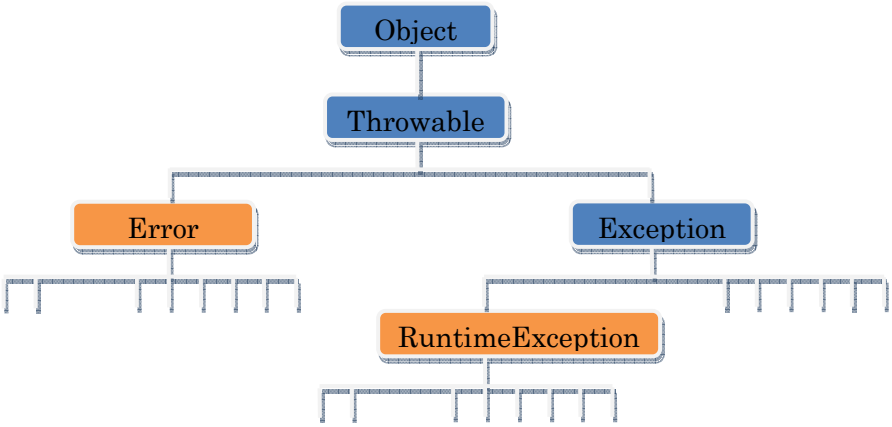
Ｊａｖａプログラミング Ⅱ

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
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

[例外クラスの階層]



例外クラスの意味

Error クラス以下　→　Virtual Machine のエラーなど続行不可能な例外

Exception クラス以下　→　I/O エラーなど実行時に対処する必要のある例外

RuntimeException クラス以下　→　配列要素の数を超える代入など予め防げる例外

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する

以下　Unchecked 例外　　〃　　チェックをしない

※例外処理の約束については３ページ目

例外処理

例外処理とは　　例外に対して行われる処理

宣　言（try 文）

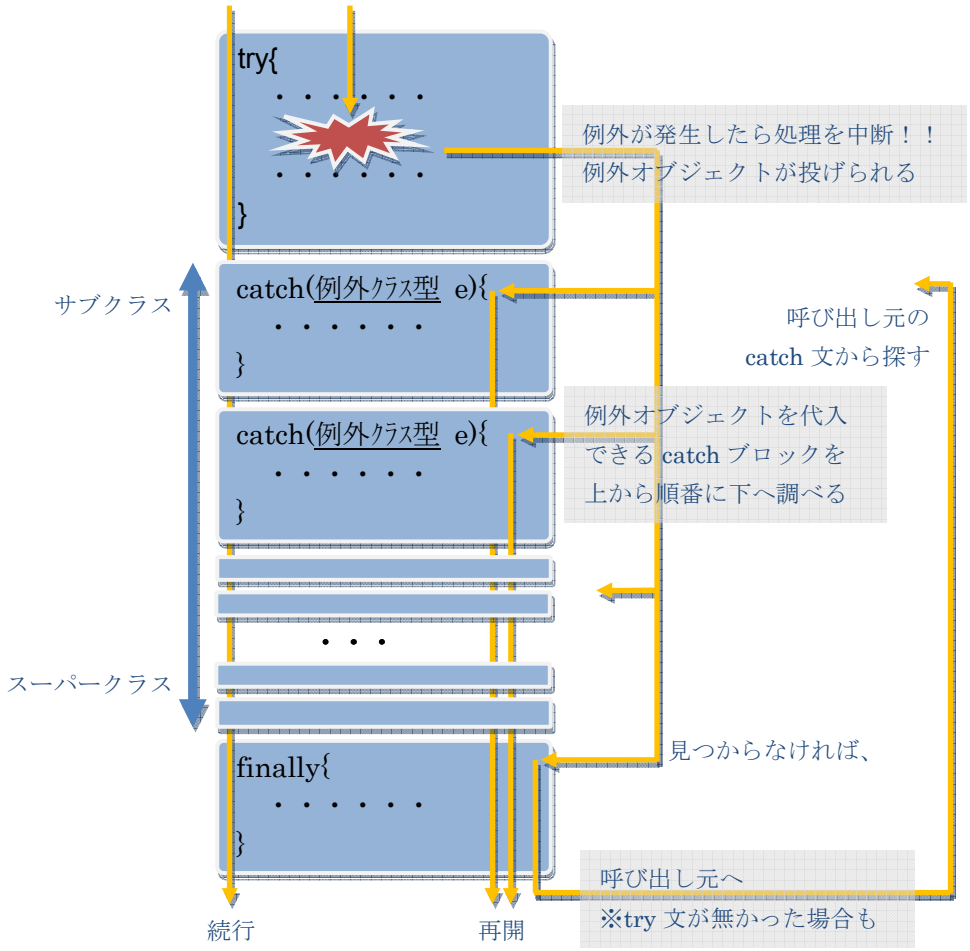
```
try{
   文;
}catch(例外ｸﾗｽ型 1　e){
   文;
}catch(例外ｸﾗｽ型 2　e){
   文;
   :
}finally{
   文;
}
```

※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も
try 文を抜ける時に必ず実行される

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

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

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する
※メソッドが例外を発生する可能性があることを明示する

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

例外処理の約束

例外を送出するメソッドやコマンドを実行する場合、
基本的に次の
１．try 文によりそのメソッド内で例外を処理してしまう
→ 使用者は例外を気にせずにコードを作成できる
２．throws 宣言をする
→ 使用者に例外が発生する可能性があることを伝える
のどちらかをする

Checked 例外の場合 → コンパイラがこの約束をチェックする
Unchecked 例外の場合 → プログラマに任される

throws 宣言をもつメソッドのオーバーライド

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 宣言を次のようにする

継承の場合（単一）:

```
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」
とする

継承と実装を組み合わせた場合（多重）：

void　A( ) throws E1　と　void　A( ) throws E2

void　A( ) throws E3

例外クラス E3 は、
「例外クラス E1 と同じ、又はそのサブクラス」
かつ
「例外クラス E2 と同じ、又はそのサブクラス」
とする