

第23章 オブジェクト指向の三大要素

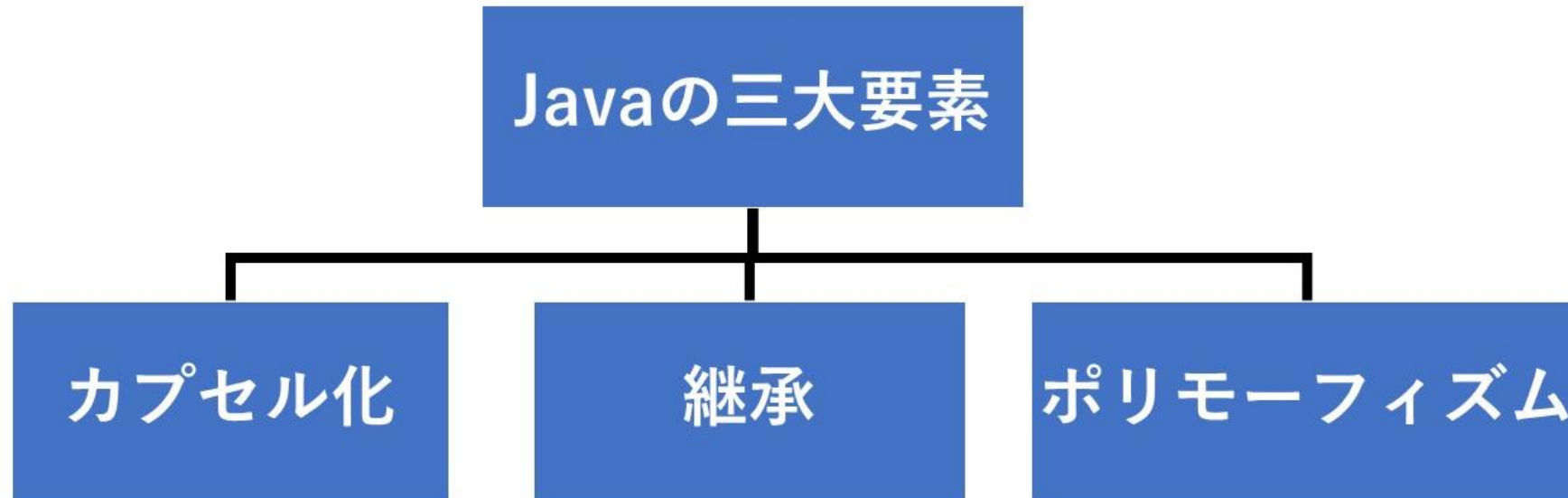
目次

- 三大要素とは

三大要素とは

オブジェクト指向の3つの重要な要素。

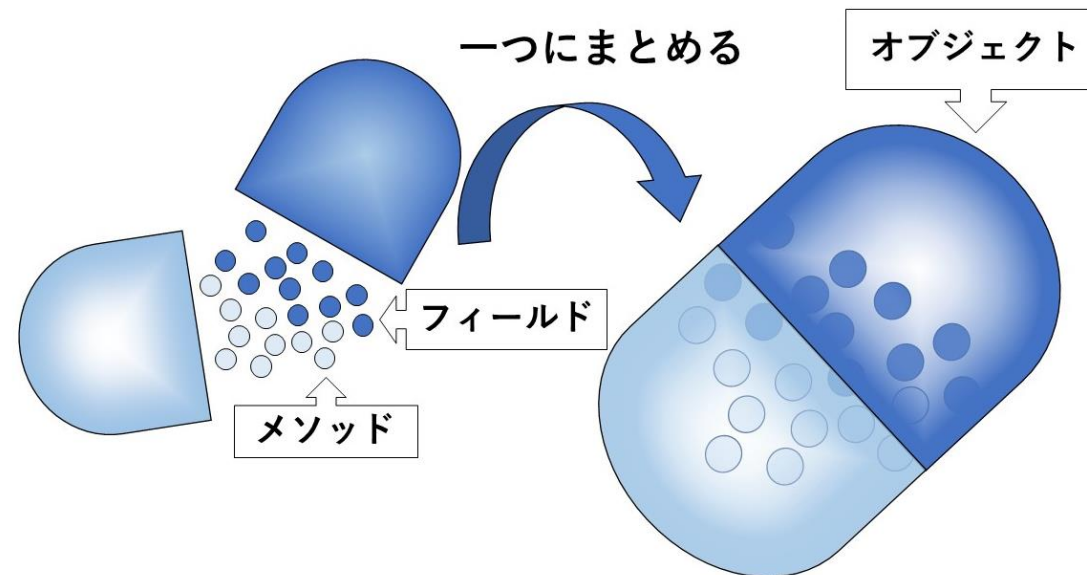
「カプセル化」、「継承」、「ポリモーフィズム(多態性)」



カプセル化

カプセル化とは...

フィールドとメソッドを1つのクラスにまとめ、保護したいメンバにprivateをつけ、外からのアクセスを制限する仕組みのこと。



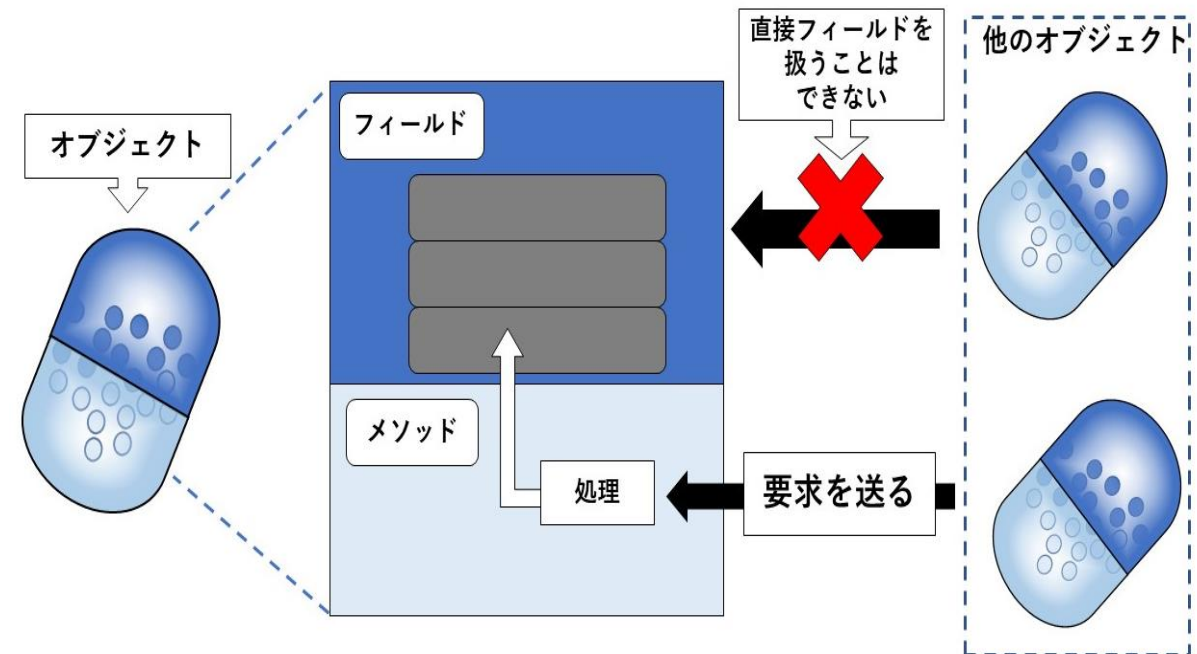
カプセル化

カプセル化を用いることで、
必要なデータにだけアクセス可能となり、
保守性の向上やバグに強いプログラムの作成に繋がる。

カプセル化

カプセル化されたクラスでは、フィールドにはprivateメンバメソッドにはpublicメンバを付与するのが一般的。

カプセル化されたクラスの情報にアクセスするには、アクセサメソッド（getterメソッド、setterメソッド）を介す必要がある。



カプセル化

継承関係にあるクラスの場合、サブクラスからのアクセスを可能にする場合はスーパークラスのフィールドの修飾子をprivateでなく、protectedに指定する。

| 修飾子名 | アクセスを許可する範囲 | 記述できる場所 |
|-----------|--------------------------|------------------------|
| private | 同一クラスのみ | フィールド、メソッド、コンストラクタ |
| 指定なし | 同一パッケージ内のクラス | フィールド、メソッド、コンストラクタ、クラス |
| protected | 同一パッケージ内のクラスか、自身を継承したクラス | フィールド、メソッド、コンストラクタ |
| public | すべてのクラス | フィールド、メソッド、コンストラクタ、クラス |

継承

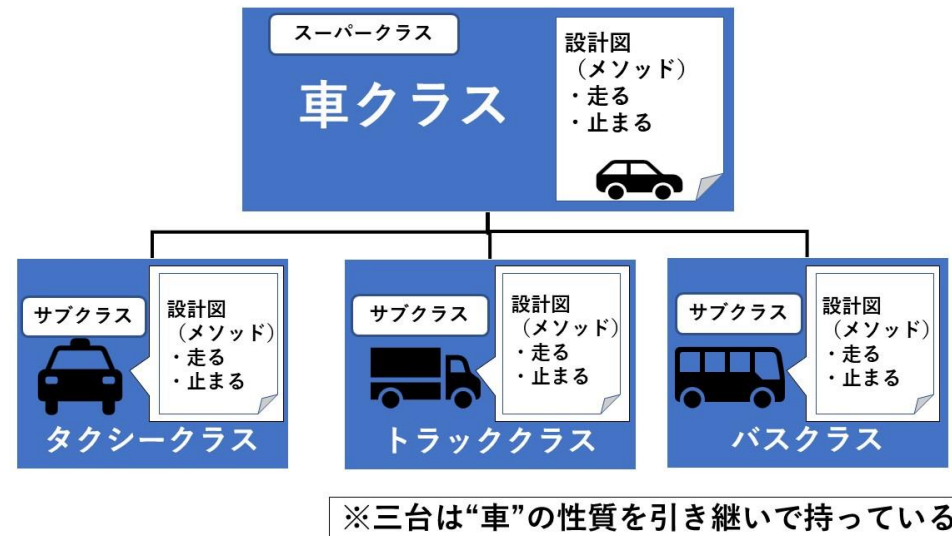
継承とは、
「あるクラスのメンバを他のクラスに引き継がせる」仕組みのこと。

継承機能を用いることでプログラムの再利用性を高め、
保守・修正を容易にできる。

継承（例）車

タクシー、トラック、バスはそれぞれ車の種類が異なる。

車を走らせる機能や車を止める機能はすべての車に共通している。共通している機能をまとめて車クラスとして管理し、車クラスを継承することで同じ機能を再利用し、無駄を省ける。



ポリモーフィズム(多態性)

同じ名前のメソッドでも、オブジェクトに応じて、それぞれ異なる振る舞いを持たせることができる仕組みのこと。

活用することで、開発者が把握すべき情報が最小限に抑えられる。
また、同一カテゴリのクラスのメソッドを統一できるため、各クラスが直感的に利用しやすくなる。

ポリモーフィズム（多態性）（例）社会人に属する職業

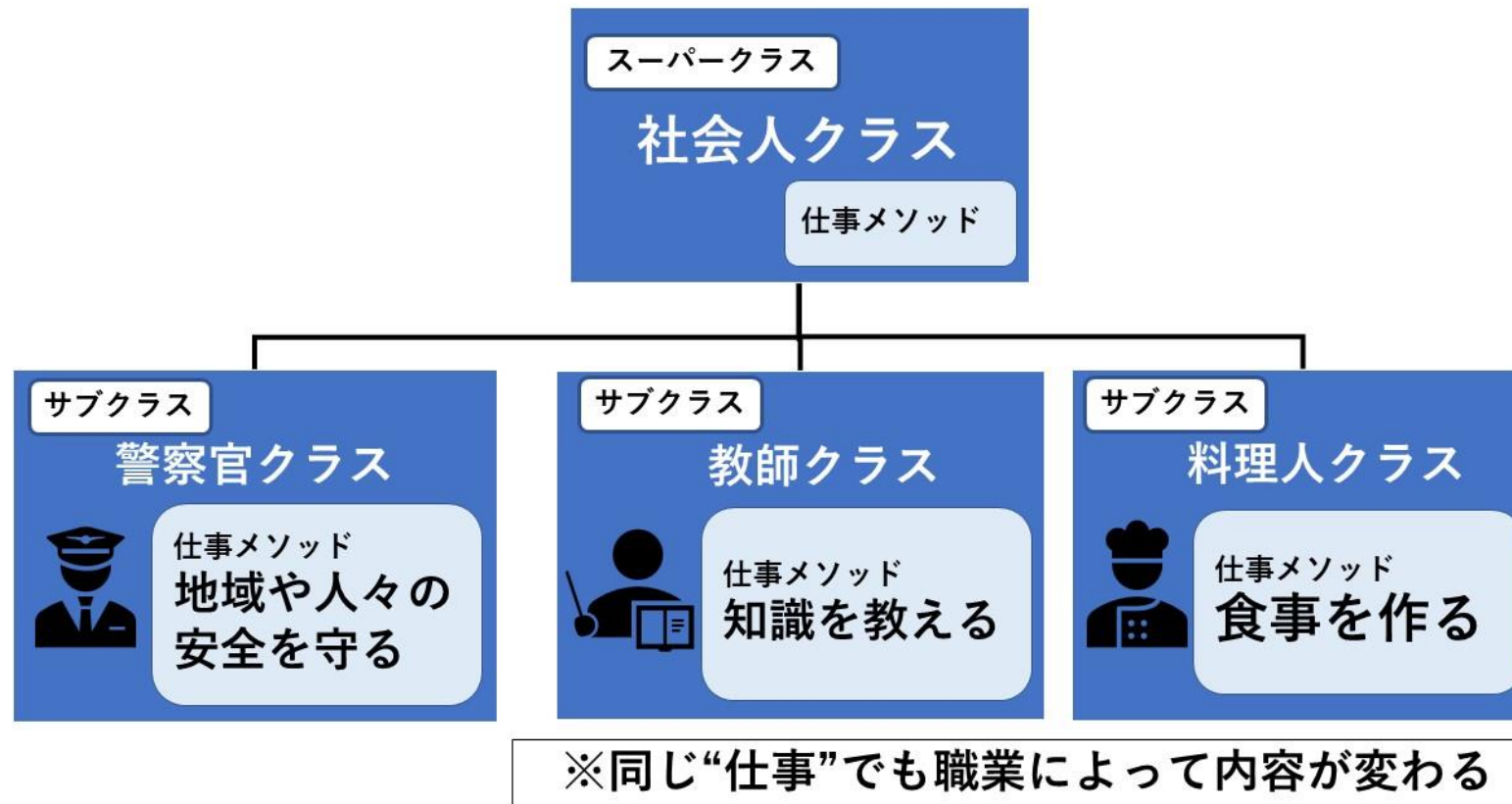
「警察官」「教師」「料理人」

各職業は1つのクラス。いずれの職業の人も「社会人」という大枠のクラスに含まれ、「仕事」という機能を行うことが主な役割。それぞれ仕事の内容（処理結果）が異なる。

- 警察官の仕事： 地域や人々の安全を守る
- 教師の仕事： 知識を教える：
- 料理人の仕事： 食事を作る

言葉としては同じ行動でも、実際に行う内容は異なるということをポリモーフィズムという。

ポリモーフィズム(多態性) (例)社会人に属する職業



ポリモーフィズム(多態性)

ポリモーフィズムを実現する仕組み①オーバーロード
一つのクラスの中に、名前が同じメソッドを
複数定義できる仕組みのこと。

本来、処理が似ているからとメソッド名に
同じ名前を付けることはできない。

引数の型、個数が違う場合は、
例外的に同じメソッドを複数定義できる。

ポリモーフィズム(多態性)

ポリモーフィズムを実現する仕組み①オーバーライド
サブクラス内にスーパークラスで定義されたメソッドと
同じシグネチャのメソッドを定義できる仕組みのこと。

クラスのフィールドなどはそのまま再利用して、
変更したいメソッドだけを再定義するなど、
メソッドの変更が必要な場合に変更箇所が少なく済む。
これにより、「再利用しつつも、変えたいところは自由に変えられる」
というポリモーフィズム(多態性)の思想を実現できる。

【Sample2301 Javaの三大要素を利用する】 を作成しましょう



Sample2301のポイント

定数を管理するためのクラス(Contstantクラス)が定義されている。

定数:

「static」と「final」の2つの修飾子が付与されたフィールドのこと。

Sample2301のポイント

実際のシステム開発では多くの定数(変更しない値)を使用する。
定数があちこちに書かれていると管理が大変である。
1つのクラスにまとめて定義して、
定数を一元管理するという工夫がよく行われる。

この用途で定義されたクラスを「定数クラス」と呼ぶことがある。

Sample2301のポイント

各クラスのメンバはprotected修飾子、
public修飾子が付与されて、カプセル化されている。

- 他のクラスから直接アクセスされるのを防ぎたい
- サブクラスからのアクセスは許可したい

2つの要件を満たすために、各クラスのフィールドには
protected修飾子が付与されている。

他のクラスからもアクセスできるように、
各クラスのメソッドはpublic修飾子が付与されている。

Sample2301のポイント

Policeクラス、Teacherクラス、ChefクラスはいずれもWorkerクラスを継承している。

各クラスはWorkerクラスのメンバをあたかも自分のメンバであるかのように利用でき、再利用性が高い作りとなっている。

```
public class Chef extends Worker {  
}
```

Sample2301のポイント

Displayクラスでは、配列に格納された各職業のオブジェクトを取得し、showIntroduction()メソッドとdoWork()メソッド呼び出している。

```
for (Worker worker : workers) {  
    worker.showIntroduction();  
    worker.doWork();  
}
```

各職業クラス内ではdoWork()メソッド内に独自の処理を実装している。doWork()メソッドを呼び出すと、クラス各の定義内容に応じて異なる処理を実行する。

ポリモーフィズムが活用された状態となっている。

章のまとめ

- Javaではオブジェクト指向の三大要素に則ったプログラミングができます。
- カプセル化により、必要なデータにのみアクセス可能なクラスを定義できます。
- 継承により、複数のクラスで共通のメソッドを再利用できます。
- ポリモーフィズムにより、同一カテゴリのクラスを直感的に利用できます。