

# プログラミング A 第6回・宿題 回答

## 1 宿題 1

Factory.java

---

```
class Factory {  
    String name;  
    Factory(String n){  
        name = n;  
    }  
    String produce() {  
        return null;  
    }  
}
```

---

CarFactory.java

---

```
class CarFactory extends Factory {  
    CarFactory() {  
        super("車工場");  
    }  
    String produce() {  
        return "車";  
    }  
}
```

---

BikeFactory.java

---

```
class BikeFactory extends Factory {  
    BikeFactory() {  
        super("自転車工場");  
    }  
    String produce() {  
        return "自転車";  
    }  
}
```

---

## 2 宿題 2

Figure.java

---

```
public abstract class Figure {  
    public abstract void move(int mx, int my);  
    public abstract String draw();  
}
```

---

Point2.java

---

```
public class Point2 extends Figure {
    private int x = 0;
    private int y = 0;
    public Point2(int nx, int ny) {
        x = nx;
        y = ny;
    }
    public String draw() {
        return "(" + x + "," + y + ")";
    }
    public void move(int mx, int my) {
        x += mx;
        y += my;
    }
}
```

---

Line2.java

---

```
public class Line2 extends Figure {
    private Point2 p1 = null;
    private Point2 p2 = null;
    public Line(Point2 np1, Point2 np2) {
        p1 = np1;
        p2 = np2;
    }
    public String draw() {
        return "(" + p1.draw() + p2.draw() + ")";
    }
    public void move(int mx, int my) {
        p1.move(mx, my);
        p2.move(mx, my);
    }
}
```

---

### 3 宿題 3

Countable.java

---

```
public interface Countable {
    int count();
}
```

---

Text.java

---

```
public class Text implements Countable {
    String string = "";
}
```

```

public Text(String s) {
    if(s != null) string = s;
}
public int count(){
    return string.length();
}
}

```

---

IntArray.java

---

```

public class IntArray implements Countable {
    int[] iarray = new int[0];
    public IntArray(int[] ia) {
        if(ia != null) iarray = ia;
    }
    public int count() {
        return iarray.length;
    }
}

```

---

## 4 宿題 4

### (4-1) クラスとインタフェースの共通点と相違点

- － 共通: 参照型。インスタンスメソッドを持てる (ただしインタフェースは public な抽象メソッドのみ)。クラスフィールドを持てる (ただしインタフェースは public final な定数フィールドのみ)。
- － 相違: クラスのインスタンスを作れるが、インタフェースのインスタンスは作れない。クラスと異なり、インタフェースにインスタンスフィールドやクラスメソッドを定義できない。

### (4-2) クラス拡張とインタフェース実装の共通点と相違点

- － 共通: 多態性を実現可能。拡張元・実装元のインスタンスメソッドを、拡張側・実装側において必ず具体的に持つ。拡張元・実装元の型の変数に、拡張・実装したクラスのインスタンスを参照させられる。
- － 相違: クラスは、1 つのみスーパークラスを指定して拡張可能だが、スーパーインタフェースは複数指定して実装可能。クラスの拡張上の階層に対して、インタフェースの実装は横断的に可能。

### (4-3) クラスと抽象クラスの共通点と相違点

- － 共通: モジュール単位。メソッドやフィールドをまとめあげる。
- － 相違: クラスと異なり抽象クラスのインスタンスは作れない。クラスと異なり抽象クラスは抽象メソッドを持てる。