

Programación Orientada a Objetos.

Práctica 08.

Objetivos:

- Practicar el uso de colecciones en un programa orientado a objetos.
- Probar la funcionalidad de “generics” en OOP.
- Implementar colecciones en el sistema bancario.

Antes de iniciar los ejercicios, baje de la página del curso, sección Material, el archivo Practica08.zip al escritorio y descomprímalo para obtener los archivos necesarios para esta práctica.

Ejercicio 1. Implementar un diccionario inglés-español utilizando una colección Map.

- 1) Cree un nuevo proyecto de Netbeans llamado Practica08 de tipo Java Application.
Deseleccione la casilla Create Main Class.
- 2) Dentro del proyecto Practica08, cree un paquete llamado ejercicio1.
- 3) En el paquete ejercicio1, cree una clase de Java llamada Diccionario que lleve a cabo lo siguiente
 - a) Defina una variable de instancia privada de clase HashMap (implementación de la interfaz Map) de tipo genérico cuyos dos elementos, llave y valor sean de tipo String. Como la clase HashMap está en el paquete java.util, siga la sugerencia de Netbeans para incluir el postulado import necesario. **Nota.** la llave del mapa representa una palabra en inglés que se sugiere llamar **ingles** y el valor es su significado en español que puede llamarse **espanol**.
 - b) Incluya un constructor que instancie un objeto de la clase definida en el inciso anterior y a continuación invoque el método privado **cargaDiccionario** que se describe más adelante.
 - c) Defina un método **public String get(String ingles)** que, utilizando el método **get** de la interfaz Map, obtenga el significado de la palabra y lo entregue como resultado del método.
 - d) Defina otro método **public void put(String ingles, String espanol)** que, utilizando el método **put** de la interfaz Map introduzca en el diccionario la palabra en inglés y su

significado en español.

- e) Defina un método privado de tipo void llamado **cargaDiccionario** que cargue en el diccionario unas cinco o seis palabras en inglés con su significado en español utilizando el método put de la clase Diccionario.
- 4) Copie la clase proporcionada en el material de la práctica llamada ConsultaDiccionario.java al paquete ejercicio1 y estudie su contenido. Este programa lee una palabra en inglés de la consola y despliega su significado extrayéndolo del diccionario. Estudie el programa y note principalmente el procedimiento para leer de la consola que estudiaremos en el siguiente módulo del curso.
- 5) Ejecute el programa ConsultaDiccionario verificando su correcto funcionamiento.

Ejercicio 2. Implementar los conjuntos de clientes y cuentas del Sistema Bancario como colecciones Java.

- 1) Abra el proyecto de netbeans BankLtd que se ha venido desarrollando hasta ahora.
- 2) Borre la clase TestBanking de la versión anterior del proyecto.
- 3) Modifique la clase Bank, para usar un ArrayList para implementar la lista de clientes. Utilice la sintaxis de la funcionalidad Generics en donde aplique. No olvide importar clases del paquete java.util.
 - a. Modifique la declaración del atributo customers para que sea de tipo ArrayList en vez de arreglo de Customers.
 - b. Elimine el atributo numOfCustomers.
 - c. Modifique el constructor de Bank para inicializar customers como objeto de clase ArrayList.
 - d. Modifique el método addCustomer para que utilice el método add de ArrayList.
 - e. Modifique el método getCustomer para que utilice el método get de ArrayList.
 - f. Modifique el método getNumberOfCustomers para que use el método size de ArrayList.
 - g. Añada un método getCustomers, que entregue un ListIterator para la lista de clientes.
- 4) Modifique la clase Customer para usar un ArrayList para implementar las cuentas. Use el mismo procedimiento delineado en el punto anterior utilizando los métodos correspondientes y añadiendo el método getAccounts.
- 5) Modifique la clase CustomerReport de la siguiente manera:

- a. Obtenga el iterador de clientes utilizando el método `getCustomers`.
 - b. Substituya el primer ciclo `for` por un ciclo `while` que use el método `hasNext()` del iterador de clientes como variable de control.
 - c. Obtenga el cliente utilizando el método `next()` del iterador. (Es necesario un casting que será sugerido por Netbeans).
 - d. Obtenga el iterador de cuentas utilizando el método `getAccounts`.
 - e. Substituya el segundo ciclo `for` por otro ciclo `while` que utilice el método `hasNext()` del iterador de cuentas como variable de control.
 - f. Obtenga la cuenta utilizando el método `next()` del iterador. (También requiere casting sugerido por Netbeans).
- 6) Copie la clase proporcionada en el material de la práctica llamada `TestBanking.java` al paquete `banking` y estudie su contenido.
 - 7) Ejecute el programa `TestBanking` y estudie el resultado.

Diagrama de Clases.

