

/* クラスとインスタンスのまとめ

* 色々な利用方をまとめてみました。実行結果を見ながら順を追って動きを確認してください。

* コンパイルは InstanceTest、実行は Execute

*/

class InstanceTest{

// インスタンスのみで使えるフィールド(インスタンスフィールド)

String instanceVariable;

// インスタンスがなくても使えるフィールド(クラスフィールド)

static String staticVariable;

// 引数ありコンストラクタ

public InstanceTest(String Value) {

 this.instanceVariable = Value;

 // this.は省略してもコンパイラが this を自動記入する。

}

// インスタンスメソッド(非クラスメソッド)の宣言(インスタンス必要)

public void print() {

 // インスタンスフィールドの参照

 System.out.println(instanceVariable + "%t インスタンスメソッドの呼び出し");

 // インスタンスメソッドからクラスフィールドの参照はできる

 System.out.println(staticVariable); // 初期値の null が表示される

} // print

// クラス(スタティック)メソッドの宣言(インスタンス不必要)

public static void staticPrint() {

 // クラスフィールドの参照

 System.out.println(staticVariable + "%t クラスメソッドの呼び出し");

 // クラスメソッドからインスタンスフィールドは参照できない(コンパイルエラー)

 // System.out.println(instanceVariable);

} // staticPrint

} // InstanceTest クラス

* 次ページに実行クラスあり

// 実行クラス

```
class Execute {  
    /* コンストラクタを一つも宣言しなかった場合、デフォルトコンストラクタを  
     * 下記のようにコンパイル時にコンパイラが自動で記入する。  
     * Execute() {  
     *     super();  
     * }  
     */  
    public static void main(String[]args) {  
        /* デフォルトコンストラクタが存在するので自クラス(Instance クラス)の  
         * コンストラクタを呼び出してもコンパイルエラーにならない。 */  
        Execute ex = new Execute();  
  
        // InstanceTest クラスのインスタンス生成  
        InstanceTest itt = new InstanceTest("インスタンスフィールドの参照");  
  
        // インスタンスメソッド(要インスタンス)の呼び出し  
        itt.print();  
        /* インスタンスを利用しないと呼び出せない(コンパイルエラー)  
         * print()や InstanceTest.print()では不可*/  
  
        // InstanceTest クラスのクラスフィールドに値を代入  
        InstanceTest.staticVariable = "クラスフィールドの参照";  
  
        // クラスメソッドの呼び出し  
        InstanceTest.staticPrint();  
  
        // インスタンスを利用しても呼び出せる  
        itt.staticPrint();  
  
    }// main  
  
} // Execute クラス
```