



# Javaゼミ六回目

---

6.7 変数の修飾子

6.8 コンストラクタの修飾子

6.9 メソッドの修飾子

6.10 ObjectクラスとClassクラス

発表日:5月24日 発表者:加藤 友宏



# 変数の修飾子(1)

- 変数宣言の直前に指定できる修飾子
- 修飾子は 7種類ある

キーワード	意味
final	定数として使える変数
private	同じクラスのコードからしかアクセスできない変数
protected	サブクラスまたは同じパッケージ内のコードからしかアクセスできない変数
public	ほかのクラスからアクセスできる変数
static	インスタンス変数ではない変数
transient	クラスの永続的な状態の一部ではない変数
volatile	不意に値が変更されることがある変数



## 変数の修飾子(2)

---

- 一部の修飾子は、同時に使えない。  
public,protected,privateは、3つのうち一つのみ
- 修飾子を指定しないと、非final,非transient,非volatileなインスタンス変数になる。
- 同じパッケージ内のコードからしかアクセスできない



# 変数の修飾子の例

finalを用いて、値を変更できないxを使ったプログラム

```
class L {  
    static final int x = 5;  
}
```

```
class StaticFinal {  
    public static void main(String args[]) {  
        System.out.println(L.x);  
        //L.x = 10;  
    }  
}
```

L.xは、finalに指定されているので変更できない。

実行結果

5



# コンストラクタの修飾子

- コンストラクタの宣言に指定できる修飾子
- 修飾子は3種類ある。

キーワード	意味
private	同じクラスのコードからしか アクセスできないコンストラクタ
protected	サブクラスまたは同じパッケージ内のコードからしか アクセスできないコンストラクタ
public	ほかのクラスからアクセスできるコンストラクタ

- ほかの修飾子と同時に使えない。
- 指定しなければ、同じパッケージ内のコードからしかアクセスできない。



# コンストラクタの修飾子の例

publicとprivateを用いた例

```
class Person {
    String name;
    int age;

    public Person(String name, int
        age) {
        this.name = name;
        this.age = age;
    }

    private Person() {
    }
}
```

```
class PrivateConstructor {
    public static void main(String args[]) {
        // publicコンストラクタは呼び出せる
        Person p1 = new Person("John", 30);
        System.out.println(p1.name);
        System.out.println(p1.age);

        // privateコンストラクタは呼び出せない
        // Person p2 = new Person();
    }
}
```

実行結果

John  
30



# メソッドの修飾子(1)

- メソッドの宣言に指定できる修飾子。
- 修飾子は、8種類ある。

キーワード	意味
abstract	このクラスでは実装しないメソッド
final	オーバーライドできないメソッド
native	Javaのバイトコードではなく、 ホストCPUで使われるマシン語で実装されるメソッド
private	同じクラス内のコードからしか呼び出されないメソッド
protected	サブクラスまたは同じパッケージ内のコード からしかアクセスできないメソッド
public	ほかのクラスからアクセスできるメソッド
static	インスタンスメソッドではないメソッド
synchronized	実行を開始するときにロックを取得メソッド



## メソッドの修飾子(2)

---

- abstractメソッドを含むクラスは、クラスをabstractで宣言する必要がある。
- public,protected,privateは同時に使えない。
- synchronized修飾子は、マルチスレッドのプログラムを作成するときに必要。
- 修飾子を指定しないと、同じパッケージ内のコードからしかアクセスできないメソッドになる。



# メソッドの修飾子の例

```
class Singleton {
    static Singleton singleton;
    private Singleton() {
    }
    public static Singleton getInstance() {
        if (singleton == null)
            singleton = new Singleton();
        // singletonオブジェクトがまだないと
        // きには作成
        return singleton;
        // singletonオブジェクトを返す
    }
}
```

```
class SingletonDemo {
    public static void main(String args[]) {
        Singleton s1 = Singleton.getInstance();
        Singleton s2 = Singleton.getInstance();
        if (s1 == s2)
            System.out.println("Equal");
        else
            System.out.println("Not equal");
    }
}
```

実行結果

Equal



# Objectクラス

- Javaクラス階層の頂点。Objectクラスに定義された状態と動作は他の全てのクラスに継承される
- Objectクラスのメソッド

・equals()メソッド

```
boolean equals(Object obj)
```

・2つの変数が同じオブジェクトを参照しているか調べる

・getClass()メソッド

```
Class getClass()
```

・現在のオブジェクトのClassオブジェクトを返す

・toString()メソッド

```
String toString()
```

・現在のオブジェクトを表す文字列を返す



# Classクラス

---

- 実行中のプログラム内のクラス、インターフェイスに関する情報がカプセル化。
- 基本データ型とvoid、特定の次元を持つ所定のデータ型の配列にClassオブジェクトがある



# Classクラス

---

- Classクラスのメソッド

- ・getName()メソッド

データの名前を返す

```
String getName()
```

- ・getSuperclass()メソッド

現在のクラスのスーパークラス  
のClassオブジェクトを返す

```
Class getSuperclass()
```

- ・forName()メソッド

```
Static Class forName(String className)  
throws ClassNotFoundException
```

名前指定されたクラスのClassオブジェクトを返す



# ObjectクラスとClassクラスの例

---

Double型のオブジェクトに関する有益な情報を表示するプログラム

```
class ObjectDemo {  
    public static void main(String args[]) {  
        Double obj = new Double(8);  
        Class cls = obj.getClass();  
        System.out.println(cls);  
    }  
}
```

実行結果

```
class java.lang.Double
```



# 練習問題 1

---

以下のようなクラスがあるとします。

```
1. class Widget extends Thingee{
2.     static final int maxWidgetSize = 40;
3.     static String Title ;
4.     public Widget ( int mx, String T ){
5.         maxWidgetSize = mx ;
6.         Title = T;
7.     }
8.     //ほかのコード
9. }
```

このコードをコンパイルし、次に示すコードでWidgetオブジェクトを作成するプログラムを実行しようとする、どうなるか。理由も述べよ

```
Widget myWidget = new Widget ( 50, "Bigger Widget" );
```



## 練習問題 2

---

二つのクラスがある。クラスAのメソッドmethodAの1行目の下線部分に指定できるものを下から選べ。

[クラスA]

```
public class A {  
    _____ void methodA(){  
        System.out.println("A");  
    }  
}
```

[クラスB]

```
public class B extends A {  
    protected void methodA(){  
        System.out.println("B");  
    }  
}
```

A . private    B . protected    C . public    D . 何も指定しない



## 練習問題3

---

- abstractクラスAnimalに、3種類のDog, Cat, Turtleサブクラスを作成せよ。  
また、それぞれの動物の値段をあらわす変数をfinalを使って宣言し、すべての動物の名前と値段の合計を表示せよ。  
表示例: DogとCatとTurtleの合計は6400円  
ObjectのtoString()メソッドをオーバーライドすること。