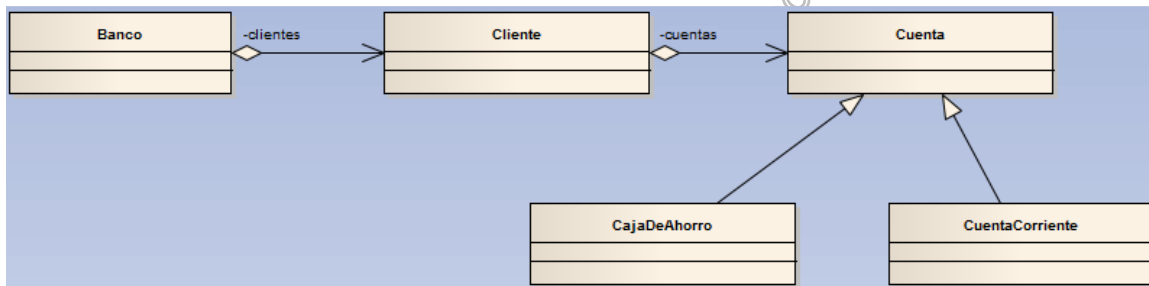
Para realizar todas las tareas descriptas, crear un nuevo proyecto, llamarlo ejercicio2, copiar las clases de la solución del ejercicio1 y cambiar el nombre del espacio de nombres en cada clase a ejercicio2 (en VB no es necesario porque lo toma por defecto).

Ejercicio 3

En este ejercicio se va a usar la clase ListaEnlazada modificada en el ejercicio 2 para representar asociaciones de clases en el modelo de dominio del proyecto que maneja las operaciones bancarias.

En el diseño anterior se usaron vectores para implementar la multiplicidad en las relaciones entre el banco sus clientes, y entre los clientes y sus cuentas. Este diseño tiene varias limitaciones importantes, la principal es que el vector, una vez creado, tiene un tamaño fijo.

La siguiente figura muestra el modelo de dominio del proyecto con las asociaciones de clases: un banco presta servicio (tiene) a muchos clientes y un cliente tiene muchas cuentas.



Copiar las clases de la solución del ejercicio 2

1. Copiar las clases Nodo y ListaEnlazada además de la interfaz ILista

Modificación de la clase Banco

2. Eliminar la declaración `private int MAX_CLIENTES = 10;` o `Private MAX_CLIENTES As Integer = 10` (C# o VB)
1. Modificar la declaración de la variable de instancia clientes para que sea del tipo `ListaEnlazada<Clientes>`. Incluir el espacio de nombres definido para la clase (debería ser ejercicio 2 si no se modificó). Borrar la inicialización de la variable de instancia `numeroDeClientes` en el constructor. Crear el objeto del tipo `ListaEnlazada<Clientes>` en el constructor.
2. Modificar el método `AgregarCliente`. Eliminar la declaración de la variable entera como el incremento y asignación a ésta de la variable de instancia `numeroDeClientes`. Reemplazar la asignación del nuevo cliente al vector por el agregado de un nuevo nodo de la lista. Asignar la propiedad `Cantidad` de la lista enlazada a la variable `numeroDeClientes`, luego de incorporar el nuevo cliente.

Modificación de la clase Cliente

3. Modificar la declaración de la variable de instancia cuentas para que sea del tipo `ListaEnlazada <Cuenta>`.

4. Borrar la inicialización de la variable de instancia numeroDeCuentas en el constructor. Crear el objeto del tipo ListaEnlazada<Clientes> en el constructor.
5. Modificar el método AgregaCuenta para que utilice el objeto del tipo ListaEnlazada para agregar cada cuenta y para que la variable de instancia numeroDeCuentas almacene el valor de la propiedad Cantidad de dicho objeto, luego de incorporar la nueva cuenta.

Verificación de la clase ReporteCliente

6. Corroborar que la clase ReporteCliente no tiene ningún error de compilación

Ejecución del programa

7. Ejecutar el programa desde la clase PruebaOperacionesBancarias. La salida debe ser similar a la siguiente:

El Cliente [Perez, Juan] tiene un balance en cuenta corriente de 500 con una protección por sobregiro de 500.00.

Cuenta Corriente [Juan Perez]: retira 150.00

Cuenta Corriente [Juan Perez]: deposita 22.50

Cuenta Corriente [Juan Perez]: retira 147.62

Cuenta Corriente [Juan Perez]: retira 470.00

Excepción: Fondos Insuficientes Deficit: 245,12

El Cliente [Perez, Juan] tiene un balance en cuenta corriente de 224,88

El Cliente [Toma, Oscar] tiene un balance en cuenta corriente de 200

Cuenta Corriente [Oscar Toma]: retira 100.00

Cuenta Corriente [Oscar Toma]: deposita 25.00

Cuenta Corriente [Oscar Toma]: retira 175.00

Excepción: No hay protección por sobregiro Deficit: 50

El Cliente [Toma, Oscar] tiene un balance en cuenta corriente de 125

El Cliente [García, Pedro] tiene un balance en cuenta corriente de 500

Cuenta Corriente [Pedro García]: retira 50.00

Cuenta Corriente [Pedro García]: deposita 250.00

Cuenta Corriente [Pedro García]: retira 400.00

El Cliente [García, Pedro] tiene un balance en cuenta corriente de 300

REPORTE DE CLIENTES

=====

Cliente: Perez, Juan

Cuenta Corriente: el balance actual es 200

Caja de Ahorro: el balance actual es 224,88

Cliente: Toma, Oscar

Cuenta Corriente: el balance actual es 125

Caja de Ahorro: el balance actual es 700

Cliente: García, Pedro

Cuenta Corriente: el balance actual es 300

Caja de Ahorro: el balance actual es 900

Cliente: Soley, Maria

Caja de Ahorro: el balance actual es 700

Universidad Tecnológica Nacional - Derechos Reservados