

Chapter 5

例外処理・入出力・配列

以下の指示に従って、クラス、オブジェクトを作成・実行せよ。なお、文中では個別に指示しないが、変更を行ったら適宜コンパイルすること。

5.1 例外処理

- 次のクラスを作成し、実行せよ

```
class Warizan {
    int a=0;    //int であることに注意
    int b=0;    //int であることに注意
    void setA(int a) {
        this.a=a;
    }
    void setB(int b) {
        this.b=b;
    }
    int div() {
        int r = this.a/this.b;
        return r;
    }
}

class Test05A {
    public static void main(String[] args) {
        Warizan w = new Warizan();
    }
}
```

```

        w.setA(10);
        w.setB(3);
        int result = w.div();
        System.out.println("result=" + result);
    }
}

```

- `w.setB(3);` という行を `w.setB(0);` としたらどうなるか。例外が発生する場合、どのような例外が発生しているか調べよ。また、その例外はどこで発生しているか調べよ。
- `ArithmeticException` が発生した場合、“Divided by zero.” を画面に表示するよう処理を追加せよ。
 - Warizan クラスで例外ハンドラ (try-catch) を利用する方法を実際に行ってみよ。
 - (発展問題) Warizan クラスでは例外をキャッチせず、Warizan クラスのオブジェクトの利用側 (今の例では Test05A) に例外をスローする方法を試してみよ。(throws というキーワードで調べよ。なお、この場合は Test05A 側で例外ハンドラ (try-catch) を利用する必要がある。
- 加えて、`ArithmeticException` 以外の例外が発生した場合、“Other exception.” を画面に表示するよう処理を追加せよ。(Warizan クラスで例外ハンドラを利用する方法のみでの実現でよい)
- Warizan クラスで例外ハンドラを利用する方法において、例外が発生する・しないに関わらず、`div` メソッドの終了時に “div メソッド終了” と表示せよ。(ただし、 $r = a/b$ を実行した後に表示処理を行うものとする)

5.2 ファイル入力

- カレントディレクトリにあるファイル `Sample.txt` を読み取り画面に出力する `main` メソッドをもつクラス `Test05B` を作成せよ。ただし、ファイルは Shift JIS で書かれていることとし、読み取り時に例外が発生すれば `Exception` クラスの `printStackTrace()` を実行するようにせよ。ただし、`FileInputStream`、`InputStreamReader`、`BufferedReader` を使う

こと。また、例外が起こると起こらないに関わらず、必ず open したオブジェクトは close すること。(finally 文を使え。ただし、finally 文の中括弧内でさらに try-catch 文を使っても良い。1 限のプリントを参考にせよ。)

- カレントディレクトリに Sample.txt が存在しないときには、“ファイルがありません。”と表示するように例外処理を変更せよ。また、実際に Sample.txt を別の名前に変更し、プログラムを実行せよ。

5.3 配列

- 次のクラスを作成し、実行せよ。

```
class Test05C {
    public static void main(String[] args) {
        int[] intArray = new int[10];
        for(int j=0; j<10; j++) {
            intArray[j] = (int)(Math.random() * 10) + 1;
        }
        System.out.println(intArray[0]);
    }
}
```

- main メソッドで、intArray[0] を表示しているが、これを配列に格納されている整数を全て画面に表示するように処理を変更せよ。
- java Test05C 20 のように、引数から整数を受け取り、配列 intArray をこの整数の数だけ要素を持つように処理を変更せよ。ただし、繰り返しには for 文を用い、繰り返しの終了条件には配列 intArray の要素数を用いよ。(要素数は intArray.length のように配列名.length として取得できる。これを試せ。)
- java Test05C a のように、数字以外の文字列が引数として与えられたときの例外処理を行え。ただし、例外処理は、NumberFormatException 例外に対してのみ行い、“数字以外の文字が入力されました”を画面に表示するものとする。