Ejercicios Interfaces

Semana del 16 de Marzo

1) Diseñar el API de una clase de utilidad con un método de ordenación “sort”, para ordenar listas de String. El método sort debe ordenar los elementos de la lista por un criterio de orden a definir por el usuario, y que se le pasa como parámetro.

2) Crea una clase de utilidad `PrettyPrinter` para imprimir por pantalla estructuras en forma de árbol de manera indentada. La clase de utilidad ha de ser altamente reutilizable, por ejemplo con la clase Directorio del ejericio de la sección anterior.

3) Se quiere construir una clase `SemiGroup`, que modele el concepto matemático de semigrupo. Un semigrupo es un conjunto con una operación que toma dos elementos del conjunto y produce otro elemento del conjunto. Por simplificar, consideraremos que los semigrupos son conjuntos de enteros y no haremos ninguna suposición adicional sobre la operación (que debería ser asociativa).

Así pues, la clase `SemiGroup` debe tener un constructor que tome como parámetros la operación y una colección de enteros. La operación debe ser conforme a la siguiente interfaz.

```
interface IOperation {
   Integer operate(Integer a , Integer b );
}
```

Un semigrupo debe tener un método `calculate`, que invoca la operación y devuelve null si los parámetros o el resultado no pertenecen al conjunto. Como ves en el código de más abajo, la operación `AddModulo` implementa la interfaz `IOperation` realizando la suma módulo el número que se le pasa como parámetro en el constructor.

Completa el siguiente programa para que se obtenga la salida de más abajo

```
package semiGroup;
import java.util.Arrays;

public class SemiGroupMain {
	public static void main(String[] args) {
		SemiGroup sg = new SemiGroup(new AddModulo(4), Arrays.asList(0, 1, 2, 3));

		System.out.println(sg.calculate(1, 2));
		System.out.println(sg.calculate(3, 4));
		System.out.println(sg);
	}
}
```

Salida esperada

```
3
null
[0, 1, 2, 3] with operation + modulo 4
```

Nota: Una mejor idea para señalar los dos tipos de errores es generar excepciones en lugar de devolver null (agregaremos esta característica en la próxima lección).