

第7章 例外処理①

(1) 次の空欄を埋めよ。

例外処理は、try～catch 句を使用して対処していく。

```
try{
    例外が発生する可能性のある処理
} catch (例外クラス 変数名){
    引数の例外がおきたときの処理
} ( _____ ) {
    最後に必ず行われる処理
}
```

[finally]

(2) 次のプログラムをコンパイルし実行した結果として、正しいものは次のどれか。

```
public class Sample110 {
    public static void main(String[] args) {
        String data = "Hello Silver";
        data.substring(5, 3);
        System.out.println(data);
    }
}
```

- a. NullPointerException が発生する
- b. NumberFormatException が発生する
- c. ClassCastException が発生する
- d. StringIndexOutOfBoundsException が発生する
- e. IOException が発生する

[d]

(3) 次のプログラムをコンパイルし、コマンドライン引数に a b を指定して実行した結果は次のどれか。

```
public class Mainclass3 {
    public static void main(String[] args) {
        try{
            int i1 = Integer.parseInt(args[0]);
            int i2 = Integer.parseInt(args[1]);
            System.out.println(i1 / i2);
        }catch(ArithmeticException e){
            System.out.println(e + "が発生しました");
        }catch(ArrayIndexOutOfBoundsException e){
            System.out.println(e + "が発生しました");
        }
        System.out.println("終了");
    }
}
```

Java II 第7章小テスト

- a. なにも出力されない
- b. java.lang.ArrayIndexOutOfBoundsException: 1が発生しました
- c. 終了
- d. java.lang.NumberFormatExceptionが発生し、プログラムが終了する
- e. java.lang.ArithmeticException: 1が発生しました

[d]

(4) 次のプログラムはコンパイルできない。このプログラムをコンパイルできるように書き換える方法として正しいものは次のどれか。2つ選択しなさい。

```
public class Mainclass1 {  
    public static void main(String[] args) {  
        String name = "test.java";  
        FileReader fr = null;  
        fr = new FileReader(name);  
        //省略  
        fr.close();  
    }  
}
```

- a. main()メソッドに throw IOException を記述する
- b. FileNotFoundException と IOException を補足するために try {} ブロックと catch {} ブロックを記述する
- c. FileNotFoundException を補足するために try {} ブロックと catch {} ブロックを記述する
- d. main()メソッドに throws IOException を記述する
- e. main()メソッドに throws FileNotFoundException を記述する

[b] [d]

(5) try/catch ブロックの記述でコンパイルに失敗するものは次のどれか。

- a.

```
public void method() {  
    try {  
    } catch (Exception e) {  
    } finally {  
    }  
}
```
- b.

```
public void method() {  
    try {  
    } catch (Exception e) {  
    }  
}
```
- c.

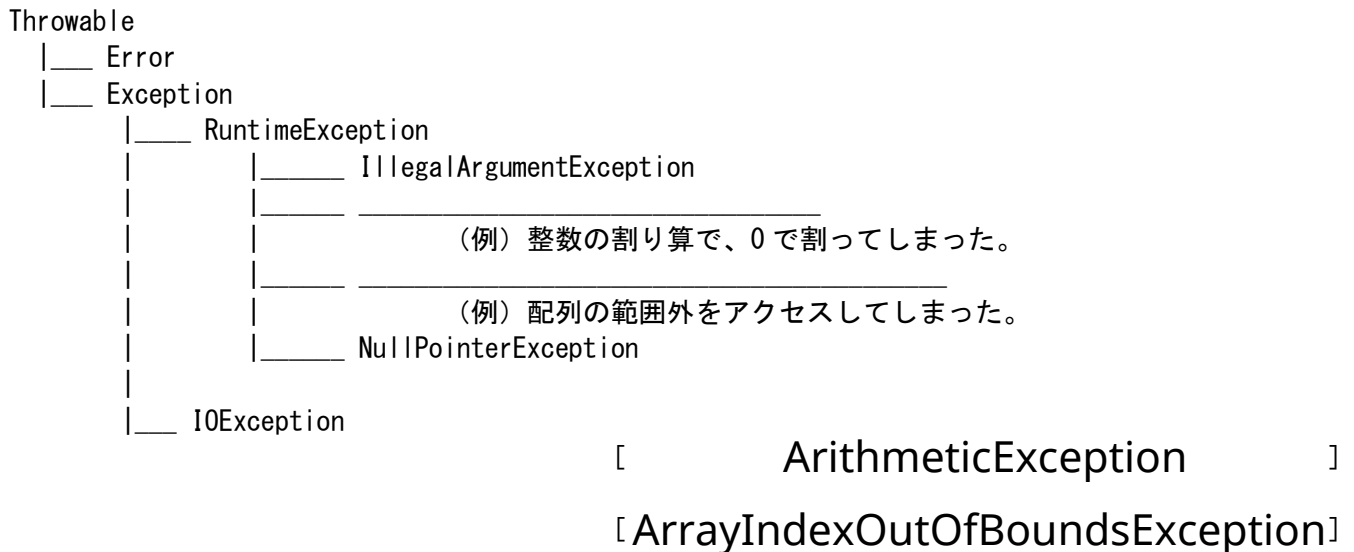
```
public void method() {  
    try {  
    }  
}
```
- d.

```
public void method() {  
    try {  
    } finally {  
    }  
}
```

[c]

第8章 例外処理②

(1) 次の例外クラスのツリー図の空欄を埋めよ。



(2) 例外クラスに関する説明文で、正しいものは次のどれか。

- a. RuntimeException のサブクラス以外の例外は、例外処理をしなくてよい
- b. Exception クラスは、例外に関するクラスの最上位クラスである
- c. 発生した例外に対する処理を記述する用法は、例外が発生したメソッドで try {} catch {} ブロックを記述する方法のみである
- d. すべての例外に対する処理は、必要に応じて任意で実装する
- e. Error クラスは、プログラムの物理的な実行環境などに関する致命的なエラー

[e]

(3) 次のプログラムをコンパイルし実行した結果出力されるものはどれか。

```

public class Sample {
    public static void main(String[] args) {
        int[] ary = {0, 1, 2, 3, 4};
        for (int i=0; i<ary.length; i++) {
            try {
                method(ary[i]);
                System.out.print(ary[i]);
            } catch (Exception e) {
            }
        }
    }
    private static void method(int num) throws Exception{
        if (num == 3) {
            throw new Exception();
        }
    }
}
    
```

J a v a II 第 8 章小テスト

- a. 01234
- b. 0124
- c. 01
- d. 012

[b]

(4) 次のプログラムをコンパイルし実行した結果として正しいものは次のどれか。

```
public class Sample {
    public static void main(String[] args) {
        try {
            method();
            System.out.println("no exception.");
        } catch (Exception e) {
            System.out.println("exception caught.");
        }
    }
    private static void method() throws Exception{
        try {
            throw new Exception();
        } catch (Exception e) {
            throw e;
        } finally {
            return;
        }
    }
}
```

- a. no exception.
- b. exception caught.
- c. no exception.
exception caught.
- d. 何も表示されない

[a]

(5) チェックされる例外の独自例外クラスの定義として正しいものは次のどれか。

- a.

```
public class MyException extends Exception {
}
```
- b.

```
public class MyException extends RuntimeException {
}
```
- c.

```
public class MyException extends Error {
}
```
- d.

```
public class MyException {
}
```

[a]

第9章 スレッド

(1) 次の空欄を埋めよ。

・複数のスレッドが動くプログラムのことを、_____のプログラムと言う。

[マルチスレッド]

(2) 次の空欄を埋めよ。

スレッドを生成するには、以下の2つの方法がある。

① クラス Thread のサブクラスを作成する方法

```
class Thread1 extends Thread{    // Thread クラスを継承したクラスを作成
    public void run(){            // Thread クラスの run メソッドをオーバーライドする
        //このスレッドで行いたい処理を記述
    }
}
class Test1{
    public static void main(String[] args){
        Thread1 th1 = new Thread1();
        th1. ; //Thread1 クラスを開始するメソッドを呼び出す
    }
}
```

[start()]

(3) 次の空欄を埋めよ。

② インターフェイス Runnable を implements したクラスを作成する方法

```
class Thread2 implements Runnable{    // Runnable インターフェイスを実装したクラスを作成
    public void run(){                // Runnable インターフェイスの run メソッドをオーバーライドする
        //このスレッドで行いたい処理を記述
    }
}
class Test2{
    public static void main(String[] args){
        Thread2 th2 = new Thread2();    //Thread2 クラスのオブジェクトを引数にした、
        Thread th = new Thread(th2);    // Thread クラスのオブジェクトを作成
        ;                // スレッドを開始する
    }
}
```

[th.start()]

(4) 現在のスレッドの動作を 1.5 秒間止めるための命令文を記述せよ。

[sleep(1500)]

第10章 スレッドの同期

(1) 次の空欄を埋めよ。

・複数のスレッドが同じ変数にアクセスする場合には、スケジューリング次第で想定外のことが起きる可能性があるため、同期が必要になる場合がある。

・J a v a では、_____キーワードを使用する。

[synchronized]

(2) スレッドについての正しい記述はどれか。3 つ選択しなさい。

- a. スレッドの制御はスレッドスケジューラが行い、スレッドスケジューラの機能は環境に依存する。
- b. スレッドの処理内容は、メソッド run に記述する。
- c. run メソッドに引数を渡してスレッドを制御できる。
- d. run メソッドがリターンしたスレッドでも、再実行できる。
- e. スレッドの状態には、実行中、実行可能、実行不可能、死んでいる(消滅)の4つがある。

[a] [b] [e]

J a v a II 第 10 章小テスト

(3) 次のプログラムをコンパイルし実行した結果出力されるものはどれか。

```
class X implements Runnable{
    int a;
    public void run() {
        for(int i = 0; i < 10; i++){
            System.out.print(a + " ");
        }
    }
}
class Test{
    public static void main(String[] args){
        Thread t1 = new Thread(new X());
        Thread t2 = new Thread(new X());
        t1.start();
        t2.start();
    }
}
```

- a. a の値が交互に表示され、表示される値も同じである
- b. a の値が交互に表示されるが、表示される値は同じ値とは限らない
- c. t1 と t2 の a の値は表示されるが、表示される順序は不定
- d. t1 の a の値が表示し終わったら、t2 の a の値が表示される
- e. コンパイルエラーになる

[]

第11章 ファイルアクセス

(1) 次のプログラムに関する説明で、正しいものはどれか。

```
public class Mainclass {  
    public static void main(String[] args) throws IOException{  
        String name = "test.txt";  
        FileReader fr = null;  
        fr = new FileReader(name);  
        // (略)  
        fr.close();  
    }  
}
```

- a. このプログラムをコンパイルするためには、import 文が必要である。
- b. このプログラムで使用している FileReader クラスは、java. awt パッケージに含まれている。
- c. このプログラムで使用している FileReader クラスは、java. lang パッケージに含まれている。
- d. このプログラムは Java API を使用していない
- e. このプログラムはこのままコンパイルが可能である。

[a]

(2) 次のプログラムに関する説明として、正しいものはどれか。

```
public class Mainclass {  
    public static void main(String[] args) throws Exception {  
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
        String s1 = br.readLine();  
        System.out.println(s1);  
    }  
}
```

- a. readLine() メソッドは java. lang パッケージに含まれている。
- b. readLine() メソッドを使用するには、import 文は記述しなくてよい。
- c. readLine() メソッドはクラスメソッドとして定義されている。
- d. readLine() メソッドは指定した入力装置から入力された情報を取得するための機能を提供している。
- e. readLine() メソッドはインターフェースクラスとして定義されている。

[d]

J a v a II 第 11 章小テスト

(3) 次のプログラムがコンパイルできるようにするために空欄を埋めるためには、どれが正しいか。

```
public class Mainclass {  
    public static void main(String[] args)  IOException{  
        String name = "a.java";  
        FileReader fr = null;  
        fr = new FileReader(name);  
        // (略)  
        fr.close();  
    }  
}
```

- a. extends
- b. implements
- c. throw
- d. throws
- e. catch

[d]

第12章 コレクション

(1) ArrayList 型のインスタンスを、生成できないものは次のどれか。2 つ選択しなさい。

- a. ArrayList<Object> a = new List<Object>();
- b. ArrayList a = new ArrayList();
- c. ArrayList<Object> a = new ArrayList<Object>();
- d. ArrayList<Object> a = new ArrayList<>();
- e. ArrayList<> a = new ArrayList<Object>();

[a] [e]

(2) 次のプログラムをコンパイルし実行した結果として、正しいものは次のどれか。

```
1: import java.util.ArrayList;
2: class ArrayListTest {
3:     public static void main(String[] args) {
4:         ArrayList<Integer> list = new ArrayList<>();
5:         list.add(100);
6:         list.add(101);
7:         System.out.println(list.get(list.size()));
8:     }
9: }
```

- a. 4 行目が原因でコンパイルエラーが発生する。
- b. 6 行目が原因でコンパイルエラーが発生する。
- c. 7 行目が原因で例外エラーが発生する。
- d. 100
- e. 101

[c]

(3) 次のプログラムをコンパイルし実行した結果出力されるものはどれか。

```
import java.util.ArrayList;

public class Sample {
    public static void main(String[] args) {
        ArrayList<String> a = new ArrayList<String>();
        a.add(new String("春"));
        a.add(new String("夏"));
        a.add(null);
        a.add(new String("秋"));
        a.add(new String("冬"));
        a.remove(new String("夏"));
        for (String b : a) {
            System.out.print "[" + b + " ";
        }
    }
}
```

Java II 第12章小テスト

- a. [春][][秋][冬]
- b. [春][秋][冬]
- c. コンパイルに失敗する
- d. [春][null][秋][冬]
- e. [春][夏][秋][冬]

[d]

(4) 次のプログラムをコンパイルし実行した結果として、正しいものはどれか。

```
import java.util.*;

class ArrayListTest {
    public static void main(String[] args) {
        ArrayList<Integer> list = new ArrayList<Integer>();
        list.add(1); list.add(5); list.add(2);
        Collections.sort(list);
        list.add(3);
        Collections.reverse(list);
        System.out.println(list);
    }
}
```

- a. [1, 2, 3, 5]
- b. [1, 2, 5, 3]
- c. [5, 3, 2, 1]
- d. [3, 5, 2, 1]
- e. コンパイルエラーが発生する

[d]

(5) 次のプログラムをコンパイルし実行した結果として、正しいものはどれか。

```
import java.util.*;

class ArrayListLoop {
    public static void main(String[] args) {
        ArrayList<Integer> list = new ArrayList<Integer>();
        for(int i = 0; i < 5; i++)
            if(i % 2 == 0) list.add(i);
        list.add(7);
        System.out.println(list);
    }
}
```

- a. コンパイルエラーが発生する
- b. 何も出力されない
- c. [7, 0, 2, 4]
- d. [1, 3, 5, 7]
- e. [0, 2, 4, 7]

[e]

第13章 ネットワーク通信

(1) 次の空欄を埋めよ。

ネットワーク通信を簡単に実現するためのクラスが存在し、細かな制御は_____クラスで行ってくれるため、我々は簡単に Web ページの取得が行うことができる。

[java.net.URL]

(2) 次の空欄を埋めよ。

<java.net.URL クラスの一般的な利用手順>

①URL クラスを_____する。

```
URL url = new URL("http://www.yahoo.co.jp/");
```

[インスタンス化]

②openStream() を呼び出して、データを取り出す_____を取得する。

```
InputStream is = url.openStream();
```

[ストリーム]

③read() を呼び出して、ストリームから 1 バイトずつ情報を取り出す。

```
int data = is._____;
```

[read()]

(3) 次の空欄を埋めよ。

①read() を呼び出して、ストリームから最初の 1 バイトを取り出し、終わりか判定する

```
int c = isr.read();
```

```
while(c != _____) {
```

[-1]

```
System.out.print((char) c);
```

```
//read() を呼び出して、ストリームから次の 1 バイトを取り出す
```

```
c = isr.read();
```

```
}
```

```
//ストリームを閉じる
```

```
isr._____;
```

[close()]

J a v a II 第 13 章小テスト

(4) 次の空欄を埋めよ。

<Socket を用いた接続と切断の一般的な利用手順>

①IP アドレスまたはサーバー名と_____を指定して Socket をインスタンス化する。

```
Socket sock = new Socket("yahoo.com", 80);
```

[**ポート番号**]

②Socket から入力ストリームと_____を取得する。

```
InputStream is = sock.getInputStream();
```

```
OutputStream os = sock.getOutputStream();
```

[**出力ストリーム**]

③2 つのストリームを読み書きする。

```
int data = is.read();
```

```
os._____ ("HELLO");
```

[**write**]

④ソケットを閉じる。(Socket の close() を呼べば、入力と出力の 2 つのストリームも自動的に閉じられる。

```
sock._____;
```

[**close()**]