



# 独習 Java (第3版)

7.5 パッケージ

7.6 クラスパス

7.7 importステートメント

7.8 アクセス制御とパッケージ



## 7.5 パッケージ (1)

- パッケージ

クラスやインターフェイスの集まり

クラスとインターフェイスをパッケージに割り当て  
packageステートメント

package packageName; packageNameはパッケージ名

packageステートメントを指定しない場合  
既定のパッケージが割り当てられる

ソースファイルの最初の行に挿入  
複数の指定はコンパイルエラー



## 7.5パッケージ(2)

- 複数のパッケージの階層構造

例 `package engineering.electrical.signals;`

指定したパッケージは

そのパッケージ名に一致するディレクトリ下に存在させる。

`engineering\electrical\signals`というディレクトリ下に作る。

# 7.5 パッケージ (3)

- Javaのパッケージ

パッケージ	説明
java.applet	アプレットを作成するためのクラス群
java.awt	グラフィカルユーザーインターフェイスを作成する AWT(Abstract Window Toolkit)のクラス群
java.awt.event	AWTコンポーネントからのイベントを処理するクラス群
java.io	入出力動作を扱うクラス群
java.lang	Javaの中核的な機能を提供するクラス群
java.net	ネットワーク機能を提供するクラス群
java.util	ユーティリティ機能を提供するクラス群

## 7.5 パッケージ (4)

- 4つのソースファイルに別々に定義したクラスをpというパッケージにまとめる方法

p\PackageDemo.javaのソースファイル

```
package p;

class PackageDemo{
public static void main(String args[]){
A a=new A();
a.a1();
B b=new B();
b.b1();
C c=new C();
c.c1();
}
}
```

p\A.javaのソースファイル

```
package p;
class A{
void a1(){
System.out.println("a1");
}
}
```

p\B.javaのソースファイル

```
package p;
class B{
void b1(){
System.out.println("b1");
}
}
```

p\C.javaのソースファイル

```
package p;
class C{
void c1(){
System.out.println("c1");
}
}
```



## 7.5パッケージ (5)

< 手順 >

1. 作成した4つのソースファイルをpディレクトリに格納
2. pの親ディレクトリで4つのソースファイルをコンパイル  
例 `javac p\*.java`

または ディレクトリpでコンパイルする場合

例 `javac PackageDemo.java A.java B.java C.java`



## 7.5パッケージ (6)

3. pの親ディレクトリで実行

例 `java p.PackageDemo`

p.PackageDemoを完全修飾クラス名という

完全修飾クラス名とはパッケージ名とクラス名を合わせたもの

<出力結果>

a1

b1

c1



## 7.6 クラスパス ( 1 )

### ■ クラスパス

JavaコンパイルやインタプリタなどのJDK/JREツールに対して.classファイルの検索先を指示するための環境変数

ディレクトリ、.jarファイル、zipファイルが順番に格納されてる

Windowsプラットフォームの場合の環境変数の設定

例 `set CLASSPATH=c:\project1;c:\project2`

環境変数の現在の設定の確認

例 `echo %CLASSPATH%`

## 7.6 クラスパス (2)

- JDK/JREツールが.classファイルを探す手順

例 環境変数を以下のように設定したとする

```
set CLASSPATH=c:\chemistry;c:\physics;c:\calculus;c:\graph
```

logarithmic.Plotというクラスを見つける場合

1. カレントディレクトリ下のlogarithmic\Plot.classファイル
2. c:\chemistry\logarithmic\Plot.class
3. c:\physics\logarithmic\Plot.class
4. c:\calculus\logarithmic\Plot.class
5. c:\graph\logarithmic\Plot.class
6. Javaクラスライブラリ

もしファイルが見つからなかったらコンパイラからNoClassDefFoundErrorメッセージが表示される。



## 7.7 importステートメント(1)

- Importステートメント

```
import fullyQualifiedTypeName;  
import packageName.*;
```

1つ目の構文は fullyQualifiedTypeName にクラス、インターフェイスを指定  
短い名前アクセス可能

例 `import java.awt.event.MouseEvent;`

`java.awt.event.MouseEvent`を完全修飾名という。

完全修飾名の`java.awt.event.MouseEvent`を`MouseEvent`だけで代用できる



## 7 . 7 importステートメント(2)

例 `import java.awt.event.*;`

java.awt.event内の全ての型を短い名前で参照できる。

サブパッケージを取り込めない例

`import java.awt.*;`

この場合java.awt.event内のパブリックな型は取り込めない。

`import java.awt.event.*;`とする。

# 7 . 7 importステートメント ( 3 )

importステートメントのプログラム

ImportDemo.java,R.java,S.javaの  
3つのソースプログラムを作成する。

ImportDemo.javaの中身

```
package q;  
import r.*;  
import s.*;
```

```
class ImportDemo{  
public static void main(String args[]){  
R r=new R();  
r.r1();  
S s=new S();  
s.s1();  
}  
}
```

R.Javaの中身

```
package r;  
public class R{  
public void r1(){  
System.out.println("r1");  
}  
}
```

S.Javaの中身

```
package s;  
public class S{  
public void s1(){  
System.out.println("s1");  
}  
}
```

## 7 . 7 importステートメント(4)

- 3つのパッケージの構築

- ・親ディレクトリに切り替えてコンパイルする。

```
javac q\*.java
```

```
javac r\*.java
```

```
javac s\*.java
```

または

```
javac q\*.java r\*.java s\*.java (ただし親ディレクトリが全て同じ場合)
```

- ・実行はqの親ディレクトリで行う

```
Java q.ImportDemo
```

出力結果

```
r1
```

```
s1
```

## 7.8 アクセス制御とパッケージ(1)

- public,protected,privateの各キーワードの意味

キーワード	説明
public	全てのクラスにアクセスを許可する
protected	同じパッケージ内のコード、または別のパッケージ内のサブクラスにのみアクセス許可する。
private	同じクラス内のコードのみアクセス許可する。

これらのキーワードは同時に複数指定できない。

1つも指定しなければ既定で同じパッケージ内のコードにしかアクセスが許可されない。

## 7.8 アクセス制御とパッケージ (2)

public,protected,privateの違い  
e\E.javaのソースコード  
package e;

```
public class E{  
    public int e1=11;  
    protected int e2=22;  
    private int e3=33;  
}
```

f\F.javaのソースコード  
package f;  
import e.\*;

```
public class F extends E{  
    public void display(){  
        System.out.println(e1);  
        System.out.println(e2);  
        //System.out.println(e3);  
    }  
}
```

g\G.javaのソースコード  
package g;  
import e.\*;

```
public class G{  
    public void display(){  
        E e=new E();  
        System.out.println(e.e1);  
        //System.out.println(e.e2);  
        //System.out.println(e.e3);  
    }  
}
```



## 7.8 アクセス制御とパッケージ (3)

h\ProtectedDemo.javaのソースコード

```
package h;  
import f.F;  
import g.G;
```

```
class ProtectedDemo{  
public static void main(String args[]){  
F f=new F();  
f.display();  
G g=new G();  
g.display();  
}  
}
```

・これらの4つのパッケージを構築

```
javac e\*.java f\*.java g\*.java h\*.java
```

・実行を行う

```
java h.ProtectedDemo
```

出力結果

11

22

11



# 練習問題

## 問題1

本とCDを買った合計金額を表示せよ。ただし、3つのソースファイルに別々に定義したクラスをp.q.rというパッケージにまとめる。

1つ目のクラスは本の金額、作品名、著者名を表示し、金額を返す。2つ目のクラスはCDの金額、作品名、アーティスト名を表示し、金額を返す。3つ目のクラスは任意のデータを1つ目、2つ目のクラスに渡し、金額を受け取り、その合計金額を表示する。

また、このパッケージの.classファイルはc\a.p.q.rに格納するときこのパッケージに含まれるクラスとインターフェイスをJDK/JREツールから検索できるようにするにはCLASSPATH環境変数をどのように設定すればよいか答えよ。

## 問題2

問題1の3つのソースファイルのパッケージをそれぞれp,r,sにして作成せよ。