第24章 パッケージ

目次

- パッケージ
- インポート

パッケージとは

クラスをまとめて、分類するための仕組み。
クラスを機能や役割別に分類すれば管理しやすくなる。

パッケージのルール

- Javaのクラスは、必ずパッケージに属さなければいけない。
- Javaの仕様上、パッケージに属していないクラスは存在しない。
- パッケージを指定しない場合、そのクラスは暗黙的に
  決められたパッケージ（デフォルトパッケージ）に属する。

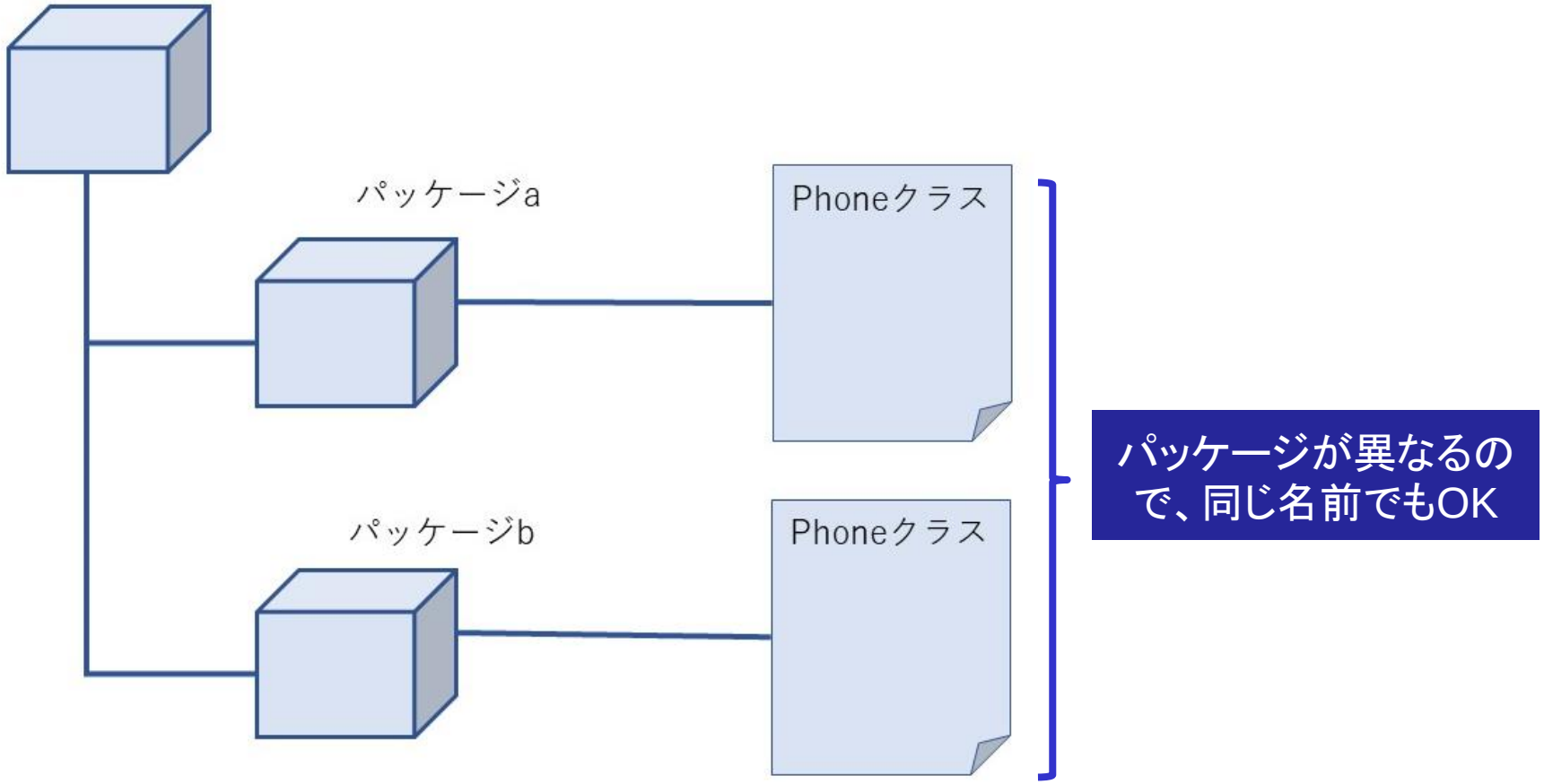
パッケージはクラスを分類するために使用します。
クラスは必ずパッケージに属します。
パッケージが指定されていないクラスはデフォルトパッケージに属します。

パッケージ名でクラスを見分ける

通常、 1つのシステム内に同じ名前のクラスを
複数作成することはできない。
パッケージが異なれば、同名のクラスを
複数作成することができる。

名前空間:
パッケージのように、クラス名等の識別子を
重複させないための仕組みのこと。

パッケージ名でクラスを見分ける



異なるパッケージのクラスを使う

SampleクラスとPhoneクラスが異なるパッケージに
存在している状態でクラス名のみを記述した場合、
そのクラスは同じパッケージ内に存在しているクラスと認識される。
しかし、同じパッケージ内にPhoneクラスが存在しないため、
コンパイルエラーが発生している。

```
package a;

public class Sample {
    public static void main(String[] args) {
        Phone phone = new Phone();
    }
}
```

パッケージaにPhoneクラスがないため、コンパイルエラー

異なるパッケージのクラスを使う

異なるパッケージのクラスを使うには…

① アクセスされるクラスのアクセス修飾子をpublicにする。
② 異なるパッケージのクラスにアクセスする際に、
　パッケージ名を含めてクラス名を指定する（完全修飾名）。

【Sample2401 異なるパッケージのクラスを使う】を作成しましょう

【Sample2401 異なるパッケージのクラスを使う】を作成しましょう

※eclipseでパッケージの中にパッケージを作成する方法

①親となるパッケージを作成

②子パッケージを作る際、パッケージ名を「親パッケージ.子パッケージ」と入力

なお、ここでは便宜上、子パッケージと表現しましたが、パッケージの中のパッケージのことを

「サブパッケージ」と呼びます。

Sample2401のポイント

lesson24.b.Sample2401クラスから
lesson24.a.Phone2401クラスにアクセスしたい場合、
異なるパッケージのため、パッケージ名も含めた
完全修飾名を記述する。

```
lesson24.a.Phone2401 phone = new lesson24.a.Phone2401();
```

異なるパッケージ内のクラスは完全修飾名で記述するとアクセスできる。

インポートとは

対象のクラスを取り込んで、
クラス内のフィールドやメソッドを直接利用できるようにすること。

異なるパッケージ内のクラスにアクセスするたびに、
完全修飾名で記述するのは不便である。
インポートするための宣言文（インポート文）を記述することで、
クラス名のみでアクセスできるようになる。

```
import パッケージ名.クラス名;
```

インポートとは

全修飾名で呼び出していたソースコードを、
import文を活用したものに書き換えた場合

```
lesson24.a.Phone2401 phone = new lesson24.a.Phone2401();
```

```
import lesson24.a.Phone2401;
・・・
Phone2401 phone = new Phone2401();
```

【Sample2402 import文を宣言する】を作成しましょう

Sample2402のポイント

import文を記述することで、クラス名のみを記述して
Phoneクラスにアクセスできる。
記述量が減るので、可読性向上に繋がる。

```
import lesson24.a.Phone2401;
・・・
Phone2401 phone = new Phone2401();
```

インポートをすると、クラス名のみで別パッケージのクラスが利用できます。

複数のクラスのインポート

同じパッケージ内の複数のクラスをインポートする場合は、
クラスごとにimport文を記述する。

```
import java.io.BufferedReader;
import java.io.IOException;
・・・
```

複数のクラスのインポート

同じパッケージ内のクラスに関するimport文は、
下記のように「*（アスタリスク）」を使用して1行にまとめられる。

```
import java.io.*
```

インポートしたいパッケージの最後に「*」と記述する。
「*」はワイルドカードであり、「対象のパッケージ直下の
すべてのクラスをインポートする」という意味。
上記のimport文では「java.io」直下のクラスを
すべてインポートする。

複数のクラスのインポート

指定したパッケージ直下にあるサブパッケージ内のクラスは
インポート対象外。対象は「*」が記載された階層のクラスのみ。

ソースコードを作成する際の作法上、
「*」を使用したインポートは以下の理由で非推奨。

- そのクラスで具体的にどのクラスを呼び出しているかが
  分からなくなるため。
- 名前の衝突が発生する可能性があるため。

クラスライブラリのインポート

クラスライブラリもパッケージごとに管理されているクラスである。
import文を記述して対象のクラスにアクセスする。

java.ioパッケージのBufferedReaderクラスのインポート文

```
import java.io.BufferedReader;
・・・
```

クラスライブラリのインポート

java.langパッケージのクラスは、利用する機会が多いため、
例外的にインポートしなくてもクラスにアクセスできる。
（Stringクラスの文字列を使用する際、
import文を記述していなかったのは、そのためである。）

パッケージ名	パッケージに含まれるクラス	主なクラス例
java.lang	基本的なクラス	Integer String
java.io	入出力関連のクラス	BufferedReader InputStreamReader
java.util	ユーティリティ関連のクラス	ArrayList<E> Date
java.math	数値の演算に関するクラス	BigDecimal

章のまとめ

- package文を使ってクラスをパッケージに含めます。
- クラスの先頭にpublicをつけると、異なるパッケージから利用できます。
- 異なるパッケージのクラスを利用するには「パッケージ.クラス名（完全修飾名）」と記述します。
- インポート文を利用すると、異なるパッケージのクラスをクラス名のみで利用できます。