7mo Ensayo de la clase

Colecciones en Java

Una colección en Java sería un conjunto de elementos, la característica de este conjunto de elementos es que sería un objeto que almacena un conjunto de objetos, otra característica es que son de tamaño variable es decir pueden crecer o decrecer según la necesidad o lo requerido.

Además el marco de trabajo de las colecciones proporciona implementaciones de alto rendimiento y alta calidad de las estructuras de datos comunes, y permite la reutilización de código.

Algunas de las interfaces del marco de trabajo de colecciones son las siguientes:

- **Collection.-** La interfaz raíz en la jerarquía de colecciones, a partir de la cual se derivan las interfaces Set, Queue y List.
- **Set.-** Una colección que no contiene duplicados.
- **List.-**Una colección ordenada que puede contener elementos duplicados.
- **Map.-** Asocia claves con valores y no puede contener claves duplicadas.
- **Queue.-**Es una colección del tipo primero en entrar, primero en salir, que modela a una línea de espera.

Interfaz Collection y Clase Collections

Interfaz Collection

Como se dijo anteriormente la interfaz Collection es la raíz en la jerarquía de colecciones a partir de la cual se derivan las interfaces Set, Queue, List; podemos mencionar además que en la interfaz Collection contiene diversas operaciones para añadir, borrar, comparar objetos que son elementos de dichas colecciones. Además de ello contiene métodos que devuelven un objeto Iterator el cual es usado para recorrer las distintas colecciones.

Clase Collections

La clase Collections es una clase que proporciona métodos de tipo static que manipulan las colecciones de manera polimórfica, estos métodos implementan algoritmos para buscar, ordenar, entre otros.

List

List es el tipo de colección del cual hicimos uso la clase anterior. Una definición para List sería la siguiente: Es un objeto Collection ordenado que puede contener elementos duplicados. La interfaz List es implementada por varias clases, entre ellas tenemos ArrayList, LinkedList y Vector, de todas ellas la que vamos a usar es ArrayList.

Un ejemplo de instanciación de una lista es el siguiente:

List <Vehiculo> vehículos= new ArrayList<Vehiculo>();

Algunos de los Métodos básicos para el manejo de una lista.

- **add(Object o).-** Añade un elemento a la lista.
- **get(int posición).-** Extrae el objeto almacenado en la posición indicada. Es necesario indicar el tipo de objeto que se extrae.
- **size().-** Devuelve el número de elementos almacenados en la lista.

Un ejemplo sería el siguiente:

```
package practica_dirigida_1;
import java.util.*;

public class Procesos {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Taller taller1=new Taller();
        Persona usuario;
        List <Persona> temp=new ArrayList<Persona>();
        Scanner scan=new Scanner(System.in);
        System.out.println("Ingrese la cantidad de personas a registrar");
        taller1.setCantidad(scan.nextInt());
        for(Integer i=0;i<taller1.getCantidad();i++){
            usuario=new Persona();
            System.out.println("Cliente "+(i+1));
            System.out.print("Nombre");
            usuario.setNombre(scan.next());
            System.out.print("Apellido");
            usuario.setApellido_paterno(scan.next());
            System.out.print("Email");
            usuario.setEmail(scan.next());
            temp.add(usuario);
        }
        taller1.setClientes(temp);
        taller1.generar_lista();
    }
}
```

El programa anterior hace uso de dos clases: Taller y Persona (cada clase contiene su respectivos métodos getter y setter) los atributos de cada una de las clases son los siguientes:

Atributos de la clase Persona:

```
private String nombre;
private String apellido_paterno;
private String email;
```

Atributos de la clase Taller:

```
private List <Persona> clientes=new ArrayList<Persona>();
private Integer cantidad;
```

Luego del primero se crea un objeto taller1, y de Persona un objeto usuario el cual va ser usado para adicionar elementos a la lista, y se crea una lista denominada temp, en este programa se hace uso de método add para añadir objetos de tipo Persona, luego taller1 hace uso del método generar_lista(), cuya sintaxis es la siguiente:

```
public void generar_lista(){
    for(int i=0;i<this.cantidad;i++){
        System.out.println(clientes.get(i).toString());
    }
}
```

Con el método get(i) accedemos a los elementos contenidos en la lista clientes y además de cada uno de ellos hacemos una llamada al método toString().

Bibliografía

- Fundamentos de programación en Java — Jorge Martínez Ladrón de Guevara
- Como Programar en Java — Deitel
- http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188

Mendoza Medrano Adrián Esteban