Resolución Trabajo Práctico - Especificaciones de acceso

Teoría

1. ¿Cuál es la utilidad de los modificadores de acceso? ¿Qué tipos de modificadores existen en el lenguaje Java?

   Permiten que un desarrollador (incluso quien desarrolla el código) permita o prohíba el uso de ciertos métodos o atributos, de acuerdo a cómo se haya diseñado la clase. De esta manera, se evitan errores por el mal uso de una clase por parte de otras clases. En Java hay 4 modificadores: por defecto (vacío), private, public, protected.

2. ¿Qué diferencia existe entre colocar public y no colocar nada en la declaración de un método o atributo?

   El modificador de acceso por defecto está limitado a la clase actual y clases del paquete actual. Las subclases y clases de otros paquetes no pueden acceder a métodos/atributos con modificadores por defecto. public es el modificador más permisivo, el método/atributo puede accederse desde cualquier parte del proyecto.

3. ¿Qué restricciones impone el modificador private en un método o atributo?

   Restringe el acceso sólo al código dentro de una clase.

4. ¿En qué casos se utiliza el modificador protected?

   Cuando se desea dar la posibilidad de extender una funcionalidad de una clase mediante herencia, pero no se desea que otras clases en otros paquetes puedan ver dicha funcionalidad.

5. ¿Qué modificadores se pueden utilizar en la declaración de una clase?

   public o por defecto (vacío). Existe el caso de clases anidadas, donde aplican los demás modificadores, pero no se verá en el curso.

Práctica

1. Proteja las variables y métodos que considere necesario, para evitar que un desarrollador (incluso usted) utilice erróneamente la siguiente clase:

```
public class Avión {
	List<Persona> pasajeros = new ArrayList<>();
	int pesoTotal= 0;
	static final int PESO_MAXIMO= 1000;

	int getPesoTotal() {
		return pesoTotal;
	}

	void actualizarPeso(int peso) {
		pesoTotal += peso;
	}
```

```
	boolean agregarPersona(Persona pasajero) {
		pasajeros.add(pasajero);
		if (getPesoTotal() + pasajero.getPeso() > PESO_MAXIMO)
			return false; //El pasajero sobrepasa el máximo
		actualizarPeso(pasajero.getPeso());
		return true;
	}

	void borrarPasajero(Persona pasajero) {
		pasajeros.remove(pasajero);
		actualizarPeso( - pasajero.getPeso());
	}
}
```

Solución Propuesta:

```
public class Avión {
	// Estos atributos deberían ser privados para que no
	// se modifiquen externamente.
	private List<Persona> pasajeros = new ArrayList<>();
	private int pesoTotal= 0;
	private static final int PESO_MAXIMO= 1000;

	int getPesoTotal() {
		return pesoTotal;
	}
	// Este método no se debería llamar desde afuera.
	private void actualizarPeso(int peso) {
		pesoTotal += peso;
	}

	boolean agregarPersona(Persona pasajero) {
		pasajeros.add(pasajero);
		if (getPesoTotal() + pasajero.getPeso() > PESO_MAXIMO)
			return false; //El pasajero sobrepasa el máximo
		actualizarPeso(pasajero.getPeso());
		return true;
	}

	void borrarPasajero(Persona pasajero) {
		pasajeros.remove(pasajero);
		actualizarPeso( - pasajero.getPeso());
```

```
	}
}
```

2. Dada la clase Noticia:

```
public class Noticia{
	private String titulo;
	private String contenido;

	// Retorna el contenido con la primer letra capital.
	private String getContenidoConFormato() {
		return contenido.substring(0,1).toUpperCase()
			+ contenido.substring(1);
	}

	// Retorna el título todo en mayúsculas
	private String getTituloConFormato() {
		return titulo.toUpperCase();
	}

	public String imprimir(){
		return getTituloConFormato() + “\n”
			+ getContenidoConFormato();
	}
}
```

Realice las modificaciones necesarias sobre Noticia que permitan extenderla. El objetivo es crear una subclase llamada NoticiaResumida que solo muestra los primeros 50 caracteres del contenido más “...” (3 puntos suspensivos) que indican que el contenido continúa. Su titulo también es limitado a 20 caracteres, pero no se colocan puntos.

(Cambios en negrita)

```
public class Noticia{
	private String titulo;
	private String contenido;

	public String getTitulo() {
		return titulo;
	}

	public String getContenido() {
		return contenido;
	}
```

```
	// Retorna el contenido con la primer letra capital.
	protected String getContenidoConFormato() {
		return contenido.substring(0,1).toUpperCase()
			+ contenido.substring(1);
	}

	// Retorna el título todo en mayúsculas
	protected String getTituloConFormato() {
		return titulo.toUpperCase();
	}

	public String imprimir(){
		return getTituloConFormato() + "\n"
			+ getContenidoConFormato();
	}
}

public class NoticiaResumida extends Noticia{
	protected String getContenidoConFormato() {
		return super.getContenido().substring(0, 100) + "...";
	}
	protected String getTituloConFormato() {
		return super.getTitulo().substring(0, 50);
	}
}
```