Introducción al manejo de Excepciones

Guía de Ejercicios

Descargá el proyecto **TP1-TP4.zip** del aula virtual que se encuentra incompleto y completá las clases según los siguientes ejercicios:

Ejercicio 1.

Escribí un pequeño programa que soporte la carga errónea de un número sin abortar (por ejemplo, ingresando un texto en vez de un número).

Para esto, creá un número e inicializalo en 0. Encerrá la carga, tal como se muestra en el material de lectura, dentro de un bloque ***try-catch-finally***. Informá por consola si hubo un error en la carga. Mostrá el valor del número antes y después de intentar la carga, haya sido exitosa o no. Para notar la diferencia, probalo siempre con valores distintos a 0 (cero).

Ejercicio 2.

Escribí una nueva versión del mismo programa, pero ahora, en vez de cargar el número como un string y transformarlo con *Integer.parseInt(scanner.nextLine())*, hacelo directamente con *scanner.nextInt()*, y acordate de *limpiar* el buffer de teclado luego de cargar el número para que no quede el ENTER en el buffer,
¿Qué sucede cuando se carga un valor no numérico?
Si hace falta, modificá el código para que el nuevo programa funcione exactamente igual que el otro.

Ejercicio 3.

Implementá el método *int pedirNumeroEntero(..)*, en la misma clase de prueba, el cual deberá procesar la carga por teclado de un entero, sin abortar. El método debe devolver el número entero ingresado por teclado.

Ejercicio 4.

Implementá las clases **RangoDeEnteros** y **LectorEnteros**, que usaremos para cargar números enteros por teclado y sin errores de ejecución:

RangoDeEnteros es una clase auxiliar que usaremos desde **LectorEnteros** para asegurarnos de que, cuando se solicite la carga de un valor dentro de un rango, este valor esté dentro del rango deseado. Su constructor recibirá dos valores enteros que serán los límites del rango. Ordenalos si fuese necesario, y guardalos en los atributos *limiteInferior* y *limiteSuperior*. Su método *incluye(int valor)* indica si el valor recibido está dentro del rango.

LectorEnteros permite cargar cualquier número entero de distintas maneras: sin validar su rango (aunque sí su tipo y evitando que el programa aborte), validando que esté dentro de un rango determinado y, por último, que esté dentro de un rango o que coincida con un valor que indique el fin de la carga. Su constructor recibe como parámetro un Scanner, el cual se usará para realizar las cargas por teclado. Este scanner no puede ser *null*; si lo es, el constructor debe devolver una excepción del tipo **IllegalArgumentException**.
Sus métodos públicos son:

- **int pedir(String mensaje)** , que recibe como parámetro el mensaje a mostrar y pide la carga de cualquier número entero, sin validarlo pero asegurándose de que la carga interrumpa la normal ejecución del programa.
- **int pedir(String mensaje, RangoDeEnteros rangoValido)** que recibe en *mensaje* que se quiere cargar (por ejemplo “Ingrese la nota del alumno”) y luego el *rango* numérico con el cual el número ingresado será validado. Debe controlar que ninguno de los parámetros recibidos sea *null* (si lo es debe devolver una excepción) y, en caso de que el mensaje esté vacío, utilizar el mensaje por defecto.

- **int pedir(String mensaje, int limiteA, int limiteB)** que recibe un *mensaje* que indica qué se quiere cargar (por ejemplo “Ingrese la nota de un alumno”) y el rango numérico dentro del cual el número será validado, incluyendo los límites (en este caso 0 y 10). El método debe complementar el mensaje recibido indicando el rango (“ entre A y B”), donde A es *limiteA* y B es *limiteB*.

Completá el programa Test para probar la clase anterior. Pedí primero la carga de un número entero cualquiera (incluso puede ser un entero negativo) y luego la carga de las fechas de nacimiento de una persona (desde 1900 hasta hoy) y la de su fallecimiento (desde la fecha de nacimiento hasta hoy, ó -1 si la persona aún vive). Para averiguar el año actual podés usar la siguiente instrucción:

```
int anioActual = calendar.getInstance().get(Calendar.YEAR);
```

o, si ya estás trabajando con Java 8....

```
int anioActual = Year.now().getValue();
```

Para terminar, luego de realizar la carga informá por consola la edad de la persona y si aún vive o no.

METODOS OPCIONALES de la clase **LectorEnteros**

- **int pedir()**, que no recibe parámetros y pide la carga de cualquier número entero con el mensaje “Ingrese un número entero cualquiera: ”.

- **int pedir(String mensaje, int limiteA, int limiteB, int valorFinCarga)**,

  que recibe un *mensaje* que indica qué se quiere cargar (por ejemplo “Ingrese la nota de un alumno”) y el rango numérico dentro del cual el número será validado, incluyendo los límites (en este caso 0 y 10). El método debe complementar el mensaje recibido indicando el rango (“ entre A y B”), donde A es *limiteA* y B es *limiteB*. Y recibe los límites “sueltos”.

- **int pedir(String mensaje, RangoDeEnteros rangoValido, int valorFinCarga)**, que es similar a los anteriores, pero agrega un nuevo parámetro que servirá para interrumpir la carga. El mensaje, siguiendo con el ejemplo anterior y siendo -1 el valor de *valorFinCarga* , sería “Ingrese la nota del alumno (entre 0 y 10, -1 para interrumpir la carga):”. En este caso también debe lanzar una excepción si el valor del rango es *null*.

Ejercicio 5.

Completá:

- la clase **Persona**, cuyo método booleano *vive()* no recibe parámetros y devuelve si la persona aún vive o no.
- la clase **Alumno**, que hereda de la clase **Persona** y tiene un único constructor parametrizado que controla directa o indirectamente todos sus valores, Si alguno de los valores recibidos es erróneo debe lanzar una excepción de tipo **IllegalArgumentException**.

  El legajo coincide con su número de DNI (investigá cuál es el rango válido para un DNI en nuestro país). Para validarlo, existe una instancia estática y pública de la clase **RangoDeEnteros** que no pueda modificarse (constante) y sirva para validar el número de documento tanto en la clase como en la carga externa. En la clase **Alumno** está de la siguiente manera (y suponiendo que el rango válido de números de documento va de 1000000-99999999)...

```
public static final RangoDeEnteros RANGO_NRO_DOCUMENTO = new
        RangoDeEnteros(1000000, 99999999);
```

Desde el programa principal escribí el código necesario para que se puedan cargar por teclado los datos para crear y mostrar una instancia de **Alumno**.