独習Javaゼミ

５．５　thisキーワード

５．６　インスタンス変数とインスタンスメソッド

５．７　静的変数と静的メソッド

2007/05/11　鈴木 慧

５．５　this キーワード

- this キーワードは実行中のオブジェクトを指す

This キーワードの使い方

this.変数	オブジェクト内のインスタンス変数の参照

this(引数);	オブジェクト内のコンストラクタの呼び出し

this キーワードの例

```
class Circle {
    double x, y;
    Circle(double x) {
        this(x, 0);
    }

    Circle(double x, double y) {
        this.x = x ; this.y = y ;
    }
}

class CircleThis {
    public static void main (String args[]) {
        Circle c = new Circle(3);
        System.out.println("c.x = " + c.x + " c.y = " + c.y );
    }
}
```

実行結果

c.x = 3.0 c.y = 0.0

５．６　インスタンス変数と
インスタンスメソッド

- そのオブジェクトに作用する変数、メソッド

インスタンス変数とインスタンスメソッドの宣言

```
型名 変数;
```

インスタンス変数の宣言

```
戻り値型名 メソッド名(パラメータ) {
    // メソッド本体
}
```

インスタンスメソッドの宣言

インスタンス変数の例

```
class Bag {
    boolean flag;
    int i = 2;
}
class BagTest {
    public static void main (String args[]) {
        Bag b = new Bag();
        System.out.println(b.flag + "¥n" + b.i);
    }
}
```

実行結果

false

2

インスタンスメソッドの例(1/2)

```
class Su {
    double x;

    Su (double x) {
        this.x = x;
    }

    void bai(double x, double y) {
        this.x = this.x * x + y;
    }               // インスタンスメソッド
}
```

インスタンスメソッドの例(2/2)

```
class Suuchi {
    public static void main(String args[]) {
        Su p = new Su(5.1);
        System.out.println("p.x = " + p.x);
        p.bai(3, 0.4);    // インスタンスメソッド呼び出し
        System.out.println("p.x = " + p.x);
    }
}
```

実行結果

p.x = 5.1

p.x = 17.4

５．７　静的変数と静的メソッド

- 静的変数、静的メッソドとは、すべてのオブジェクトから共有される変数、メソッド

静的変数と静的メソッドの宣言

```
static 型名 変数;
```

静的変数の宣言

```
static 戻り値の型名 メソッド名(パラメータ) {
    // メソッド本体
}
```

静的メソッドの宣言

静的変数の例(1/2)

```
class Thing {
    static int count;       // 静的変数
    String name;            // インスタンス変数

    Thing (String name) {
        this.name = name;
        ++count;
    }
}
```

静的変数の例(2/2)

```
class StaticVariable {
    public static void main (String args[]) {
        Thing t1 = new Thing ("Bowling Ball");
        System.out.println(t1.name + " " + t1.count);
        Thing t2 = new Thing ("Ping Pong Ball");
        System.out.println(t2.name + " " + t2.count);
    }
}
```

実行結果

Bowling Ball 1

Ping Pong Ball 2

静的メソッドの例

```
class LinearEquation {
    static double solve(double a, double b) {
        return -b / a;
    }                   // 静的メソッド
}

class StaticMethod {
    public static void main (String args[]) {
        System.out.println(LinearEquation.solve(2, 2));
    }
}
```

実行結果

-1

静的初期化ブロック

- クラスがメモリにロードされるときに実行されるコードブロックを定義する事ができる

静的初期化ブロックの書き方

```
class クラス名 {
    …
    static {
        //  ステートメントのブロック
    }
}
```

静的初期化ブロックの例

```
class X {
    static int array[];
    static {
        array = new int [6];
        for (int i = 0 ; i < 6 ; i++)  array [i] = i * 2;
    }           // 静的初期化ブロック
}

class SIBlock {
    public static void main (String args[]) {
        for (int i = 0 ; i < 6 ; i++)
            System.out.print(X.array[i] + " ");
    }
}
```

実行結果

0 2 4 6 8 10

宿題

コマンドラインから入力したInt型整数を引数として、その整数自身を除く約数の和を求めるプログラムを作成せよ。