

Programación en Lenguaje Java

Problema 9.2. Cuestiones excepciones



Michael González Harbour

Mario Aldea Rivas

Departamento de Matemáticas,
Estadística y Computación

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Problema 9.2. Cuestiones Excepciones.

Objetivos

- Practicar el patrón de excepciones recuperables.
- Practicar los criterios de selección del tipo de excepción a utilizar.

Desarrollo

Responder a las siguientes cuestiones.

Cuestión 1. Patrón de excepciones recuperables (examen septiembre 2011)

La clase `UsuariosServicio` que permite registrar usuarios en un servicio de internet se encuentra parcialmente implementada:

```
public class UsuariosServicio {  
  
    ... // atributos y métodos no relevantes para el problema  
  
    /**  
     * Registra un usuario en el servicio y le asigna el  
     * alias indicado.  
     * @param nombreCompleto nombre y apellidos del usuario  
     * @param alias nombre que el usuario desea utilizar para  
     * identificarse en el servicio  
     * @throws AliasYaUtilizado lanzada si el alias ya está siendo  
     * utilizado por otro usuario  
     */  
    public static void registraUsuario(String nombreCompleto,  
                                       String alias)  
        throws AliasYaUtilizado {  
        ... // código no relevante para el problema  
    }  
  
}
```

Se pide escribir el código de un método que llame al método `registraUsuario` y, utilizando el patrón de excepciones recuperables, verifique el comportamiento descrito por el siguiente comentario de documentación:

```
/**  
 * Pide los datos de un usuario (nombre completo y alias) mediante  
 * una ventana de lectura del paquete fundamentos. Permite repetir  
 * la introducción de los datos hasta que se elija un alias no  
 * utilizado o hasta que se cancele la operación introduciendo un  
 * string vacío en la entrada "Alias" de la ventana de lectura.  
 * @return true si se ha registrado el usuario o false si se ha  
 * cancelado la operación.  
 */
```

Cuestión 2. Uso de excepciones comprobadas vs. no comprobadas

Se está desarrollando la clase `ControlClimatizador` que sirve para controlar el sistema de climatización de un coche:

```
public class ControlClimatizador {
    // atributos
    ...

    public double maxRangoTemperatura() { ... }

    public double minRangoTemperatura() { ... }

    public void apaga() { ... }

    public ... ajustaTemperaturaDeseada(double tempDeseada)
                                   throws ...
    { ... }
}
```

Al invocar el método `ajustaTemperaturaDeseada` pueden producirse dos situaciones de error:

- La temperatura deseada está fuera del rango permitido.
- Se ha detectado un mal funcionamiento del dispositivo. En ese caso es fundamental apagar el dispositivo para evitar que continúe calentando o enfriando sin parar.

Responder a las siguientes cuestiones:

1. Indicar razonadamente qué mecanismo deberíamos utilizar para notificar cada uno de esos errores.
2. Escribir cómo quedaría la cabecera del método
3. Código correspondiente a la declaración de las excepciones utilizadas.
4. Suponiendo que es fundamental apagar el dispositivo después de su uso, ¿Qué construcción sería la más apropiada?