

オブジェクト指向言語 Javaプログラミング(2010年度) 補足資料(05/17)

桑原 寛明
情報理工学部 情報システム学科

抽象クラス

- 抽象メソッドを持つクラス
 - インスタンス化できない
- 抽象メソッド
 - 実装のないメソッド
 - 実装は継承したクラスで行う
- 継承したクラスのインスタンスが抽象メソッドを実装していることが保証できる

オブジェクト指向言語(2010年度)

2

例

```
class Circle {
    int x; int y; int r;

    void draw(Canvas c) {
        ...
    }
}
```

```
class Rectangle {
    int x; int y; int w; int h;

    void draw(Canvas c) {
        ...
    }
}
```

```
{
    Canvas canvas = new Canvas(...);
    Circle[] circles = { new Circle(...), new Circle(...), ... };
    Rectangle[] rects = { new Rectangle(...), ... };
    for (Circle c : circles) {
        c.draw(canvas);
    }
    for (Rectangle r : rects) {
        r.draw(canvas);
    }
}
```

コードが重複
図形の重なりの処理が面倒

オブジェクト指向言語(2010年度)

3

例(親クラスを導入)

```
class Shape {
    void draw(Canvas c) {
    }
}
```

drawメソッドは何もしない(何もできない)
Shapeクラスはインスタンス化してほしくない

```
class Circle extends Shape {
    int x; int y; int r;

    void draw(Canvas c) {
        ...
    }
}
```

```
class Rectangle extends Shape {
    int x; int y; int w; int h;

    void draw(Canvas c) {
        ...
    }
}
```

```
{
    Canvas c = new Canvas(...);
    Shape[] shapes = { new Circle(...), new Rectangle(...), ... };
    for (Shape s : shapes) {
        s.draw(c);
    }
}
```

CircleとRectangleを区別せず扱える

オブジェクト指向言語(2010年度)

4

例(抽象クラスの利用)

```
abstract class Shape {  
    abstract void draw(Canvas c);  
}
```

drawメソッドは継承クラスで実装
Shapeクラスはインスタンス化できない

```
class Circle extends Shape {  
    int x; int y; int r;  
  
    void draw(Canvas c) {  
        ...  
    }  
}
```

```
class Rectangle extends Shape {  
    int x; int y; int w; int h;  
  
    void draw(Canvas c) {  
        ...  
    }  
}
```

```
{  
    Canvas c = new Canvas(...);  
    Shape[] shapes = { new Circle(...), new Rectangle(...), ... };  
    for (Shape s : shapes) {  
        s.draw(c);  
    }  
}
```

例

- もしShapeクラスが抽象クラスでないとすると
 - Shapeクラスのインスタンスが作れてしまうが、以下の理由からそれは困る
 - Shapeクラスは各図形の共通要素を持つだけで、具体的な図形を表すクラスではない
 - Shapeクラスに対して図形を描画するdrawメソッドの内容は決められない
 - しかし、ShapeクラスがないとCircleクラスとRectangleクラスを完全に別々に扱わなければならない手間
- 抽象クラスとすることでインスタンス化を意図していないことを明示できる