

第21章 抽象クラス

目次

- 抽象クラス

抽象クラスとは

抽象メソッド: 処理内容を記述しないメソッド

抽象クラス: 抽象メソッドを持つクラス

具象メソッド: 具体的な処理が記述できるメソッド

具象クラス: 具象メソッドだけを持つクラス

これまで作ってきた
メソッド・クラス

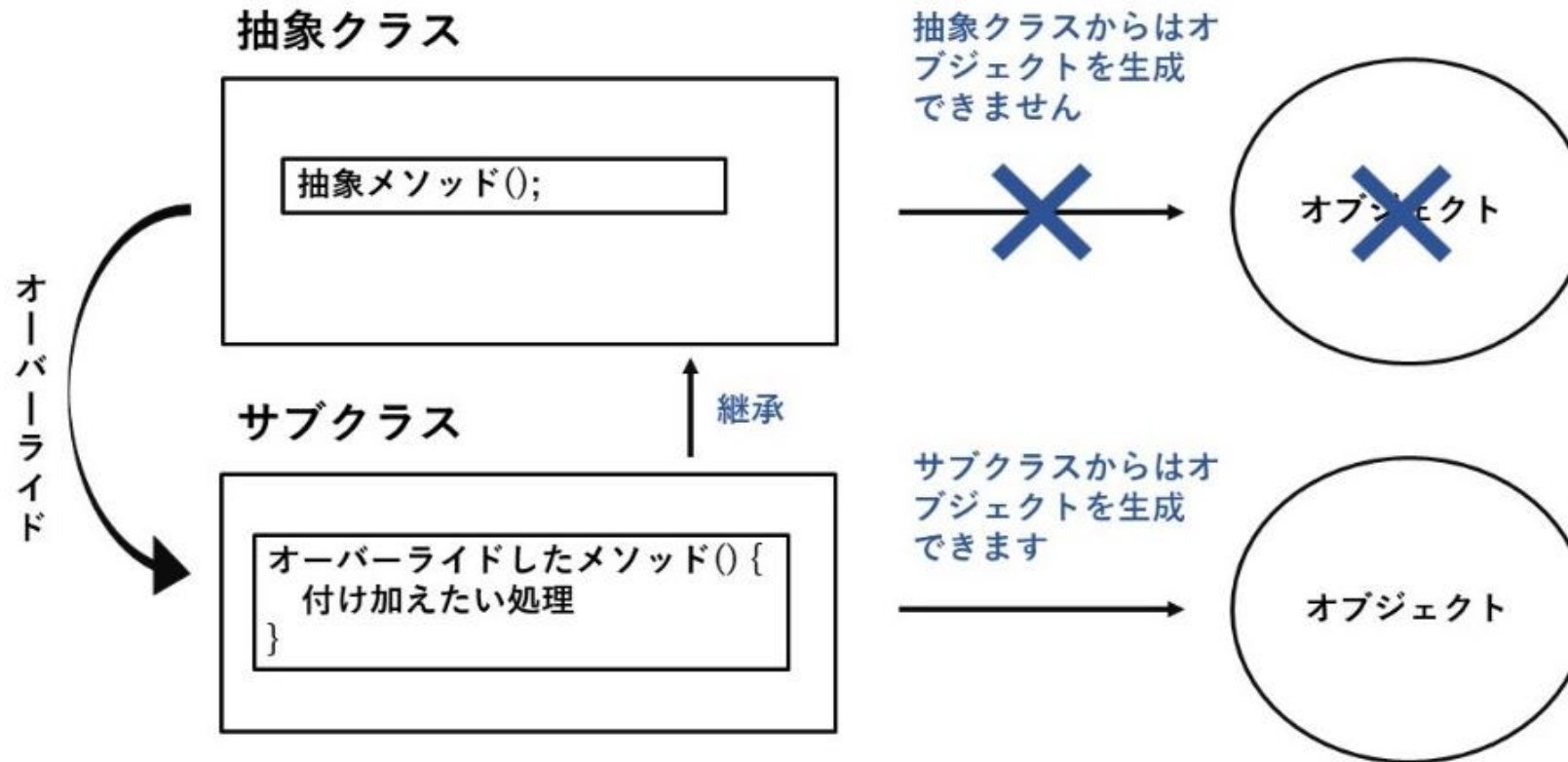
抽象クラスの特徴

- オブジェクトを生成できない
- 抽象クラスを継承して具象クラスを作成する場合、サブクラスですべての抽象メソッドを「オーバーライド」しないとコンパイルエラーとなる
- 抽象メソッドだけでなく、具象メソッドも定義できる

抽象クラスの特徴

- 抽象クラスはオブジェクトを生成できないため、単体では利用しない。
他のクラスのスーパークラスとして利用される。
抽象クラスを具象クラスに継承することで、具象クラスからオブジェクトが生成できる。
- 抽象クラスを継承したサブクラスが具象クラスの場合、抽象メソッドからオーバーライドしたメソッドは具象メソッドとして定義する。

抽象クラスの特徴



抽象クラスと抽象メソッドのメリット

抽象クラスを用いることで、クラスの定義に制約を設けることができる。

抽象クラスと抽象メソッドのメリット

類似した用途で使用するクラスを複数人のプログラマがそれぞれ作成する場合：

プログラマ各々が独自のメソッド名で処理を記述し、名前の異なるメソッドがシステム内に乱立する。

1. どのメソッドがどのような用途で使用するか把握が難しい
2. バグ修正や機能追加の際に、修正漏れが発生する
3. 各プログラムの品質に差が生じる
→システム全体の品質が低下する

抽象クラスと抽象メソッドのメリット

「ある用途のクラスを定義する際には必ず特定の抽象クラスを継承する」というルールを設けた場合：抽象クラスのサブクラス内では抽象メソッドを必ずオーバーライドする必要がある。

1. クラス内のメソッドのシグネチャが統一されるため、多態性(ポリモーフィズム)が保証される。
2. 各メソッドの把握が容易。
3. 修正や機能追加が容易。
4. メソッドの実装漏れがないクラスを定義できる。

抽象クラスと抽象メソッドの定義

abstract修飾子を付ける。

```
修飾子 abstract class クラス名 {  
    フィールド  
    修飾子 abstract 戻り値の型 メソッド名(引数リスト);  
}
```

抽象クラスと抽象メソッドの定義

抽象クラス:

abstractを付ける以外、従来のクラス定義と変わりなし

```
修飾子 abstract class クラス名 {  
  
}
```

抽象メソッド:

ブロック「{ }」を記述せずに行末に「;」を書く

```
修飾子 abstract 戻り値の型 メソッド名(引数リスト);
```

抽象クラスと抽象メソッドの定義

通常のクラス

```
class クラス名 {  
    戻り値の型 メソッド名 (引数) {  
        処理;  
    }  
}
```

抽象クラス

```
abstract class クラス名 {  
    abstract 戻り値の型 メソッド名(引数);  
}
```

【Sample2101 抽象クラスを利用する】 を作成しましょう



Sample2101のポイント

Animal2101クラスは抽象クラスであり、
抽象メソッドのbark()メソッドと
具象メソッドのgreet()メソッドが定義されている。

```
public abstract class Animal2101 {  
    abstract void bark(); } 抽象クラス  
    public void greet() {  
        System.out.println("よろしくお願ひします。"); } 具象クラス  
    }  
}
```

Sample2101のポイント

抽象クラスAnimal2201クラスを継承したPigeon2101クラスとLion2101クラスは具象クラスであるため、必ずbark()メソッドをオーバーライドする必要がある。

```
public class Pigeon2101 extends Animal2101 {  
    public void bark() {  
        System.out.println("ポッポー");  
    }  
}
```

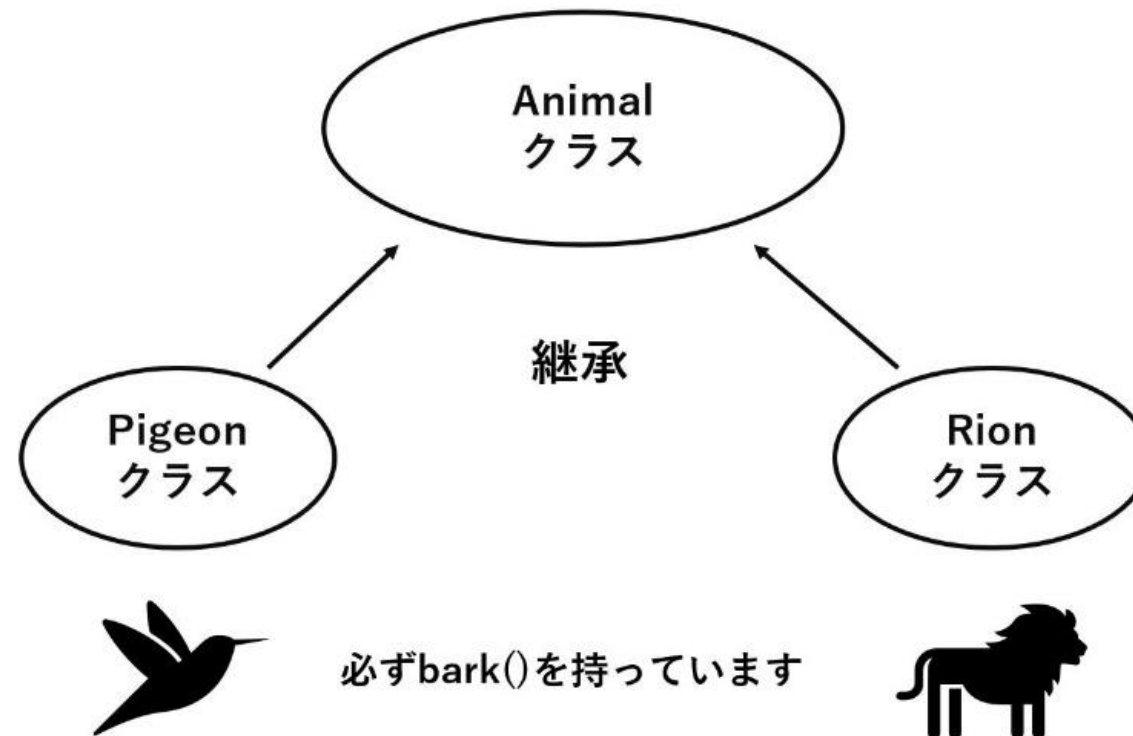
} オーバーライド

Sample2101のポイント

Animal2101クラスには、処理内容が記述されている
具象メソッドgreet()が定義されている。
抽象クラスで定義した具象メソッドは、
サブクラスに記述をしなくても呼び出せる。

```
public void greet() {  
    System.out.println("よろしくお願いします。");  
}
```


Sample2101のポイント



抽象クラスを継承した具象クラスでは、
抽象メソッドをオーバーライドする必要があります。

章のまとめ

- abstractを使って、抽象クラスを定義することができます。
- 抽象クラスは、処理を記述しない抽象メソッドを定義することができます。
- 抽象クラスのオブジェクトを生成(new)することはできません。