

プログラミングA 1班

10-11章

宿題解答

シモセラ エドガー
情報理工

Line.java

```
public class Line {
    double p1x = 0; double p1y = 0;
    double p2x = 0; double p2y = 0;
    Line(double p1x, double p1y, double p2x, double p2y) {
        this.p1x = p1x;
        this.p1y = p1y;
        this.p2x = p2x;
        this.p2y = p2y;
    }
    public String toString() {
        return "(" + p1x + "," + p1y + ")-(" + p2x + "," + p2y + ")";
    }
    public double getLength() {
        double dx = Math.abs(p1x - p2x);
        double dy = Math.abs(p1y - p2y);
        return Math.sqrt(dx * dx + dy * dy);
    }
    public static void main(String[] args) {
        Line line1 = new Line(10.5, 20.1, 100.0, 50.0);
        Line line2 = new Line(-1.0, 0.0, 0.0, 1.0);
        System.out.println(line1);
        System.out.println(line1.getLength());
        System.out.println(line2);
        System.out.println(line2.getLength());
    }
}
```

Item

```
public class Item {  
  
    String name = null;  
    int price = 0;  
    static int numberOfInstances = 0;  
  
    static int getNumberOfInstances() {  
        return numberOfInstances;  
    }  
  
    public Item(String n, int p) {  
        name = n;  
        price = p;  
        numberOfInstances++;  
    }  
  
    public String toString() {  
        return name + ", " + price;  
    }  
  
}
```

Cart

```
public class Cart {
    Item[] items = new Item[10];
    int index = 0;
    void addItem(Item item) {
        if(index < items.length) {
            items[index++] = item;
        }
    }
    void info() {
        for(int i = 0; i < index; i++) {
            System.out.println(items[i]);
        }
    }
    public static void main(String[] args) {
        System.out.println(Item.getNumberOfInstances());
        Cart cart = new Cart();
        Item i1 = new Item("PC", 98000);
        Item i2 = new Item("Display", 40000);
        cart.addItem(i1);
        cart.addItem(i2);
        cart.info();
        System.out.println(Item.getNumberOfInstances());
    }
}
```

解答例（実際にはより短くとも可）

- ただし、具体例を適切に参照していること。
- (1) コンストラクタとメソッドの共通点と相違点
 - 共通: 引数を取る。まとまった処理を実現する。
 - 相違: コンストラクタはインスタンス生成時の初期化処理を担い、戻り値を取らず、名前はクラス名と同じ。対してメソッドは初期化に限らず様々な処理を担い、戻り値を取ることができ、名前はクラス名以外の任意のもの。
- (2) クラスメソッドとインスタンスメソッドの共通点と相違点
 - 共通: 引数を取り、戻り値を取ることができる。まとまった処理を実現するメソッドである。
 - 相違: クラスメソッドはクラスに共通の振る舞いを持ち、クラスを指定して呼び出せ、内部ではクラスフィールドやクラスメソッドおよび引数のみ利用できる。対してインスタンスメソッドは各インスタンス特有の振る舞いを持ち、インスタンスを指定して呼び出せる。
- (3) クラスフィールドとインスタンスフィールドの共通点と相違点
 - 共通: 指定された型のデータ・値を表す。
 - 相違: クラスフィールドはクラスに共通の値を表し、クラスを指定して参照できる。対してインスタンスフィールドは各インスタンス固有の値を表し、インスタンスを指定して参照できる。