

例外処理

例外とは、プログラムの実行時に配列で宣言した数以上の配列番号を参照しようとしたり、整数を0で割って無権大となって、処理出来なくなったりする予期しないエラーの事を指します。

Javaでは、予めそれらが生じる場所は、検知する処理を行い、エラーが生じても対処できるようにプログラムを作ります。

一般的な例外

Java 言語では、あらかじめある条件において発生する例外がいくつか定義されています。主な例外は以下の通りです。

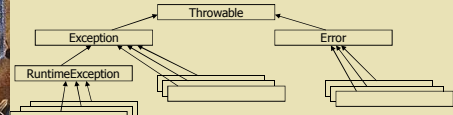
ArithmeticException
整数をゼロで割った結果発生するもの。

NegativeArraySizeException
添字が負になるような配列を獲得しようとしたときに発生するもの。

ArrayIndexOutOfBoundsException
獲得している配列の範囲外をアクセスしたときに発生するもの。

Throwable クラスとそのサブクラス

例外用クラスとして `java.lang.Throwable` が提供されており、このクラスを継承した(機能を引き継いだ)重要なクラスが3つあります。それぞれのクラス階層は以下のとおりです。



Throwable クラス
例外処理用に提供されているクラスのスーパークラス。これは直接利用できないクラスである。

Error クラス
メモリ不足などのハード的エラーが発生した際の処理を記述する。

Exception クラス
この例外クラスは多くのサブクラスを持つ、通常はこのクラスを利用する。

RuntimeException クラス
プログラム実行時に発生した例外の処理を記載する。

try - catch ブロック

try ブロック

本文中で、例外が発生する可能性のある箇所をブロック指定する。

catch ブロック

例外発生時の対処を記述する。例外のタイプによって使用される `catch` ブロックが決定される。
`catch` ブロックは上から順番に参照される。

finally

常に実行されるブロック。

```
try {  
    obj.func();  
} catch (例外用クラス a) {  
}  
} catch (例外用クラス b) {  
}  
} finally {  
}
```

throws

`throw` が実行されるメソッド内で例外処理をするのではなく、そのメソッドの呼び出し元の方で処理をさせたい場合に `throws` を使用します。
`throw` を使用したメソッドの宣言に指定します。

```
try {  
    :  
    troublesome( ); //troublesome() メソッドを呼出す側で例外処理を記述  
    :  
} catch (IOException e) {  
    :  
    :  
}  
  
public void troublesome( ) throws IOException { //throws で発生される  
    : //例外の型を指定  
    IOException err = new IOException();  
    throw err;  
    :  
}
```

参考 - 対象となる例外が複数ある場合は、カンマで区切って指定します。

ユーザ定義の例外

提供されている例外の中に使用したい例外がない場合、ユーザが独自の例外用クラスを作成することもできます。

例外は、`Throwable` クラスを継承したオブジェクトだけを `throw` することができます。

ユーザ定義の例外の定義方法は以下のとおりです。

```
public class クラス名 extends Exception {  
    :  
}
```

参考 - `Exception` クラスは、`Throwable` クラスを継承しています。

演習：例外処理 (先週の例題)

```
import java.io.*;  
class KeyInput {  
    public static void main(String args[]) {  
        String result = ""; int number=0;  
        try{//キーボードを利用して入力する先はこの  
            BufferedReader br = new BufferedReader(new  
                InputStreamReader(System.in));  
            System.out.print("点数を入力してください => ");  
            //キーボードからの文字列を int型に変換して numberへ保存する  
            number = Integer.parseInt(br.readLine());  
        }catch(IOException e){  
            System.out.println("IOException :"+ e);  
        }  
        if (number > 90) { result = "S"; }  
        else if(number > 80) { result = "C"; }  
        else { result = "D"; }  
        System.out.println("試験の結果:" + result);  
    }  
}
```

数字を入力する箇の部分で、文字が入力されるとエラーになる

```
c:\Users\User> cd c:\src  
c:\src> notepad KeyInput.java  
c:\src> javac KeyInput.java  
c:\src> java KeyInput
```

演習：例外処理 (対応後)

```
import java.io.*;
class KeyInput {
    public static void main(String args[]) {
        String result = ""; int number=0;
        try{//キーボードを利用して入力する先はこのような書式で書く
            BufferedReader br = new BufferedReader(new
                InputStreamReader(System.in));
            System.out.println("点数を入力してください => ");
            //キーボードからの文字列を int型に変換して numberへ保存する
            number = Integer.parseInt(br.readLine());
        }catch(IOException e){
            System.out.println("IOException :"+ e);
        } catch (NumberFormatException e) {
            System.out.println("引数を整数で入力してください。");
        }
        if (number > 90) { result = "S"; }
        else if(number > 80) { result = "C"; }
        else { result = "D"; }
        System.out.println("試験の結果:" + result);
    }
}
```

エラーが起きても、それに対応したプログラムに変更する

```
c:\Users\user> cd c:\src
c:\src>notepad KeyInput.java
c:\src>javac KeyInput.java
c:\src>java KeyInput
```

演習：例外処理 (対応後)

```
import java.io.*;
class KeyInput {
    public static void main(String args[]) {
        String result = ""; int number=0;
        try{//キーボードを利用して入力する先はこのような書式で書く
            BufferedReader br = new BufferedReader(new
                InputStreamReader(System.in));
            System.out.println("点数を入力してください => ");
            //キーボードからの文字列を int型に変換して numberへ保存する
            number = Integer.parseInt(br.readLine());
        }catch(IOException e){
            System.out.println("IOException :"+ e);
        } catch (NumberFormatException e) {
            System.out.println("引数を整数で入力してください。");
        } catch (Exception er) {
            System.out.println("次のメッセージを技術員にお伝え下さい。"+er);
        }
        if (number > 90) { result = "S"; }
        else if(number > 80) { result = "C"; }
        else { result = "D"; }
        System.out.println("試験の結果:" + result);
    }
}
```

エラーが起きても、それに対応したプログラムに変更する

```
c:\Users\user> cd c:\src
c:\src>notepad KeyInput.java
c:\src>javac KeyInput.java
c:\src>java KeyInput
```

課題3：コンソールから5人のデータを入力して平均を計算するプログラムに例外処理を加えなさい

```
class Input2 { /* 学生の点数をコンソールから入力して6名の平均値を表示する*/
    public static void main(String args[]) {
        int score[] = new int [5];
        score[0] = Integer.parseInt(args[0]); // 1人目の点数
        score[1] = Integer.parseInt(args[1]); // 2人目の点数
        score[2] = Integer.parseInt(args[2]); // 3人目の点数
        int average = (score[0]+score[1]+score[2])/3;
        System.out.println("1人目の点数は"+score[0]+"点でした");//入力点表示
        System.out.println("2人目の点数は"+score[1]+"点でした");//入力点表示
        System.out.println("平均は "+average+"点でした");//平均点表示
    }
}
```

次のページへ続く

これを基にして、5名の平均点の計算をしなさい。ただし、データはコマンドプロンプトから入力する。左と結果表示が完全に一致するようにしなさい。

```
c:\src>notepad Input2.java
c:\src>javac Input2.java
c:\src>java Input2
c:\src>java Input2 70 aa 90 70 90
1人目の点数は70点でした
2人目の点数が数字ではありませんでした。
再度プログラムを起動して入力して下さい。
```

課題3：コンソールから5人のデータを入力して平均を計算するプログラムに例外処理を加えなさい

```
class Input2 { /* 学生の点数をコンソールから入力して6名の平均値を表示する*/
    public static void main(String args[]) {
        int score[] = new int [5];
        score[0] = Integer.parseInt(args[0]); // 1人目の点数
        score[1] = Integer.parseInt(args[1]); // 2人目の点数
        score[2] = Integer.parseInt(args[2]); // 3人目の点数
        int average = (score[0]+score[1]+score[2])/3;
        System.out.println("1人目の点数は"+score[0]+"点でした");//入力点表示
        System.out.println("2人目の点数は"+score[1]+"点でした");//入力点表示
        System.out.println("平均は "+average+"点でした");//平均点表示
    }
}
```

これを基にして、5名の平均点の計算をしなさい。ただし、データはコマンドプロンプトから入力する。左と結果表示が完全に一致するようにしなさい。

```
c:\src>notepad Input2.java
c:\src>javac Input2.java
c:\src>java Input2
c:\src>java Input2 70 80 90 70 90
1人目の点数は70点でした
2人目の点数は80点でした
3人目の点数は90点でした
4人目の点数は70点でした
5人目の点数は90点でした
平均は 80点でした
```

課題4：配列で5人の平均を計算するプログラム

```
class Score2 { /* 学生5名の点数の平均値を表示する*/
    public static void main(String args[]) {
        int score[] = new int [2];
        int average =0;
        for(int i=0; i<2; i++){
            score[i] = ここは自分で考える ;
            average = average + score[i];
        }
        System.out.println("平均は"+average/5+"です");
    }
}
```

次のページへ続く

```
c:\src>notepad Score2.java
c:\src>javac Score2.java
c:\src>java Score2
c:\src>java Score2 70 80 90 aa 90
入力が整数では無いデータがありました。
再度プログラムを起動して入力して下さい。
```

課題4：配列で5人の平均を計算するプログラム

```
class Score2 { /* 学生5名の点数の平均値を表示する*/
    public static void main(String args[]) {
        int score[] = new int [2];
        int average =0;
        for(int i=0; i<2; i++){
            score[i] = ここは自分で考える ;
            average = average + score[i];
        }
        System.out.println("平均は"+average/5+"です");
    }
}
```

```
c:\src>notepad Score2.java
c:\src>javac Score2.java
c:\src>java Score2
c:\src>java Score2 70 80 90 70 90
5人の合計は 400点でした
5人の平均は 80点でした
```