Linkia FP

Formación Profesional Oficial a Distancia

Linkia FP

DAW – M03 – Clase 06

Control de excepciones

Linkia FP

Contenidos

- Excepciones
- Crear nuestras propias excepciones.

Excepciones

- Una excepción en términos de lenguaje de programación es la indicación de un problema que ocurre durante la ejecución de un programa.
- La palabra excepción se refiere a que este problema ocurre con poca frecuencia generalmente cuando existe algún dato o instrucción que no se apega al funcionamiento del programa por lo que se produce un error.
- El control de excepciones permite al programador controla la ejecución del programa evitando que éste falle de forma inesperada.

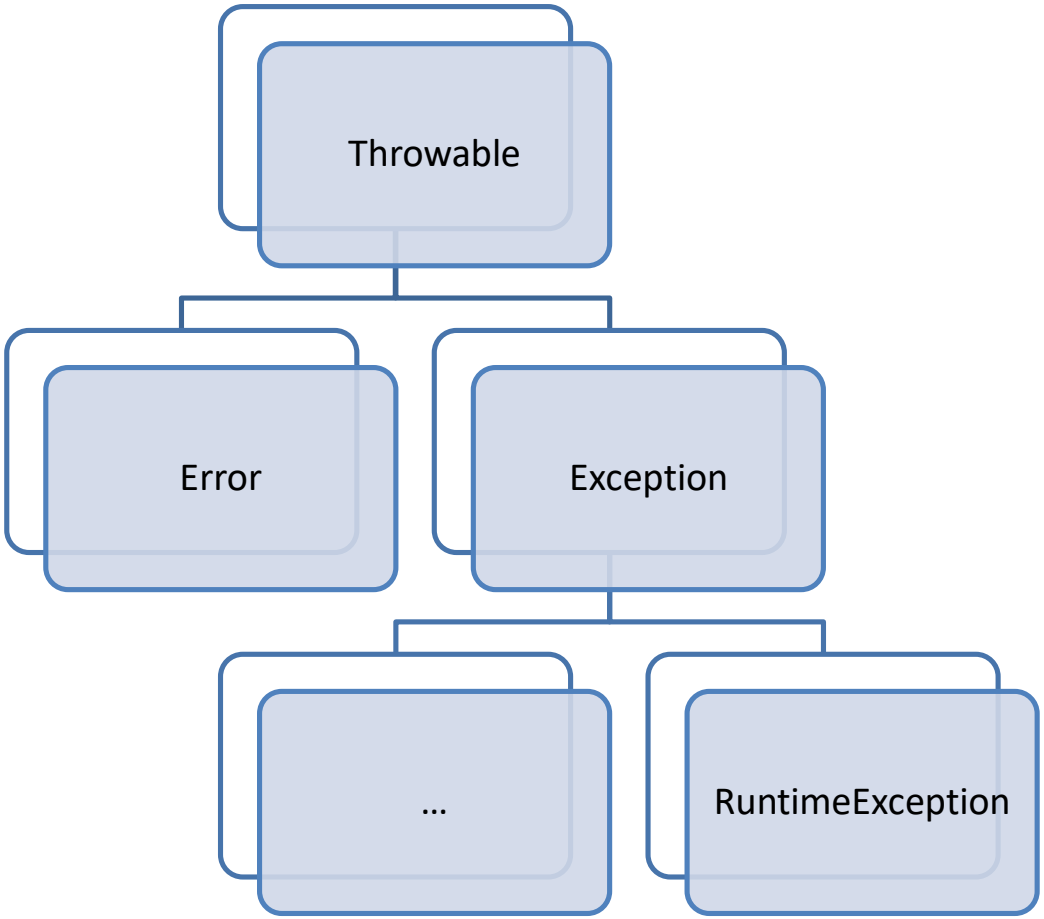
Estructura de una excepción

```
try {
      sentencias a proteger
} catch (expepcion_1) {
      control de excepción 1
}
…
} catch (excepcion_n) {
      control de excepción n
} finally {
      control opcional
}
```

Estructura de una excepción

- El programa intentará proteger las sentencias situadas dentro del bloque try.
- En el caso de que ocurra un error se intentará controlar la excepción mediante los bloques catch (dependiendo de la excepción se ejecutará un bloque de código u otro).
- El bloque finally es opcional, pero en caso de existir éste se ejecutará siempre.

Excepciones - jerarquía



https://docs.oracle.com/javase/7/docs/api/java/lang/Exception.html

Excepciones

- El lenguaje Java diferencia claramente entre tres tipos de excepciones: errores, comprobadas (checked) y no comprobadas (unchecked).
- La clase principal de la cual heredan todas las excepciones Java es Throwable.
- De ella nacen dos ramas: Error y Exception.

Excepciones

- La primera representa errores de una magnitud tal que una aplicación nunca debería intentar realizar nada con ellos (como errores de la JVM, desbordamientos de buffer, etc).
- La segunda rama, encabezada por Exception, representa aquellos errores que normalmente si solemos gestionar, y a los que comunmente solemos llamar excepciones.

Excepciones

- De Exception nacen múltiples ramas: ClassNotFoundException, IOException, ParseException, SQLException y otras muchas, todas ellas de tipo checked.
- La única excepción (valga la redundancia) es RuntimeException que es de tipo unchecked y encabeza todas las de este tipo.

Excepciones

- Una excepción de tipo checked representa un error del cual técnicamente podemos recuperarnos.
- Por ejemplo, una operación de lectura/escritura en disco puede fallar porque el fichero no exista, porque este se encuentre bloqueado por otra aplicación, etc.
- Todos estas situaciones, además de ser inherentes al propósito del código que las lanza (lectura/escritura en disco) son totalmente ajenas al propio código, y deben ser (y de hecho son) declaradas y manejadas mediante excepciones de tipo checked y sus mecanismos de control.

Excepciones

- Una excepción de tipo unchecked representa un error de programación.
- Uno de los ejemplos más tipicos es el de intentar leer en un array de N elementos un elemento que se encuentra en una posición mayor que N.
- El aspecto más destacado de las excepciones de tipo unchecked es que no deben ser forzosamente declaradas ni capturadas (en otras palabras, no son comprobadas). Por ello no son necesarios bloques try-catch ni declarar formalmente en la firma del método el lanzamiento de excepciones de este tipo.

Excepciones propias

- El programador puede crear sus propias clases de excepciones para tratar errores específicos que Java no contempla.
- Para ello hay que usar el concepto de herencia sobre la clase Exception.
- Normalmente, la nueva clase contendrá un único constructor con un parámetro de tipo String, que será el mensaje de error que queremos mostrar.
- Este mensaje lo mostraremos con el método getMessage() de la clase Exception

Linkia FP

Formación Profesional Oficial a Distancia