

# プログラミング A 第6回・演習

演習の提出は Moodle で木曜日までに行ってください。各演習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 演習 1

以下を満たす Label クラスを定義し、LabelTest から利用できることを確認の上、Label.java の内容を回答欄に貼り付けなさい。

- String 型のインスタンスフィールド caption を持つ。
- String 型のクラスフィールド common を持つ。
- String を引数にとり、caption を初期化するコンストラクタを持つ。
- 戻り値の型が void、引数をとらないインスタンスメソッド output を持ち、同メソッドを呼び出すと <common の値>: <caption の値> と標準出力に出力する。

なお、LabelTest を実行すると標準出力に以下が出力される。

図: チーズを好む皇帝ペンギン

図: 人類補完計画達成率

表: チーズを好む皇帝ペンギン

表: 人類補完計画達成率

参考資料: LabelTest.java

---

```
class LabelTest {  
    public static void main(String[] args) {  
        Label.common = "図";  
        Label l1 = new Label("チーズを好む皇帝ペンギン");  
        Label l2 = new Label("人類補完計画達成率");  
        l1.output();  
        l2.output();  
        Label.common = "表";  
        l1.output();  
        l2.output();  
    }  
}
```

---

## 2 演習 2

以下を満たす Shop クラス、Bakery クラス、FlowerShop クラスをそれぞれ定義し、ShopTest から利用できることを確認の上、Shop.java、Bakery.java、FlowerShop.java の全ての内容を回答欄に貼り付けなさい。

- Shop クラスがスーパークラス、Bakery クラスおよび FlowerShop クラスがサブクラスである。

- Shop クラス:
  - String 型のインスタンスフィールド name を持つ。
  - String を引数にとり、name を初期化するコンストラクタを持つ。
  - 引数を取らず、String を戻り値とする sell メソッドを持つ。ただし Shop インスタンスに対して sell を呼び出すと null が返る。
- Bakery クラスおよび FlowerShop クラス:
  - Shop クラスから name を継承する。
  - sell メソッドをオーバーライドする。
- Bakery クラス:
  - 引数を取らず、Shop クラスのコンストラクタを利用して name を”パン屋”として初期化するコンストラクタを持つ。
  - sell を呼び出すと、”パン”が返る。
- FlowerShop クラス:
  - 引数を取らず、Shop クラスのコンストラクタを利用して name を”花屋”として初期化するコンストラクタを持つ。
  - sell を呼び出すと、”花”が返る。

なお、ShopTest を実行すると標準出力に以下が出力される。

花屋の花

パン屋のパン

花屋の花

参考資料：ShopTest.java

---

```
class ShopTest {
    public static void main(String[] args) {
        Shop[] mall = new Shop[] {
            new FlowerShop(),
            new Bakery(),
            new FlowerShop()
        };
        for(int i = 0; i < mall.length; i++) {
            System.out.println(mall[i].name + "の" + mall[i].sell());
        }
    }
}
```

---

### 3 演習 3

以下を満たす Movable インタフェース、Animal クラス、Robot クラスをそれぞれ定義し、MovableTest から利用できることを確認の上、Movable.java、Animal.java、Robot.java の全ての内容を回答欄に貼り付けなさい。

- Movable インタフェースは、戻り値の型が void かつ引数をとらないメソッド move を宣言する。
- Animal クラスおよび Robot クラスは、Movable インタフェースを実装する。
- Animal クラスの move メソッドを呼び出すと、"Animal moved." と標準出力に出力する。
- Robot クラスの move メソッドを呼び出すと、"Robot moved." と標準出力に出力する。

なお、MovableTest を実行すると標準出力に以下が出力される。

```
Robot moved.
Animal moved.
Robot moved.
```

参考資料：MovableTest.java

---

```
class MovableTest {
    public static void main(String[] args) {
        Movable m1 = new Animal();
        Movable m2 = new Robot();
        m2.move();
        m1.move();
        m2.move();
    }
}
```

---

## 4 演習 4

以下を満たす File クラス、Directory クラスをそれぞれ定義し、FileTest から利用できることを確認の上、File.java、Directory.java の全ての内容を回答欄に貼り付けなさい。

- File クラス:
  - String 型のインスタンスフィールド name により名前を表す。
  - int 型のインスタンスフィールド size によりサイズ [byte] を表す。
  - コンストラクタによりインスタンスを生成する際に、名前とサイズを指定できる。
  - 名前、サイズをそれぞれ外部から取得するためのインスタンスメソッド getName() および getSize() を持つ。
- Directory クラス:
  - コンストラクタによりインスタンスを生成する際に、名前を指定できる。
  - 名前、サイズをそれぞれ外部から取得するためのインスタンスメソッド getName() および getSize() を持つ、または、継承している。
  - インスタンスメソッド add() により、Directory または File のインスタンスを 10 個まで追加できる。
  - ディレクトリそのもののサイズは共通 1 byte であるが、サイズを外部から問い合わせると、内包する全てのディレクトリおよびファイルのサイズを合計した結果が得られる。
- ヒント: File と Directory に共通のデータや手続きは File にまとめて、Directory を File のサブクラ

スとすればよい。これにより、Directory が File に加えて Directory も内包できるという再帰構造を形成できる。

なお、FileTest を実行すると標準出力に以下が出力される。

```
952
docs
101
```

参考資料：FileTest.java

---

```
public class FileTest {
    public static void main(String[] args) {
        Directory d1 = new Directory("docs");
        Directory d2 = new Directory("excels");
        File f1 = new File("aaa.xls", 850);
        File f2 = new File("temp.txt", 100);
        d1.add(d2);
        d2.add(f1);
        d1.add(f2);
        System.out.println(d1.getSize());
        System.out.println(d1.getName());
        Directory d3 = new Directory("test");
        for(int i = 0; i < 20; i++) {
            d3.add(new File("dummy" + i, 10));
        }
        System.out.println(d3.getSize());
    }
}
```

---