

2do Parcial Práctico Recuperatorio

Ejercicio 1: Dado la siguiente clase, indicar:

- que produce la ejecución del método main por pantalla
- explique cuáles son las consecuencias de reemplazar la sentencia “**excepciones++**” por “**throw ex**”
- A su criterio, ¿le parece acertado utilizar RuntimeException para capturar la excepción? Justifique su respuesta.

```
public class QueHace {

    public static Set<Integer> obtenerValoresNumericos(List<String> numeros){

        Set<Integer> valoresNumericos = new HashSet<Integer>();

        Integer excepciones = 0;

        try {
            for (String n: numeros){
                valoresNumericos.add(Integer.parseInt(n));
            }
        } catch (RuntimeException ex) {
            System.out.println("Error al convertir.");
            excepciones ++;
        }

        System.out.println("cantidad de excepciones: "+ excepciones);
        return valoresNumericos;
    }

    //
    public static void main(String[] args) {

        List<String> numeros = Arrays.asList("uno", "2", "tres", "2", "4", "cinco");

        for(Integer i : obtenerValoresNumericos(numeros)){
            System.out.println(i);
        }

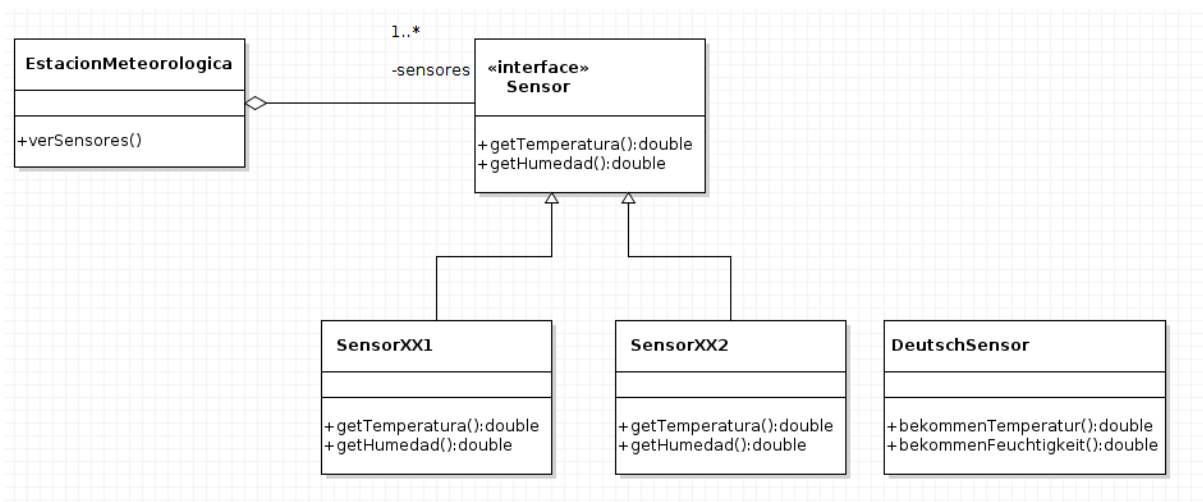
        System.out.println("terminando");
    }
}
```

Ejercicio 2: Una estación meteorológica posee una serie de sensores, periódicamente realiza un listado de la temperatura y humedad de los mismos. La estación adquirió recientemente nuevos sensores a un proveedor Aleman. Se entregaron los sensores con su respectivos drivers, pero no el código fuente de los mismos. Afortunadamente, la documentación entregada nos permite conocer su interfaz y además reveló que retornan la temperatura en grados fahrenheit y no en grados celsius como los sensores que ya poseía la estación meteorológica. Se desea integrar los sensores recientemente adquiridos al sistema existente.

Nota: No se puede modificar el método de la clase DeutschSensor porque no se cuenta con el código fuente. Recuerde que para convertir grados fahrenheit a celsius se utiliza la formula $C=(F-32)*(5/9)$

Se pide:

- Modelar en UML la solución al problema mostrando el código de los métodos y atributos que considere necesarios para su correcta interpretación.



Ejercicio 3: Dado el siguiente problema se pide que **diseñe** una solución, representándola en **UML**, y luego **implemente** la misma. Finalmente **incorpore una clase llamada Prueba que muestre como es la utilización de su propuesta**.

Se debe permitir el armado de expresiones aritméticas para su posterior evaluación. La expresión contiene valores numéricos enteros y operaciones como la suma, la resta, la multiplicación y la división.

Por simplicidad las expresiones vienen representadas en notación postfija. Ej de expresiones:

10 5 + ---> [10 + 5 luego de evaluar la expresión se retorná 15]
 2 3 * 4 - ---> [(2 * 3) - 4 valor resultado = 2]
 8 1 + 4 1 - / ---> [(8+1) / (4-1) valor resultado = 3]

Usted cuenta con una librería que se encarga de establecer la expresión en formato cadena y luego retornar los tokens individuales a procesar. Los valores de retornos pueden ser String representado un número o el símbolo de la operación "+", "-", "*", "/". Si el valor de retorno es "null" significa que no hay más elementos de la expresión. Si por algún motivo la expresión se encuentra mal formada, la llamada al método nextToken lanzará una excepción.

Para el ejemplo anterior, se establece el texto de la expresión con el método setText("10 5 +") y luego la primer llamada al método nextToken() retorna "10", la segunda "5", la tercera "+" y la cuarta "null".

Métodos de la librería Parser:

```

public static void setText(String texto){...}
public static String nextToken() throws TokenException {...}
    
```

Nota: Considere el uso de estructuras del tipo pila y/o cola como complemento a su solución (no hay que implementarlas).