

# プログラミング A 第7回・宿題

宿題の提出は Moodle で木曜日までに行ってください。各宿題ごとに提出ファイルを zip 等で一つのファイルにまとめて該当する Moodle の課題に提出しなさい。この資料や関係するコードをインターネットなどに公開することは著作権上、禁止されています。

## 1 宿題 1

以下をすべて満たすクラス A, B を作成し、提出せよ。

- A のインスタンスに対してメソッド m() を呼び出すと戻り値として、文字列リテラル "a" が返される。
- A のコンストラクタには引数を取れない。
- A は、A および B の累積のインスタンス生成個数を int 型で返すクラスメソッド getCounter() を持つ。
- B は A のサブクラスである。
- B のインスタンスに対してメソッド m() を呼び出すと戻り値として、コンストラクタの引数に指定した A または B のインスタンスに対して m() を呼び出して得られる戻り値の前後に "b" を付与した文字列を返す。

以上を応用して、例えば添付の ABCClient クラスを実行すると以下に示す期待結果を標準出力に得る。このとき、if 文や switch・case 文を使用せずに A および B を作成し、ABCClient を除く全プログラムを解答欄に記述しなさい。

```
0
a
bab
bbabb
6
```

参考資料：ABCClient.java

---

```
public class ABCClient {
    public static void main(String[] args) {
        System.out.println( A.getCounter() );
        System.out.println( new A().m() );
        System.out.println( new B( new A() ).m() );
        System.out.println( new B( new B( new A() ) ).m() );
        System.out.println( A.getCounter() );
    }
}
```

---

## 2 宿題 2-1

ExceptionHomework を実行すると、標準出力に以下を得る。以下の出力を得た理由（仕組み）を簡潔に説明せよ。

```
try
finally
10
try
catch
finally
-1
```

参考資料：ExceptionHomework.java

---

```
public class ExceptionHomework {

    static int number(String s) {
        try {
            System.out.println("try");
            return Integer.parseInt(s);
        } catch (NumberFormatException nfe) {
            System.out.println("catch");
            return -1;
        } finally {
            System.out.println("finally");
        }
    }

    public static void main(String[] args) {
        System.out.println(number("10"));
        System.out.println(number("XXXXXXXX"));
    }

}
```

---

## 3 宿題 2-2

元の ExceptionHomework について、以下を満足するように改変して、全体を提出せよ。OriginalException のコードも提出せよ。

- Exception のサブクラスとして OriginalException クラスを定義すること。
- number メソッド中の catch 節の中で、標準出力への文字列”catch” の出力の直後に、OriginalException

のインスタンスを生成して投げること。

- main メソッドにおいて OriginalException インスタンスをキャッチして、その内容を標準出力に出力すること。
- これらの実現にあたり必要な改変を適宜施して、実行すると以下を標準出力に得ること。

```
try
finally
10
try
catch
finally
OriginalException
```

## 4 宿題 3-1

添付の GcHomework を動かした時、System.gc() を行った直後の時点で参照され生き残っている（ガーベジコレクタにより収集されていない）クラス Person の全インスタンスの name の値すべてを列挙し、その結果を得た理由を説明せよ。

参考資料：GcHomework.java

---

```
class Person {
    String name = null;
    Person friend = null;
    Person(String s) {
        name = s;
    }

    void setFriend(Person p) {
        friend = p;
    }
}

public class GcHomework {

    public static void main(String[] args) {
        Person p1 = new Person("P1");
        Person p2 = new Person("P2");
        Person p3 = new Person("P3");
        p1.setFriend(p2);
        p2.setFriend(p3);
        p3.setFriend(p1);
        p1 = null;
        p2 = null;
        //p3 = null;
```

```
        System.gc();  
    }  
  
}
```

---

## 5 宿題 3-2

GcHomework においてコメントアウトされた行「`p3 = null;`」を有効にした場合に、`System.gc()` を行った直後の時点で参照され生き残っている（ガーベッジコレクタにより収集されていない）クラス `Person` の全インスタンスの `name` の値すべてを列挙し、その結果を得た理由を説明せよ。

## 6 宿題 4

以下のそれぞれについて、この小テスト（宿題）内で扱うプログラムソースコードを参照する形で説明せよ。

- (1) 例外の使いどころ（何を目的としてどのような場合に用いるのか）
- (2) 例外の種類と階層
- (3) ガーベッジコレクションの使いどころ