

# 第8章 ポリモフィズムと パッケージ（問題）

## 問題8-6

```
1
2 public abstract class Foo {
3     public Foo(String data) {
4         methodA(data);
5         methodB();
6     }
7     public void methodA(String data) {
8         System.out.println(data);
9     }
10    public abstract void methodB();
11
12    public static void main(String[] args) {
13        Foo obj = new Foo("Hello");
14        obj.methodA("bye");
15    }
16
17 }
18
```

Fooクラスが抽象クラスなので、インスタンスを生成する事ができない (p.222)

## 問題8-10

```
1 class Foo3 {  
2     String name;  
3 }  
4  
5  
6 class Bar extends Foo3 {  
7     Bar(String name) {  
8         this.name = name;  
9     }  
10    void show() {  
11        System.out.println(name);  
12    }  
13    public static void main(String[] args) {  
14        Foo3 obj = new Bar("hana");  
15  
16        //A  
17        show();  
18  
19        //B  
20        obj.show();  
21  
22        //C  
23        ((Foo)obj).show();  
24  
25        //D  
26        ((Bar)obj).show();  
27    }  
28 }
```

Barクラスをインスタンス化

Bar型にキャストする事で、コンパイル・実行が成功する

## 問題8-11

```
1  class A {}  
2  class B extends A {}  
3  class C {}  
4  class Test2 {  
5      public static void main (String[] args) {  
6          A a = new C();  
7          C c = (C)a;  
8      }  
9  }  
10  
11  
12
```

class Cはclass Aを継承していない

## 問題8-13

```
1
2 interface Foo4{}
3 class Alpha implements Foo4{}
4 class Beta extends Alpha{}
5 class Delta extends Beta{}
6 public static void main(String[] args){
7     Beta x = new Beta();
8     //問題8-13
9     Foo4 f =(Delta)x;
10 }
11 }
```

Betaクラスをインスタンス化してる  
(Foo型、Alpha型、Beta型で宣言された  
変数で扱う事が可能)

### ClassCastException

- ・ もとのクラス (型) 以外のクラスにキャストしようとするとき実行時に

## 問題8-16

```
1  interface MyInter{  
2      public static final int VAL =3;  
3      void foo(int s);  
4  }  
5  
6  public class Test implements MyInter{  
7      public static void main(String[] args){  
8          int x = 5;  
9          new Test().foo(++x);  
10     }  
11     void foo(int s){  
12         s +=VAL + ++s;  
13         System.out.println("s:"+ s);  
14     }  
15  
16 }
```

インターフェイスで定義したメソッドは必ず  
publicになる

実行

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:  
MyInter から継承されたメソッドの可視性を下げることはできません  
at Test.foo(Test.java:11)  
at Test.main(Test.java:9)
```

## 問題8-16

```
1  interface MyInter{  
2      public static final int VAL =3;  
3      void foo(int s);  
4  }  
5  
6  public class Test implements MyInter{  
7      public static void main(String[] args){  
8          int x = 5;  
9          new Test().foo(++x);  
10     }  
11     public void foo(int s){  
12         s +=VAL + ++s;  
13         System.out.println("s:"+ s);  
14     }  
15  
16 }
```

実行

s:16



## 問題8-17

```
4 class A {  
5     int num =10;  
6     void func() {num++;}  
7 }  
8 class B extends A{  
9     static int num = 30;  
10    static void func() {num++;}  
11 }  
12 class Test{  
13     public static void main(String[] args) {  
14         A obj = new B();  
15         obj.func();  
16         System.out.println(obj.num);  
17     }  
18 }
```

インスタンスメソッドをstaticメソッドでオーバーライドする事は

Main.java:10: error: func() in B cannot override func() in A  
 static void func() {num++;}  
 ^  
 overriding method is static  
1 error



## 問題8-18

```
1  class Animal {  
2      public String noise(){ return "peep";}  
3  }  
4  
5  class Dog extends Animal {  
6      public String noise() { return "bark";}  
7  }  
8  class Cat extends Animal {  
9      public String noise() { return "meow";}  
10 }  
11 class Test5 {  
12     public static void main(String[] args) {  
13         Animal animal = new Animal();  
14         Cat cat = (Cat) animal;  
15         System.out.println(cat.noise());  
16     }  
17 }  
18  
19
```

DogオブジェクトをCat型にキャストしている

実行

```
Exception in thread "main" java.lang.ClassCastException: Dog cannot be cast to Cat  
at Test5.main(Animal.java:14)
```