Fundamentos de Java

...

Elementos Avanzados

Contenido

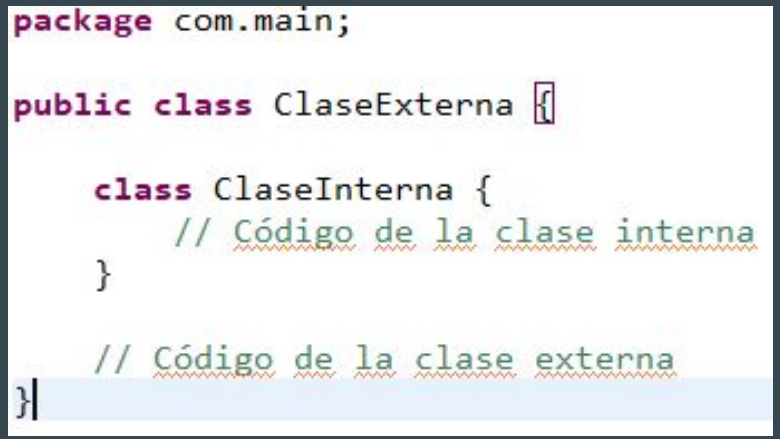
- Clases anidadas
- Clases abstractas
- Interfaces
- Excepciones:
  - Manejo de excepciones
  - Excepciones más comunes
  - Stacktrace en Java
  - Creación de nuestras excepciones
- Documentación con JavaDoc
  - Etiquetas

Clases Anidadas

Tal y como lo indica su nombre, es una clase definida dentro de otra. También se les conoce como clases internas.

Se usan para:

- Acceder a campos privados desde otra clase
- Ocultar una clase de otras dentro de un mismo paquete
- Crear clases internas anónimas
- Gestionar eventos y retrollamadas
- Acceder a los atributos ejemplares de otra clase



```
package com.main;

public class ClaseExterna {

    class ClaseInterna {
        // Código de la clase interna
    }

    // Código de la clase externa
}
```

Ejemplo

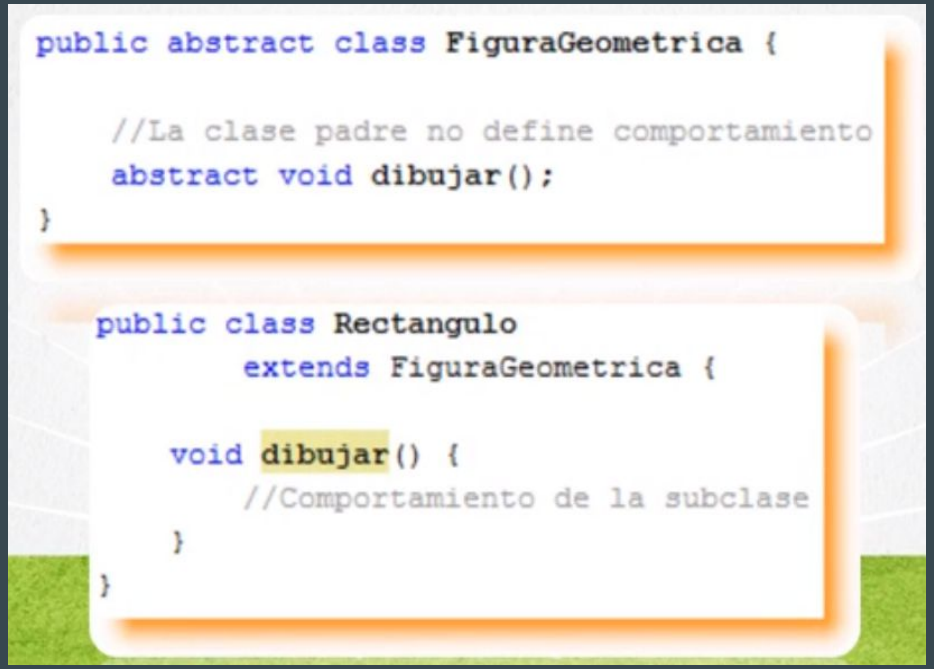
Clases Abstractas

Características:

- Definir una estructura
- Aplicar el polimorfismo
- Uso de la palabra reservada **extends**

Restricciones:

- No se pueden instanciar
- No definen comportamiento
- La funcionalidad la establece las clases hijas



```
public abstract class FiguraGeometrica {

    //La clase padre no define comportamiento
    abstract void dibujar();
}
```

```
public class Rectangulo
        extends FiguraGeometrica {

    void dibujar() {
        //Comportamiento de la subclase
    }
}
```

Ejemplo

Interfaces

Características:

- Declaración formal de un contrato
- Son abstractos, es decir, no poseen implementación
- Uso de la palabra reservada **implements**
- Podemos implementar múltiples interfaces a una clase

Estructura de una Interface

Definición de una interface en Java:

```
<modificadores>  interface <nombre_interface> [extends <interface padre>]
{
        <atributos>
        <métodos>
}
```

Uso de una interface en Java:

```
<modificadores> class <nombre_clase> [extends <superclase>] [implements
<interface1,interface2,etc>]
{
        <implementar_métodos_interface>
}
```

Ejemplo

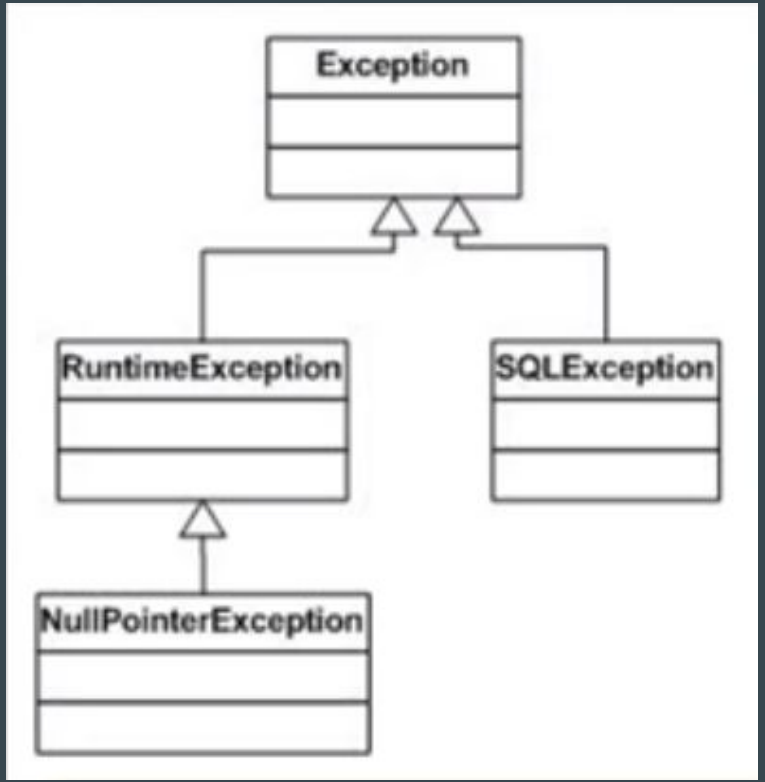
Excepciones

Check Exception

- Son revisadas por el compilador
- Es obligatorio tratarlas
- Ejemplo: SQLException

Unchecked Exception

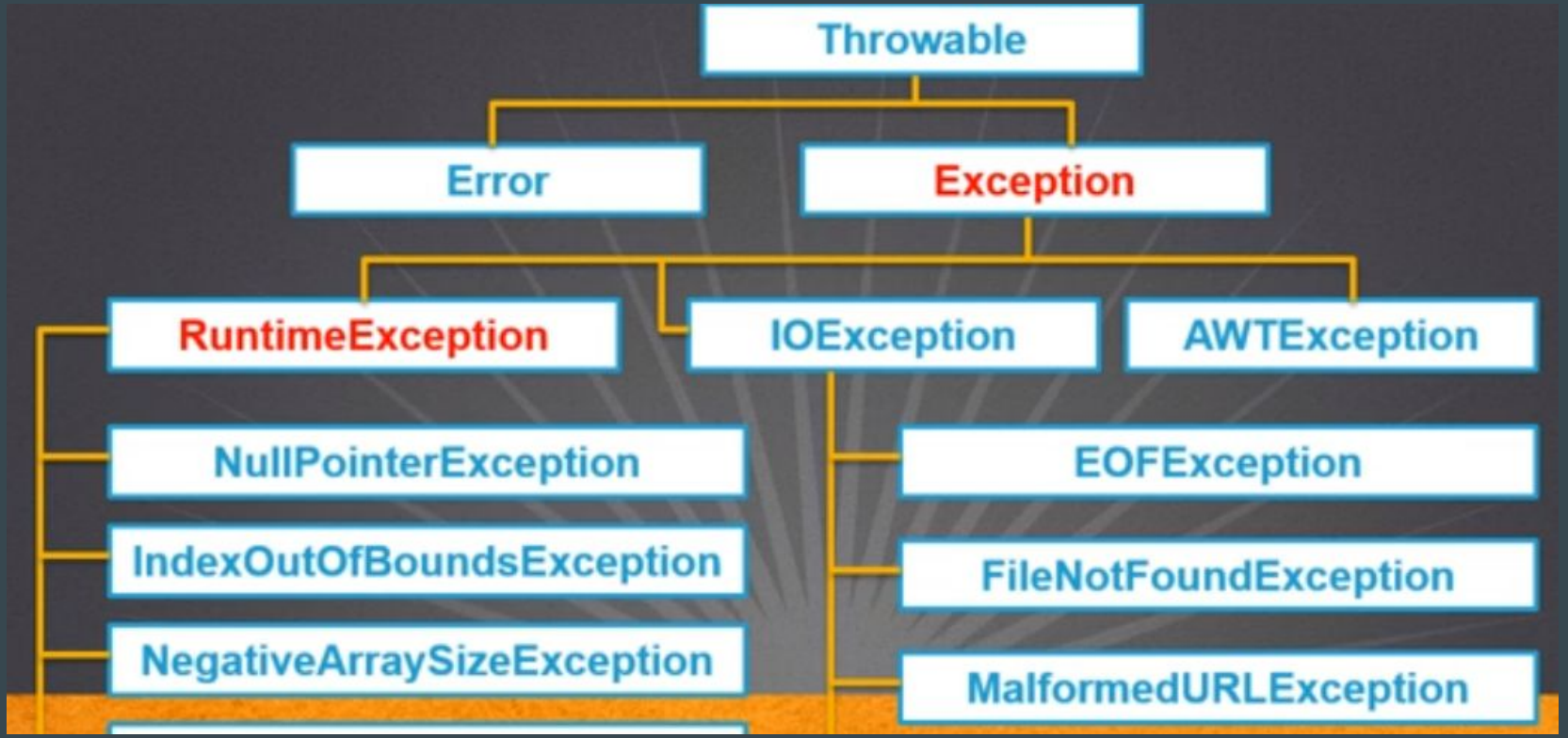
- No son revisadas por el compilador (tiempo de ejecución)
- Es opcional tratarla
- Ejemplo: NullPointerException



Manejo de Excepciones

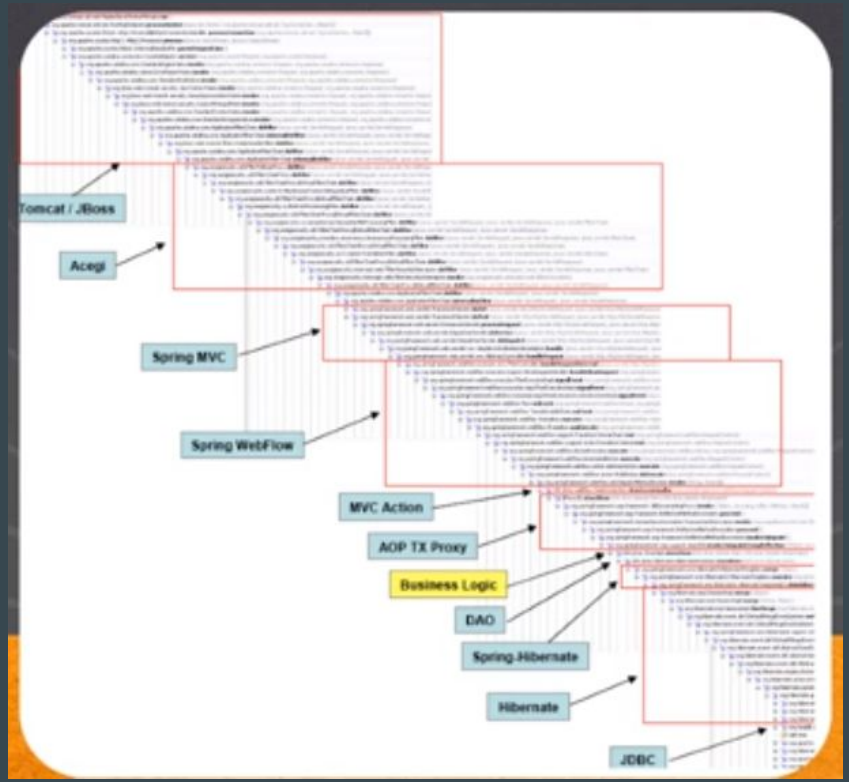
```
try {
    //do something
} catch (ExceptionType name) {

} catch (ExceptionType name) {

}  finally {
    //clean up
}
```

Excepciones más comunes



Stacktrace en Java

- Pilas de errores
- Traza del error del inicio al fin
- Flujo del error:
  - Método arroja un error
  - Si no atrapa la excepción, la propaga hasta que alguna clase la maneja
  - En caso de no manejarla el método main se satura
  - El programa se finaliza de manera anormal



Cláusula Throws

```java
public class ArrojarExcepcion {

  public void metodoX() throws Exception {
    throw new Exception("Mensaje de error");
  }
}
```

```java
public class TestArrojarExcepcion {

  public static void main(String args[]) throws Exception {
    ArrojarExcepcion ae = new ArrojarExcepcion();
    ae.metodoX();
  }
}
```

Creación de Nuestras Excepciones

```
public class MiExcepcion extends Exception{

  public MiExcepcion(String mensaje){
    super(mensaje);
  }
}
```

```
public class ArrojarExcepcion2 {

  public void metodoX() throws MiExcepcion {
    throw new MiExcepcion("Mi mensaje de error");
  }
}
```

Ejemplo

Documentación con JavaDoc

Características:

- Es un paquete de Java
- Genera automáticamente la documentación de nuestro programa
- Funciona a través de etiquetas propias y de HTML
- Crea un sitio web para nuestra documentación
- Es una buena práctica realizar la documentación de nuestro programa

Etiquetas para JavaDoc

Etiqueta	Descripción
@author	Nombre del autor del programa
@deprecated	Indica que el elemento es obsoleto, pertenece a versiones anteriores y no se recomienda su uso
@param	Definición de un parámetro de un método
@return	Descripción de que devuelve el método, no se usa en constructores o métodos **void**
@see	Indica que se asocia con otro método o clase
@version	Versión del método o clase
@excepcion	Nombre de la excepción más su descripción
@throws	Nombre de la excepción más su descripción

Ejemplo