

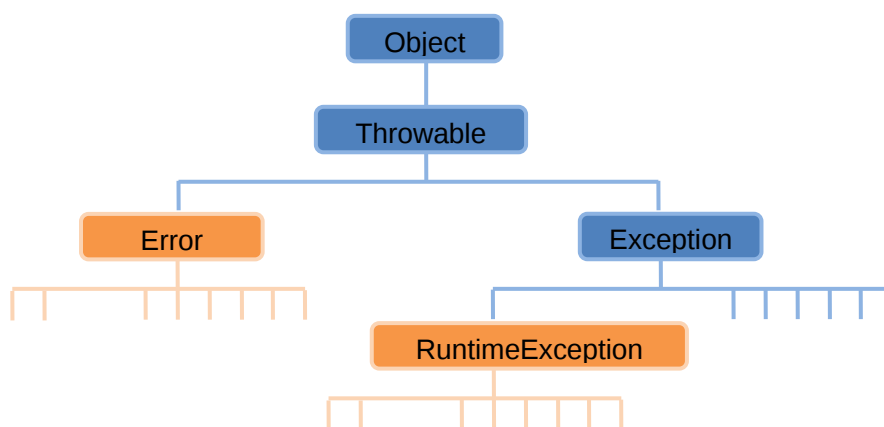
例外

例外とは

プログラムの実行中に発生するエラーです
例外は、

Throwable クラス、又はそのサブクラスのオブジェクト
で表現されます

[例外クラスの階層]



例外クラスのクラス階層と分類

Error クラスと
そのサブクラス

→ 続行不可能な例外
Virtual Machine のエラーなど

Exception クラスと
そのサブクラス

→ 実行時に対処する必要のある例外
I/O エラーなど

RuntimeException クラスと
そのサブクラス

→ 予め防げる例外
配列要素の数を超える代入など

例外クラスのタイプ

 以下 検査 (Checked) 例外 コンパイラは例外処理の約束を検査します

 以下 非検査 (Unchecked) 例外 // 検査しません

※例外処理の約束については 3 ページ目を見てください

例外処理

例外処理とは

例外に対して行われる処理です

宣言 (try 文)

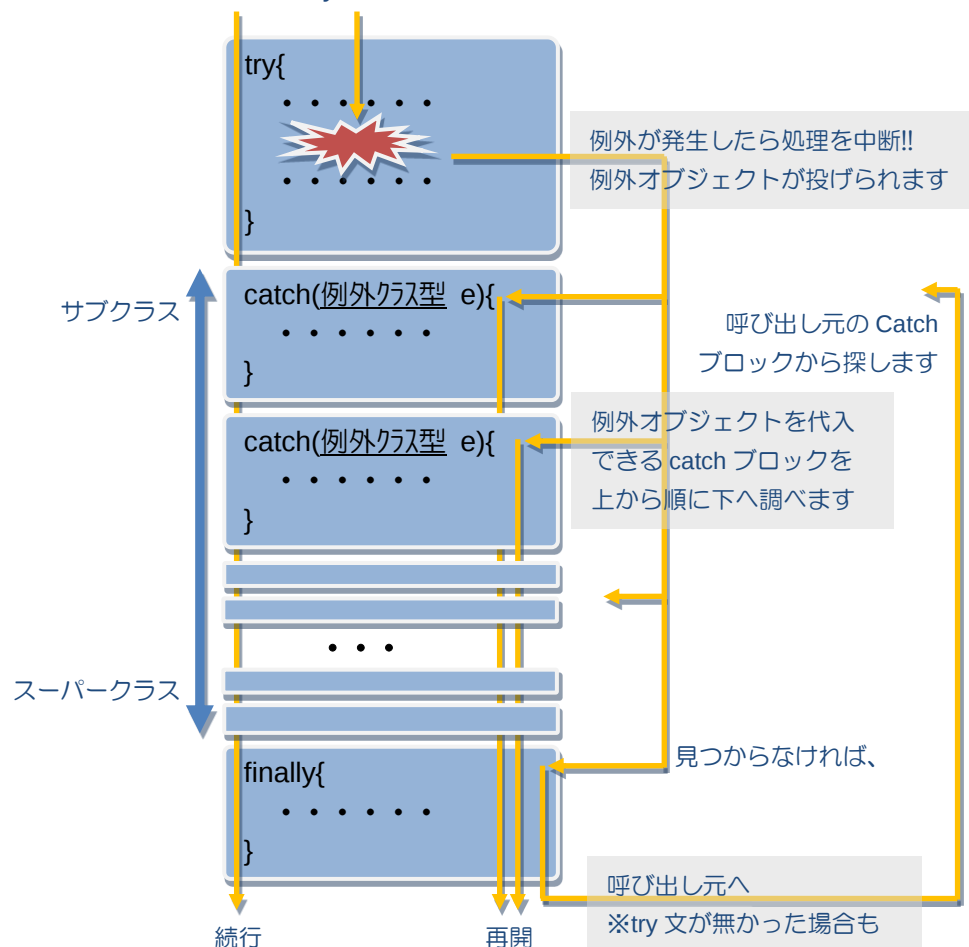
```
try{  
  文  
}catch(例外クラス型 1 e){  
  文  
}catch(例外クラス型 2 e){  
  文  
  :  
}finally{  
  文  
}
```

※ 1 つの try 文は、1 つの try ブロックと、少なくとも 1 つの catch ブロック又は finally ブロックをもちます

機能

- ①try ブロック 例外の発生を監視
- ②catch ブロック 例外が発生した場合、例外に対応する処理を実行
- ③finally ブロック try ブロックに入ったら
 - 例外の発生の有無に関わらず
 - break や return により try ブロックを出る場合try 文を抜ける時に必ず実行

処理の流れ



例外の送出

例外の送出とは 例外を発生させることです

手 順

1. Exception クラスを継承し各自の例外クラスを宣言します
※一般に Throwable クラスを継承していればよいです
また、既存の例外クラスを用いてもよいです
2. オブジェクトを生成します
3. throw オブジェクトを指す変数; により例外を発生させます

throws 宣言

throws 宣言とは メソッドが例外を発生する可能性があることを明示します

例外の送出を行うメソッド、例外処理を記述していない例外や
例外処理では catch していない例外を発生する可能性のあるメソッド
にこれを宣言します

宣 言

戻り値の型 リスト 名(引数リスト) throws 例外クラス 1, 例外クラス 2,...{
本 体
}

※例外クラス は例外に対応するクラスのスーパークラスでもよいです

例外処理の約束

あるメソッドA内で例外を送出するメソッドB（やコマンド）を実行する場合、
基本的に次の

1. try 文によりメソッドA内でメソッドBの例外を処理してしまいます

→ メソッドAの利用者は例外を気にせずにコードを作成できます

2. メソッドAに throws 宣言をします

→ メソッドAの使用者にこのメソッドは例外が発生することを伝えます

のどちらかをします

検査例外の場合 → コンパイラがこの約束をチェックします

例えば、
IOException（キーボードなどの入力に関する例外）
や Exception を継承した独自の例外クラスなど

非検査例外の場合 → コンパイラはこの約束をチェックしません
例外への対応はプログラマに任されます

例えば、
ArrayIndexOutOfBoundsException（配列範囲超え）
ArithmeticException（ゼロでの除算）など

throws 宣言をもつメソッドのオーバーライドの約束

オーバーライドの機能を妨げないように新しいメソッドの throws 宣言を次のようにします

オーバーライドする側のメソッドAは、オーバーライドされる側（スーパークラス又はインタフェース）のメソッドBが送出する例外のクラスとそのサブクラスを送出できます

理由は、オリジナルのメソッドBを使うプログラムでは、メソッドBが送出する例外に対応する例外処理しかないためです

継承の場合の例 1 :

void A() throws E1



void A() throws E2

例外クラス E2 は、
「例外クラス E1 と同じ、又はそのサブクラス」
とします

継承の場合の例 2 :

void A() throws E1, E2



void A() throws E3

例外クラス E3 は、
「例外クラス E1 または E2 と同じ、又はそれらのサブクラス」
とします

実装の場合の例 :

void A() throws E3 void A() throws E1, E2

例外クラス E3 は、
「例外クラス E1 または E2 と同じ、又はそれらのサブクラス」
とします