

Ejercicios excepciones

Ejercicio 1

Pedir un número decimal de tipo double por teclado (con la clase Scanner) o, si se introduce una letra, se volverá a pedir hasta que sea correcto.

NOTA: La excepción que lanza la clase Scanner cuando el usuario introduce un tipo de dato que no se corresponde con el leído es `InputMismatchException`.

Ejercicio 2

Igual que el anterior, pero cuando se haya introducido 5 veces mal, el programa terminará.

Ejercicio 3

Escribe un programa que pida números hasta que se introduzca un dato que no sea válido como número o hasta que se hayan dado 6 números. Después mostrará el resultado de sumar todos los números insertados.

Hay que usar al menos estos 2 métodos:

- Uno que pida una cadena.
- Otro que pida un número entero.

Si se produce un error en uno de estos métodos hay que dejar que se propague.

Ejercicio 4

Dada la clase Teclado (ver archivo adjunto).

Ahora en vez de utilizar la clase Scanner vas a utilizar la clase `BufferedReader` y además la vas a usar con los métodos de tal forma que no haya que instanciar un objeto de la clase.

Debe controlar mediante excepciones en todos los métodos, que el usuario introduce el tipo de dato que se pide. Cada vez que el usuario introduzca un tipo de dato que no es el que se pide, tendrá que volverse a pedir, hasta que introduzca uno correcto, y entonces devolver ese dato leído.

NOTA: El método `BufferedReader` lee por teclado los datos con el método `readLine()` y los convierte mediante las conversiones a entero, float, double, etc. (`Integer.parseInt()`, `Double.parseDouble()`, `Float.parseFloat()`, etc.).

Estos métodos lanzan la excepción `NumberFormatException` si lo que se le pasa por parámetro no se puede convertir a entero, float, double, etc. Tienes que capturar esta excepción para controlar que el dato introducido es correcto, de no ser así debe volver a pedir el dato.

Ejercicio 5

Crea una excepción personalizada llamada `EdadInvalidaException`, cuyo mensaje sea “No se puede tener una edad menor que 0”.

Crea una clase llamada `Persona` que solo tenga dos atributos: nombre y edad. Con un constructor que inicialice los dos atributos y con sus `setters` y `getters`.

Tanto en el constructor como en el método `setEdad(int edad)` debes controlar que no se introduzca una edad negativa. Para ello cuando se intente introducir una edad negativa (tanto en el constructor como en `setEdad`) se lanzará la excepción `EdadInvalidaException`.

Implementa también el método sobrescrito `toString()`.

En el método principal del programa instancia dos personas, una con edad negativa y otra con edad mayor que 0. A la segunda intenta asignarle después una edad negativa con el método `setEdad`.

Justo después de instanciar cada una de las dos personas y de llamar al método `setEdad` muestra por pantalla sus datos por medio del método `toString()`.

Ejercicio6

Crea una excepción que se llame `StringTooLongException`, cuyo constructor reciba como parámetro un mensaje y una cadena. Esta cadena será atributo de esta clase, por lo que también deberás crear el `setter` correspondiente.

Crea un método `compruebaLongitud(String cadena)` que compruebe si una cadena mide más de dos caracteres, en caso de que mida más, dicho método debe propagar la excepción.

Finalmente, en el `main`, pide una cadena y comprueba la longitud de la misma, gestionando de forma adecuada la excepción. En caso de que no mida dos caracteres, vuélvela a pedir. En caso de que mida dos caracteres, sácala por pantalla.

EJERCICIO 1

1. Crea una excepción personalizada llamada `EdadInvalidaException`, cuyo mensaje sea “No se puede tener una edad menor que 0”.
2. Crea una clase llamada `Persona` que solo tenga dos atributos: nombre y edad. Con un constructor que inicialice los dos atributos y con sus setters y getters. Tanto en el constructor como en el método `setEdad(int edad)` debes controlar que no se introduzca una edad negativa. Para ello cuando se intente introducir una edad negativa (tanto en el constructor como en `setEdad`) se lanzará la excepción `EdadInvalidaException`. Implementa también el método sobreescrito `toString()`.
3. En el método principal del programa instancia dos personas, una con edad negativa y otra con edad 0. A la segunda intenta asignarle después una edad negativa con el método `setEdad`. Justo después de instanciar cada una de las dos personas y de llamar al método `setEdad` muestra por pantalla sus datos por medio del método `toString()`.

EJERCICIO 2

1. Crea una excepción personalizada llamada `NoPlace`, cuyo mensaje sea: “No hay sitios disponibles”.
Crea una excepción personalizada llamada `Invalid`, cuyo mensaje sea: “No se puede”.
2. Crea una clase llamada `Bus`, con los atributos capacidad (inicializada a 0), capacidad máxima (serán los ocupantes que puede transportar), id de ruta.
3. En el método principal del programa se pide crear bus, incrementar ocupante, decrementar ocupante y salir. Si se decrementa un pasajero y la capacidad es 0, saltará la excepción `Invalid`. Si la capacidad es igual a la capacidad máxima e incrementamos pasajeros, saltará la excepción `NoPlace`.
4. Instancia tres buses, incrementar y decrementar pasajeros sobrepasando sus límites.

EJERCICIO 3

1. Crea una excepción personalizada llamada `InvalidId`, cuyo mensaje sea: “Identidad ya existe”.
Crea una excepción personalizada llamada `NoColor`, cuyo mensaje sea: “Color no disponible”.
Crea una excepción personalizada llamada `NoMarca`, cuyo mensaje sea: “Marca no disponible”.
2. Crea una clase llamada `Coche`, con los atributos id, color, marca y matricula.
 - ++ La id debe de ser única.
 - ++ color puede tomar el valor rojo, verde, azul, amarillo, blanco o negro.
 - ++ marca puede tomar el valor lancia, ford, nissan y toyota.
 - ++ matricula se genera de forma aleatoria 4 números del 0 al 9 y 3 letras mayúsculas.

3. En el método principal del programa se pide crear coche y ver listado de coches.
4. Instancia 10 coches de forma aleatoria.
String vColor = {'rojo','rosa','verde','amarillo','plateado','blanco','negro','cyan','gris','naranja','azul'};
String vMarca = {'ford','nissan','toyota','lancia','opel','dacia','renault','skoda','ferrari'};