Ｊａｖａプログラミング Ⅱ

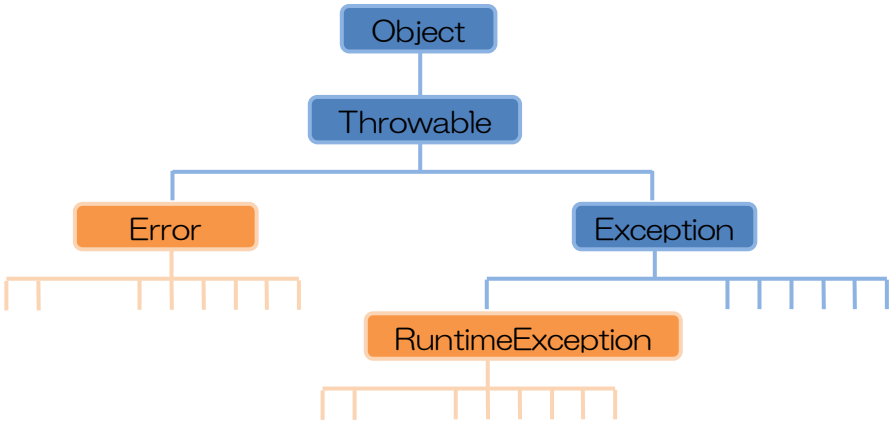
９回目　例外処理

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　Checked 例外　　コンパイラは例外処理の約束をチェックします

以下　Unchecked 例外　　〃　　チェックしません

※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）

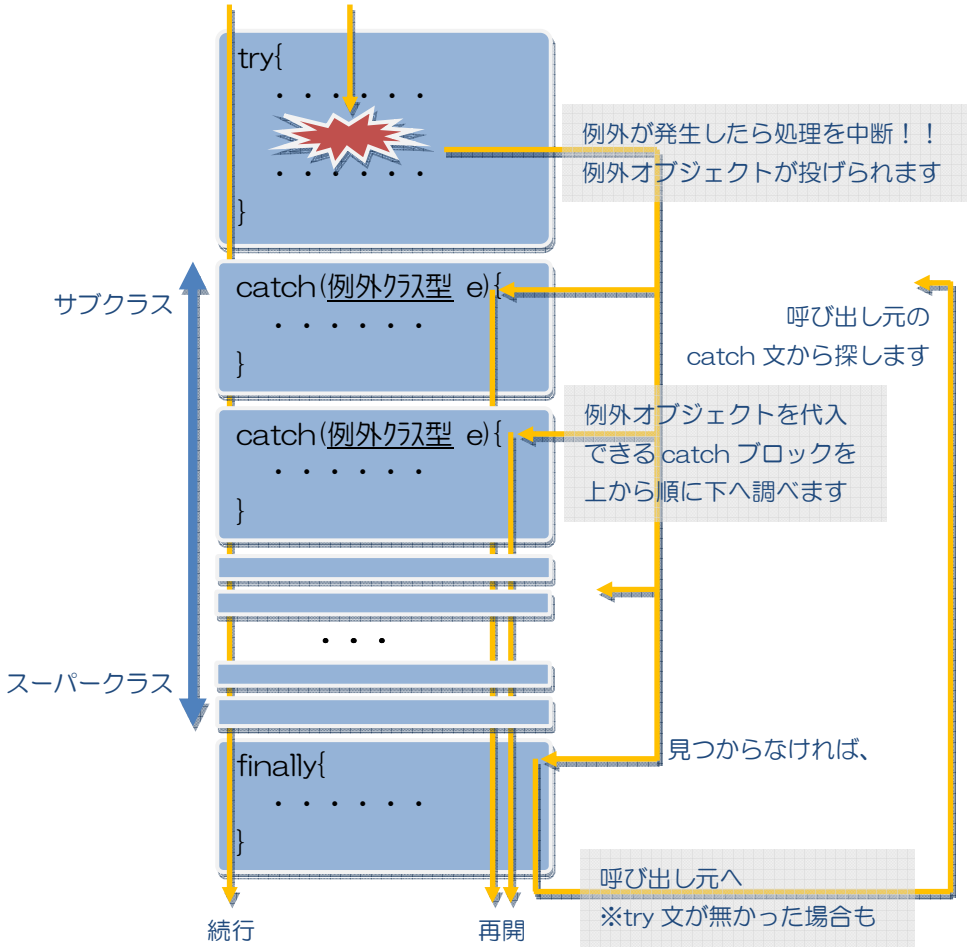
```
try{
   文
}catch(例外ｸﾗｽ型 １　e){
   文
}catch(例外ｸﾗｽ型 ２　e){
   文
   :
}finally{
   文
}
```

※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 ブロックを伴う

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　例外処理を記述していない例外、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
※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

宣　　言　戻り値の型　メソッド名(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

Checked 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

Unchecked 例外の場合　→ コンパイラはこの約束をチャックしません
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 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