

[補助資料]Javaのクラス・メソッド・変数

[補助資料]Javaのクラス・メソッド・変数

クラス

メソッド

インスタンスメソッド

クラスメソッド

変数

インスタンス変数

クラス変数

ローカル変数

まとめ

私自身もわかっていないけれども、調べた感じこうなのかな？って言う呼び方のまとめです。

新登場のものには

```
/* コメント */
```

とアスタリスク `*` を使ったコメント文が、 以前に登場したものには

```
// コメント
```

と `//` のみのコメント文がついています。

クラス

```
/* クラス */  
public class Hoge1 {}
```

言わずもがなですね。説明は省きます。

メソッド

```
// クラス
public class Hoge1 {

    /* メソッド */
    public void Fuga1() {
        System.out.println("Hello World");
    }

}
```

これも言わずもがなですね。

インスタンスメソッド

クラスを **new** することで使えるようになるメソッド。 **new** して作成したものをインスタンスと呼び、行為はインスタンスの生成と言う。このインスタンスごとに生成されるメソッドである。

```
// クラス
public class Hoge1 {

    /* メソッド 〃 インスタンスメソッド */
    public void Fuga1() {
        System.out.println("Hello World");
    }

}
```

```
public class Hoge2 {
    Hoge1 h1 = new Hoge1(); /* インスタンスの生成 */
    h1.Fuga1(); /* インスタンスを生成したことで呼び出せる */
}
```

クラスメソッド

static をつけて定義したもの。クラスのインスタンスを生成しなくても呼び出すことができる。

```
// クラス
public class Hoge1 {

    // メソッド ㇿ インスタンスメソッド
    public void Fuga1() {
        System.out.println("Hello World");
    }

    /* メソッド ㇿ クラスメソッド */
    public static void Fuga2() {
        System.out.println("Hello world");
    }
}
```

```
public class Hoge2 {
    Hoge1 h1 = new Hoge1(); // インスタンスの生成
    h1.Fuga1(); // インスタンスを生成したことで呼び出せる

    Hoge1.Fuga2(); /* インスタンスを生成しなくても呼び出せる */
}
```

変数

```
// クラス
public class Hoge1 {

    public int i = 10; /* 変数 */

}
```

これだって言わずもがなですね.

インスタンス変数

クラスのインスタンスを生成することで使用できる変数. インスタンスごとに固有の値を持つ.

```
// クラス
public class Hoge1 {

    public int i = 10; /* 変数 ㇿ インスタンス変数 */

}
```

```

public class Hoge2 {
    Hoge1 h1_1 = new Hoge1(); // インスタンスの生成
    Hoge1 h1_2 = new Hoge1();

    /* それぞれのインスタンスのiに数値を代入*/
    h1_1.i = 1;
    h1_2.i = 2;

    /* それぞれのインスタンスごとに固有の値を持つ */
    System.out.println(h1_1.i); //output: 1
    System.out.println(h1_2.i); //output: 2
}

```

クラス変数

static をつけることで、インスタンスの生成をしなくとも使用できる変数。クラスで1つの値を持つ。インスタンスごとに固有の値にはならない。

```

// クラス
public class Hoge1 {

    public int i1 = 10; // 変数 ｝ インスタンス変数

    public static int i2 = 10; /* 変数 ｝ クラス変数 */

}

```

```

public class Hoge2 {
    Hoge1 h1_1 = new Hoge1(); //インスタンスの生成
    Hoge1 h1_2 = new Hoge1();

    /* それぞれのインスタンスのi2に数値を代入*/
    h1_1.i2 = 1;
    h1_2.i2 = 2;

    /* h1_2.i2 = 2 で代入した値を、h1_1.i2 も持っている */
    System.out.println(h1_1.i2); //output: 2
    System.out.println(h1_2.i2); //output: 2
}

```

インスタンスごとに値を持つのではなく、クラスごとに値を持つため、片方のインスタンスが値を書き換えると、もう片方のインスタンスも書き換わった後の値が参照される。

ローカル変数

メソッド内でのみ使用できる変数。

```

public class Hoge1 {

    public int i1 = 10; // 変数 ㊟ インスタンス変数

    public static int i2 = 10; // 変数 ㊟ クラス変数

    // メソッド ㊟ インスタンスメソッド
    public void Fuga1() {
        String s = "Hello world"; /* ローカル変数 */
        System.out.println(s);
    }

}

```

```

public class Hoge2 {
    Hoge1 h1 = new Hoge1();

    /* メソッド内のsを使って, Hello world と出力される */
    h1.Fuga1(); //output: Hello World

    /* sはローカル変数のため, メソッド外からは呼び出すことができない */
    System.out.println(h1.s); // Error!(Exception)
}

```

まとめ

```

// クラス
public class Hoge1 {

    public int i1 = 5; // インスタンス変数

    public static int i2 = 10; // クラス変数

    // インスタンスメソッド
    public void print() {
        String s = "i1,i2 = "; // ローカル変数
        System.out.println(s + i1 + "," + i2);
    }

}

```

```

// クラス
public class Hoge2 {

    // クラスメソッド(普段使うmainもクラスメソッド)
    public static void main(String[] args) {
        // インスタンスの生成
    }
}

```

```
Hoge1 h1_1 = new Hoge1();
Hoge1 h1_2 = new Hoge1();

h1_1.i1 = 3;
h1_1.i2 = 5;
h1_1.print(); // output: i1,i2 = 3,5

h1_2.i1 = 7;
h1_2.i2 = 9;
h1_2.print(); // output: i1,i2 = 7,9

h1_1.print(); // output: i1,i2 = 3,9
}

}
```