

15 例外処理

【フローチャートあり】

問題 15_1

Python / Java

2つの整数を入力してもらい、割り算を実行した結果を以下の実行例を参考に画面に表示するプログラムを作成しなさい。ただし、0で割った場合には「0 除算例外」を補足してメッセージを画面に表示しなさい。

【実行例 1】0 以外の数で割ったとき

整数 1 = 10

整数 2 = 2

10 / 2 = 5.0

処理終了

【実行例 2】0 で割ったとき

整数 1 = 10

整数 2 = 0

0 による割り算です！！

処理終了

【フローチャートあり】

問題 15_2

Python / Java

ユーザに整数を入力してもらい、偶数か奇数か判定をして画面に表示するプログラムを作成しなさい。ただし、入力された値が整数以外であれば、「整数と認識できません！！」と画面に表示しなさい。

【実行例 1】偶数が入力された時

整数を入力 = 2

2 は偶数

【実行例 2】奇数が入力された時

整数を入力 = 3

3 は奇数

【実行例 3】実数値が入力された時

整数を入力 = 3.13

整数と認識できません！！

問題 15_3

Python / Java

5 つの整数が格納されている配列を準備しなさい。ユーザに整数を入力してもらい、その整数が配列の領域内であれば、インデックス番号とそれに対応する要素を画面に表示するプログラムを作成しなさい。ただし、配列の領域外であれば、「領域外参照例外」を補足してメッセージを画面に表示しなさい。

【実行例 1】領域内の時

```
base_list = [1, 2, 3, 4, 5]
インデックス番号 = 3
base_list[3] = 4
終了
```

【実行例 2】領域外の時

```
base_list = [1, 2, 3, 4, 5]
インデックス番号 = 6
領域外参照です
終了
```

問題 15_4

Python / Java

ユーザ定義例外” DividedByZero” を定義しなさい。2 つの数値を引数として受け取り、割り算をする関数(メソッド)を定義しなさい。その関数(メソッド)を利用してユーザが入力した 2 つの数値の割り算の結果を画面に表示するプログラムを作成しなさい。ただし、0 で割った場合は定義したユーザ定義例外を送出し、関数の呼び出し元で補足したのち、その旨のメッセージを画面に表示しなさい。

【実行例 1】0 以外の数で割ったとき

```
数値 1= 10
数値 2= 2
10 / 2 = 5.0
```

【実行例 2】0 で割ったとき

```
数値 1 = 10
数値 2 = 0
0 で割りました！！
```

ユーザ定義例外” OutOfIndex” と” MinusIndex” を定義しなさい。また、数値の配列とインデックス番号を引数として受け取り、配列の中の指定されたインデックスの要素が偶数か奇数か判定をし、偶数の場合には真を返し、奇数の場合には偽を返す関数(メソッド)を作成しなさい。ただし、インデックス番号が配列の領域外であれば” OutOfIndex” を、インデックス番号が負の数値であれば” MinusIndex” を送出するようにしなさい。その関数(メソッド)を利用して、以下の配列の、ユーザが入力したインデックス番号の要素が偶数か奇数か判定をするプログラムを作成しなさい。例外が送出された場合には、補足し例外の内容を画面に表示するようにしなさい。

【配列】

[1, 2, 3, 4, 5]

【実行例 1】 インデックス番号がリストの要素数より大きいときの実行結果

元のリスト = [1, 2, 3, 4, 5]

インデックス番号 = 7

領域外参照です。

処理終了

【実行例 2】 インデックス番号が負の値の時の実行結果

元のリスト = [1, 2, 3, 4, 5]

インデックス番号 = -1

インデックス番号に負の値を指定しています。

処理終了

【実行例 3】 インデックス番号の要素が偶数のとき

元のリスト = [1, 2, 3, 4, 5]

インデックス番号 = 1

base_list[1] = 2 は偶数です。

処理終了

【実行例 4】 インデックス番号の要素が奇数のとき

元のリスト = [1, 2, 3, 4, 5]

インデックス番号 = 2

base_list[2] = 3 は偶数ではありません。

処理終了