

# 接口

Java Platform Standard Edition

Java教学部

# 课程目标

## CONTENTS

ITEMS **1** 什么是接口

ITEMS **2** 接口与类的异同

ITEMS **3** 接口的应用

ITEMS **4** 接口的规范

ITEMS **5** 常量接口

ITEMS **6** 接口回调

# 接口的语法

- 接口相当于特殊的抽象类，定义方式、组成部分与抽象类类似。

使用 interface 关键字定义接口

没有构造方法，不能创建对象

```
interface MyInterface{  
    public static final String FIELD = "value";  
    public abstract void method();  
}
```

只能定义：公开静态常量、公开抽象方法

- 相同：

- 可编译成字节码文件。
- 不能创建对象。
- 可以作为引用类型。
- 具备Object类中所定义的方法。

- 不同：

- 所有属性都是公开静态常量，隐式使用public static final修饰。
- 所有方法都是公开抽象方法，隐式使用public abstract修饰。
- 没有构造方法、动态代码块、静态代码块。

# 什么是接口

- **微观概念：接口是一种能力和约定。**
  - 接口的定义：代表了某种能力。
  - 方法的定义：能力的具体要求。
- **经验：Java为单继承，当父类的方法种类无法满足子类需求时，可实现接口扩充子类能力。**
- **接口支持多实现，可为类扩充多种能力。**

- 任何类在实现接口时，必须实现接口中所有的抽象方法，否则此类为抽象类。
- 实现接口中的抽象方法时，访问修饰符必须是public。

- 同父类一样，接口也可声明为引用，并指向实现类对象。
- 注意：
  - 仅可调用接口中所声明的方法，不可调用实现类中独有的方法。
  - 可强转回实现类本身类型，进行独有方法调用。

# 接口的多态

```
public class TestPolymorphic {  
    public static void main(String[] args) {  
        Dog myDog = new Dog();  
        Animal a = myDog;  
        Runnable r = myDog;  
        Swimmable s = myDog;  
    }  
}  
  
interface Runnable{  
    public abstract void run();  
}  
  
interface Swimmable{  
    public abstract void swim();  
}  
  
abstract class Animal{  
    public void eat(){} //父类方法  
    public void sleep(){} //父类方法  
}  
  
class Dog extends Animal implements Runnable , Swimmable{  
    public void run(){} //接口方法  
    public void swim(){} //接口方法  
    public void shout(){} //独有方法  
}
```

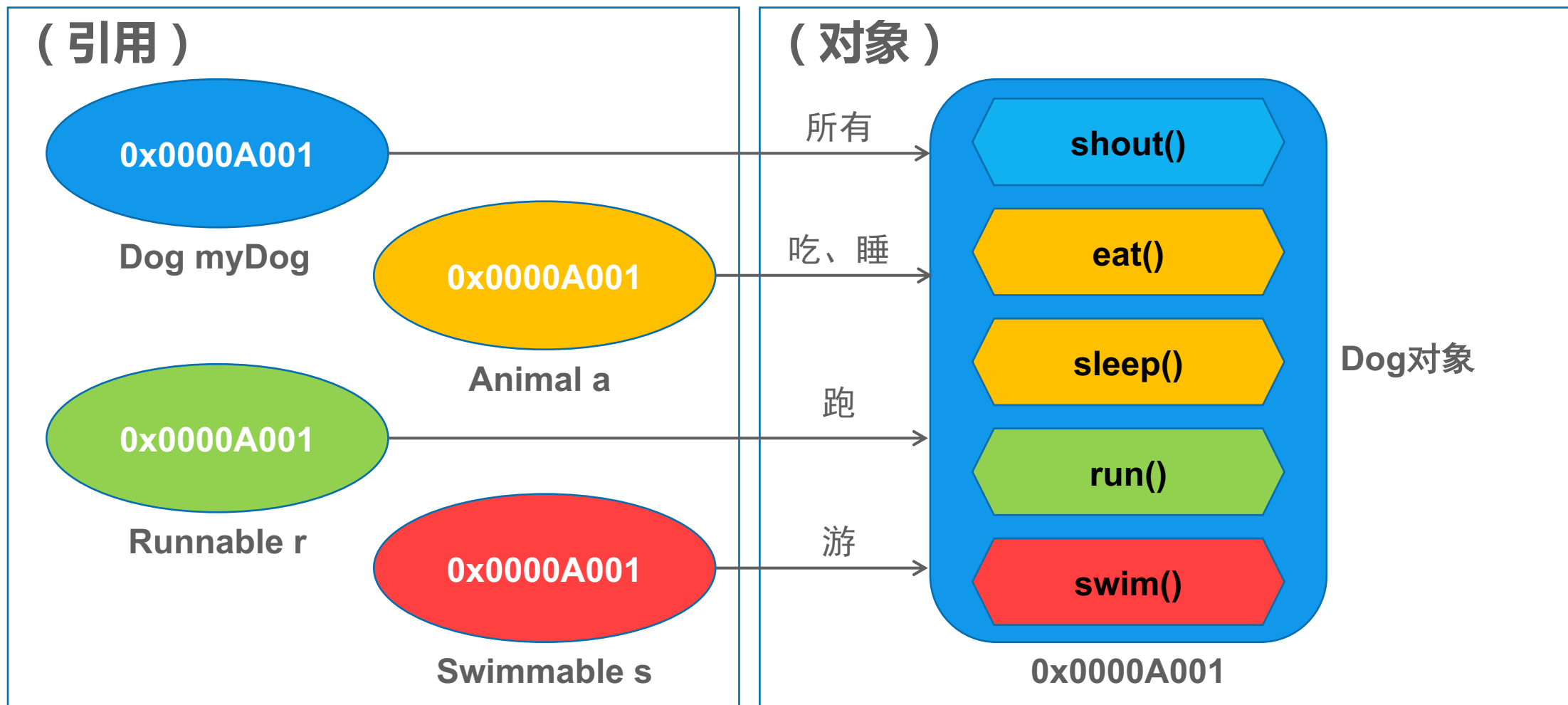
多种不同类型的引用指向同一个对象时，表示看待对象的视角不同。

- Dog myDog：将狗当狗看
- Animal a：将狗当动物看
- Runnable r：将狗当会跑的东西看
- Swimmable s：将狗当会游的东西看

不同引用所能看到的对象范围不同，只能调用自身类型中所声明的部分。



# 接口的多态



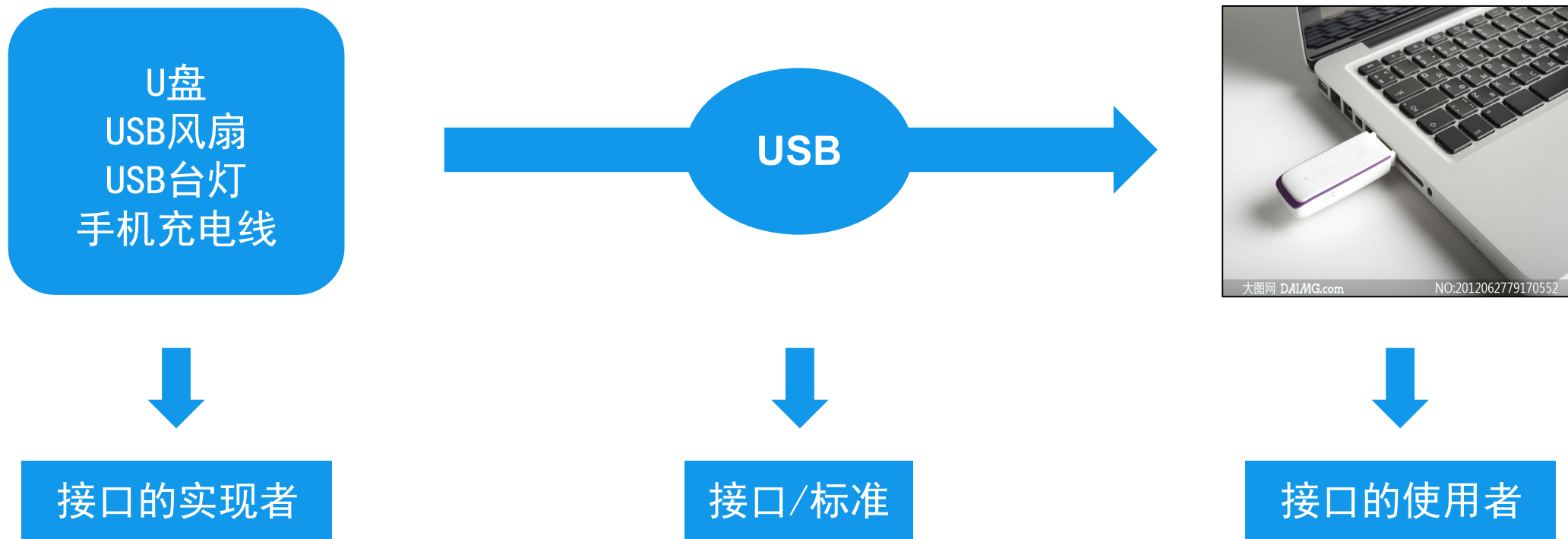
不同引用类型，仅可调用自身类型中所声明的方法。

- 类与类：
  - 单继承
  - extends 父类名称
- 类与接口：
  - 多实现
  - implements 接口名称1, 接口名称2, 接口名称n
- 接口与接口：
  - 多继承
  - extends 父接口1, 父接口2, 父接口n

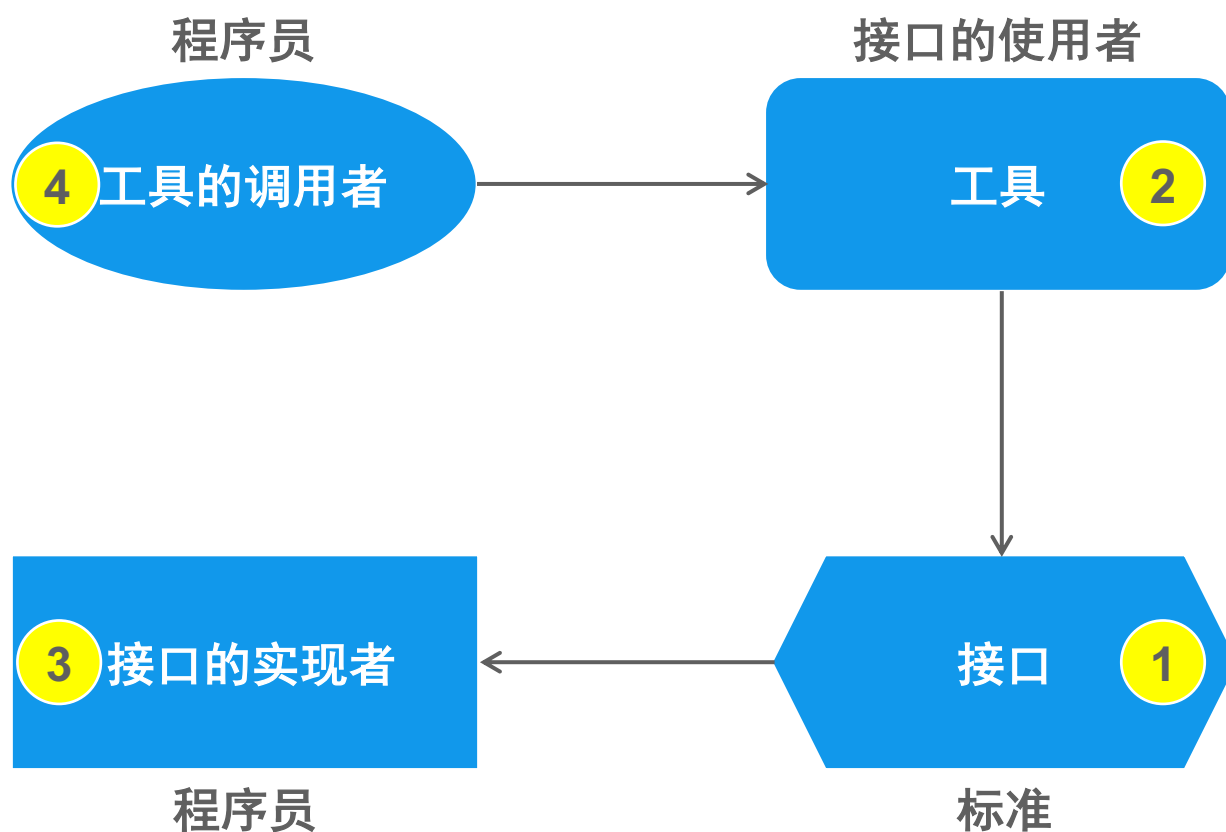
- 将多个常用于表示状态或固定值的变量，以静态常量的形式定义在接口中统一管理，提高代码可读性。

# 什么是接口

- 宏观概念：接口是一种标准。



接口回调：先有接口的使用者，后有接口的实现者



# 接口的好处

---

- 程序的耦合度降低。
- 更自然的使用多态。
- 设计与实现完全分离。
- 更容易搭建程序框架。
- 更容易更换具体实现。

- **什么是接口：**
  - 微观：接口是一种能力和约定。
  - 宏观：接口是一种标准。
- **接口与类的异同：**
  - 没有构造方法，仅可定义公开静态常量与公开抽象方法。
- **接口中的应用**
  - Java为单继承，当父类的方法种类无法满足子类需求时，可实现接口扩充子类能力。
- **接口的规范**
  - 任何类在实现接口时，必须实现接口中所有的抽象方法，否则此类为抽象类。
  - 实现接口中的抽象方法时，访问修饰符必须是public。
- **什么是常量接口**
  - 将多个常用于表示状态或固定值的变量，以静态常量的形式定义在接口中统一管理。
- **什么是接口回调**
  - 先有接口的使用者，后有接口的实现者。