

例外處理

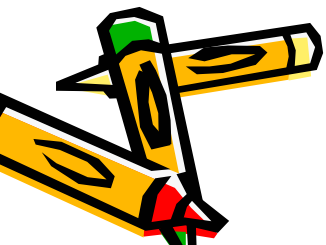
第十二章

Java2全方位學習



大綱

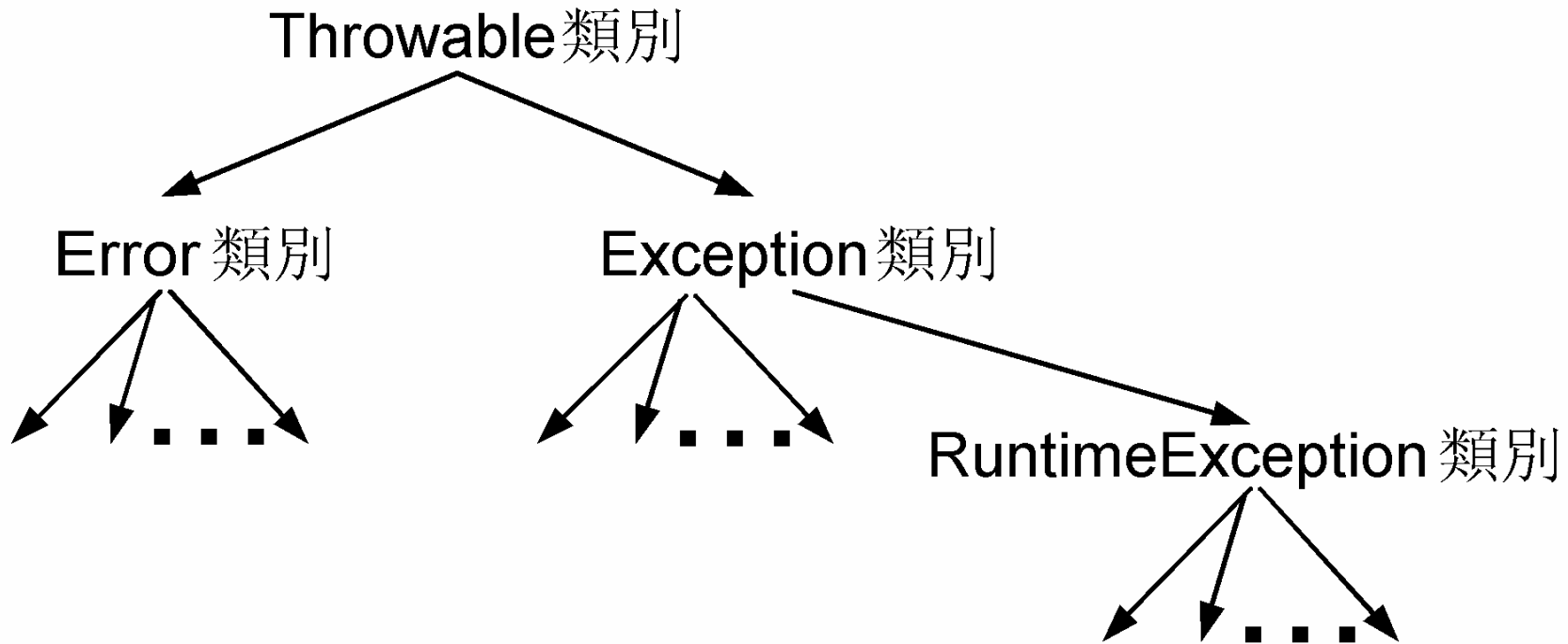
- 認識例外類別
- 使用例外處理機制
- 自行設計例外
- 例外與方法的覆寫



什麼事例外

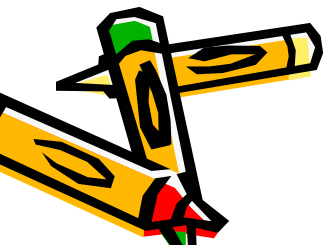
- 程式在執行時因為一些內在或外在的因素，導致程式中斷執行
- 例外處理機制
 - 增加程式的穩定度(robust)

例外類別架構



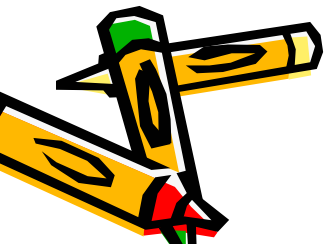
Error和Exception

- Error
 - 系統本身的錯誤
 - 無法控制
- Exception
 - 程式產生的錯誤
 - 可以控制



Exception種類

- RuntimeException
 - 撰寫程式時的疏忽
 - Java預設不會去處理
- Checked Exception
 - 除了RuntimeException之外的例外
 - 撰寫程式時需手動去處理

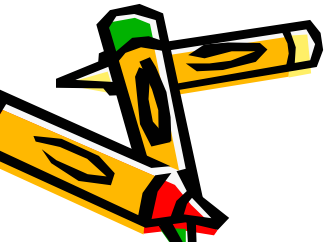


常見的RuntimeException

- ArithmeticException
- ArrayIndexOutOfBoundsException
- ArrayStroeException
- ClassCaseException
- IllegalArgumentException
- NullPointerException
- NativeArraySizeException
- SecurityException

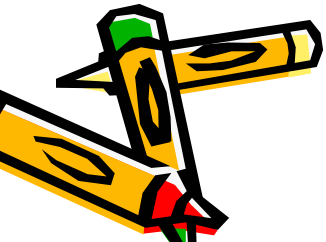
常見的Checked Exception

- ClassNotFoundException
- FileNotFoundException
- InterruptedException
- IOException
- SQLException



常見的Error

- OutOfMemoryError
- StackOverflowError
- AWTError
- ThreadDeath
- UnknowError

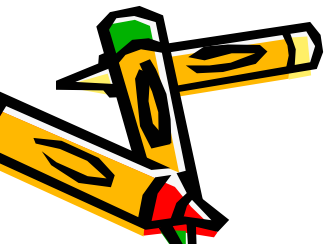


例外處理

```
try {  
    <可能會產生例外的程式碼>  
}  
catch (欲捕捉的例外類別名稱 物件)  
{  
    <例外處理程式碼>  
}
```

Throwable類別方法

- 取得例外的訊息
- toString()
- getMessage()
- printStackTrace()



finally

- 不論有沒有例外產生，都會去執行finally區塊裡面的程式碼
 - System.exit方法除外
- 若沒有捕捉到例外，也會執行finally區塊裡面的程式碼

例外捕捉順序

- 多個catch區塊
- 由上而下依序檢查
- 若產生的例外可以被自動轉型成catch想要捕捉的例外，那麼就會被處理
- 使用多到catch區塊時，需注意例外之間的繼承問題

throws

- 方法宣告時使用
- 可丟出多個例外

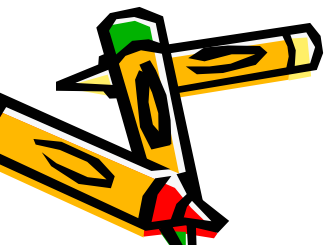


throw

- 產生例外(丟出例外物件)
- 陳述式
 - throw <例外物件>
 - 例：
throw IllegalArgumentException("Wrong number");

自訂例外

- 繼承某個Exception類別
- 撰寫建構子
 - 無參數
 - 字串參數
 - 呼叫父類別建構子
 - super



例外與方法的覆寫

- 兒子犯的錯不能比老爸還多
 - 不可丟出原有方法所丟出例外類別的父類別或上層類別
 - 丟出的例外類別數目不可以比原有的方法丟出的還多

Q&A

