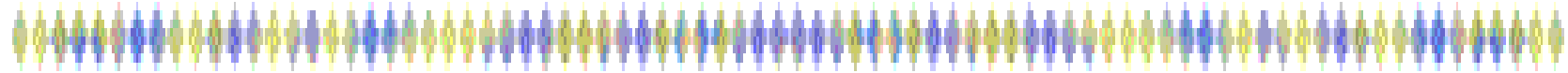


第8回 Javaゼミ



8.1 例外処理

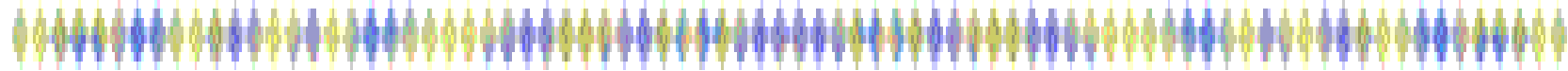
8.2 catchブロックの検索

8.3 throwステートメント

発表日 : 2007/6/8

発表者 : 阿部 竜之介

例外



例外とは

■プログラムの実行中に発生した問題を通知するために、実行時に生成されるオブジェクトのこと

例外が発生するケース

- 整数を0で除算した場合
- 配列のインデックスが負の値または配列の領域を超える値だった場合
- ファイルが見つからなかった場合
- 数値の形式が不正だった場合 ...など

既定の例外処理(例外ハンドラ)

例として整数値を0で除算し、故意に算術例外を発生させる。その例外は既定の例外処理メカニズムによって処理される。

```
class DBZ {  
    public static void main( String args[] ) {  
        System.out.println( 1 / 0 );  
    }  
}
```

上記のプログラムを実行すると以下のようなメッセージが表示される

```
Exception in thread "main" java.lang.ArithmeticException: / by zero  
    at DBZ.main(DBZ.java:3)
```

例外に関する情報が表示され、アプリケーションの実行は停止する(既定の例外処理)。

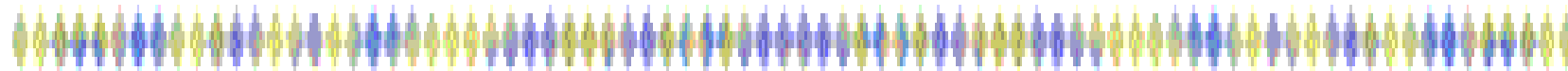
例外処理(1)

- ◆ Java言語ではプログラムの実行中に発生した例外を、ユーザの用意した処理ルーチンで処理することができる
- ◆ 以下の構文を使用する

```
try {  
    // tryブロック  
}  
catch ( ExceptionType param ) {  
    // 例外処理ブロック  
}  
.  
.  
.  
finally {  
    // finallyブロック  
}
```

例外処理

例外処理(2)



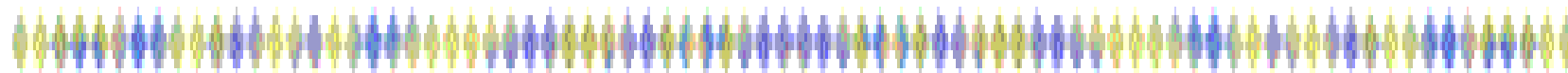
tryブロック

- ここに例外の発生を監視する必要があるコードを記述
- このコードの実行中に問題が発生した場合、例外が投げられる
- 少なくとも1つのcatchブロックかfinallyブロックが必要

catchブロック

- 各種の例外を処理する
- tryブロックの直後に、いくつかのcatchブロックが続く
- catchブロックの先頭のcatchステートメントでは引数(発生した問題に関する情報を格納した例外オブジェクト)が受け取られる
- パラメータの型がjava.lang.Throwableオブジェクトでなければエラーになる

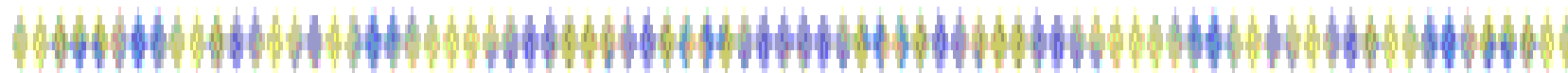
例外処理(3)



問題発生時の処理の流れ

- tryブロック内のコード実行時、問題が発生すると、JVMではtryブロックの実行を停止し、その種類の例外を処理できるcatchブロックの検索をする
- tryブロックの残りの処理は実行されない
- 検索は最初のcatchブロックから開始される
- 例外オブジェクトの種類がcatchブロックのパラメータと一致したら、その処理が実行される
- 一致しなければ次のcatch節に検索が進む
- catchブロックの実行が終わった時点で、finallyブロックに移行

例外処理(4)



finallyブロック

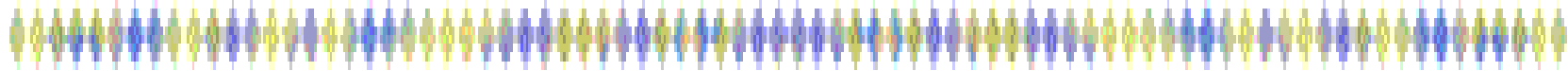
- 必ず実行されるブロック
- tryブロックの処理が正常に終了すると、finallyブロックが実行される
- tryブロック内にreturnステートメントがあったとしても、呼び出し元に戻る前に必ずfinallyブロックが実行される
- 省略可能

catchブロックの検索

catchブロックの検索の流れ

- 例外が発生したtryブロックの直後にある1つ目のcatchブロックから検索が開始され、例外オブジェクトの型とcatchブロックのパラメータの型が一致するまで継続される
- 現在のメソッド内に該当するcatchブロックがなければ、呼び出しメソッドが検索される
- main()メソッドまでさかのぼっても該当するcatchブロックが見つからない場合は既定の例外ハンドラが呼び出され、例外メッセージが表示されてプログラムの実行が停止する
- 検索が現在のメソッドから先に進む前に必ずfinallyブロックが実行される

catchブロックの検索例(1)

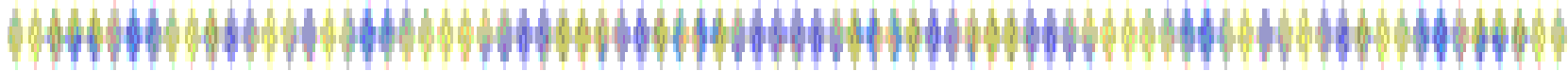


catchブロックの検索の例

```
class CatchSearch {
    public static void main(String args[]) {
        try {
            System.out.println("Before a");
            a();
            System.out.println("After a");
        }
        catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("main: " + e);
        }
        finally {
            System.out.println("main: finally");
        }
    }
}
```

```
public static void a() {
    try {
        int array[] = new int[4];
        array[10] = 10; //例外の発生
    }
    catch (ArithmeticException e) {
        System.out.println("a: " + e);
    }
    finally {
        System.out.println("a: finally");
    }
}
```

catchブロックの検索例(2)



実行結果

```
Before a  
a: finally  
main: java.lang.ArrayIndexOutOfBoundsException: 10  
main: finally
```

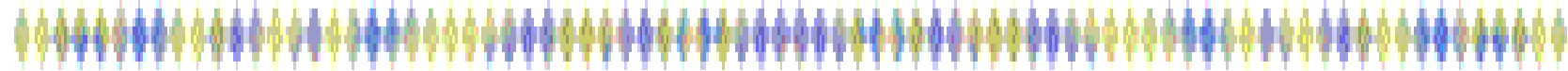
ArrayIndexOutOfBoundsException

配列のインデックスが存在しない要素を指している

ArithmeticException

算術例外が発生した(整数を0で除算した場合など)

throwステートメント(1)



throwステートメント

- 例外をプログラム内で明示的に生成するのに用いる

```
throw object;
```

throwステートメント

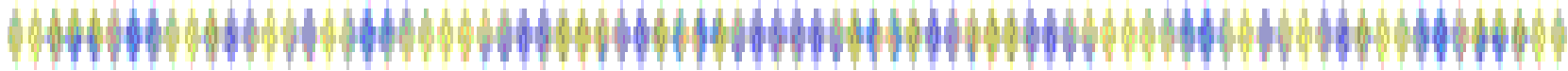
※objectはjava.lang.Throwable型でなければならない。

- catchブロック内部では、引数として受け取った例外オブジェクトを投げることができる

```
catch ( ExceptionType param ) {  
    ...  
    throw param;  
    ...  
}
```

引数で受けた例外オブジェクトの投入

throwステートメント(2)



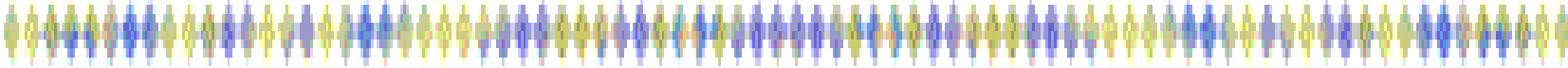
- 新しい例外オブジェクトを作成して投げるには次の構文を使う

```
throw new ExceptionType ( args );
```

新しい例外オブジェクトの投入

- 処理がthrowステートメントに進むと、該当するcatchブロックの検索が開始される
- 同じtryまたはcatchブロック内の残りの処理は実行されない

練習問題



コマンドラインから複数の引数を受け取り、その引数をそれぞれint型に変換してから表示するプログラムを作成せよ。ただし、整数ではない引数を受け取り、問題が発生した場合、catchブロックで「整数ではありません」、finallyブロックで「処理終了」と表示するように作成すること(問題が発生しなくても「処理終了」は表示される)。

※発生する例外はNumberFormatException