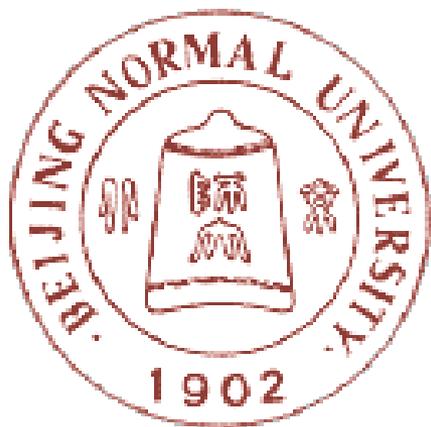


# 04面向对象程序设计思想

## (3) 接口与内部类 & Eclipse



宋杰 硕士研究生  
[songjiesdnu@163.com](mailto:songjiesdnu@163.com)

北京师范大学教育技术学院  
现代教育技术研究所



# 内容提要

- **5.1 接口**
- **5.2 内部类**
- **5.3 Eclipse**
  - 5.3.1 简介
  - 5.3.2 工程的建立、基本代码编写
  - 5.3.3 包和接口创建
  - 5.3.4 调试、代码自动完成
  - 5.3.5 Javadoc自动生成



# 从抽象类说起

- 一个问题：Dog和Cat对象是很合理的，但是Animal对象呢？

- 长什么样子？
- 什么颜色？
- 大小？
- 几条腿？



换句话说，  
Animal类不能被  
“new” 出来

- 我们一定要要有Animal这个类来产生继承，但是要限制只有他的子类才能被初始化

# 抽象类

```
abstract class Animal{  
    String name;  
    public abstract void roam();  
    public String getName(){ return name;}  
}
```

抽象方法!

```
class Dog extends Animal{  
    ...  
    public void roam(){  
        System.out.println("Wang~ wang~ wang~...");  
    }  
}
```

抽象方法必须  
重写!

```
...  
Animal a;  
a = new Dog();  
a = new Animal();
```

抽象类不能被实例化!



# 自己创建的Dog专用list

```
public class MyDogList {  
    private Dog[] dogs=new Dog[5];  
    private int nextIndex=0;  
    public void add(Dog d){  
        if (nextIndex<dogs.length){  
            dogs[nextIndex]=d;  
            System.out.println("Dog added at "+nextIndex);  
            nextIndex++;  
        }  
    }  
}
```

MyDogList

Dog[] dogs  
int nextIndex

add(Dog d)



# 不行！也要写出Cat用的

- 方案一：另外创建一个单独的MyCatList类来处理Cat对象，不好
- 方案二：创建一个单独的DogAndCatList类，用addCat(Cat c)和addDog(Dog d)来同时处理两个不同的数组实例，这也不好
- 方案三：编写一个不同的AnimalList类让他处理Animal所有的子类，试试看。。



# 试着实现一下方案3看看

```
public class MyAnimalList {  
    private Animal[] animals=new Animal[5];  
    private int nextIndex=0;  
    public void add(Animal a){  
        if (nextIndex<animals.length){  
            animals[nextIndex]=a;  
            System.out.println("Animal added at "+nextIndex);  
            nextIndex++;  
        }  
    }  
}
```

```
public class AnimalTestDrive{  
    public static void main (String[] args){  
        MyAnimalList list = new MyAnimalList();  
        Dog d= new Dog();  
        Cat c= new Cat();  
        list.add(d);  
        list.add(c);  
    }  
}
```

小测试：抽象类  
不是不能说new  
吗??



MyAnimalList

Animal[] animals  
int nextIndex

add(Animal a)

MyDogList

Dog[] dogs  
int nextIndex

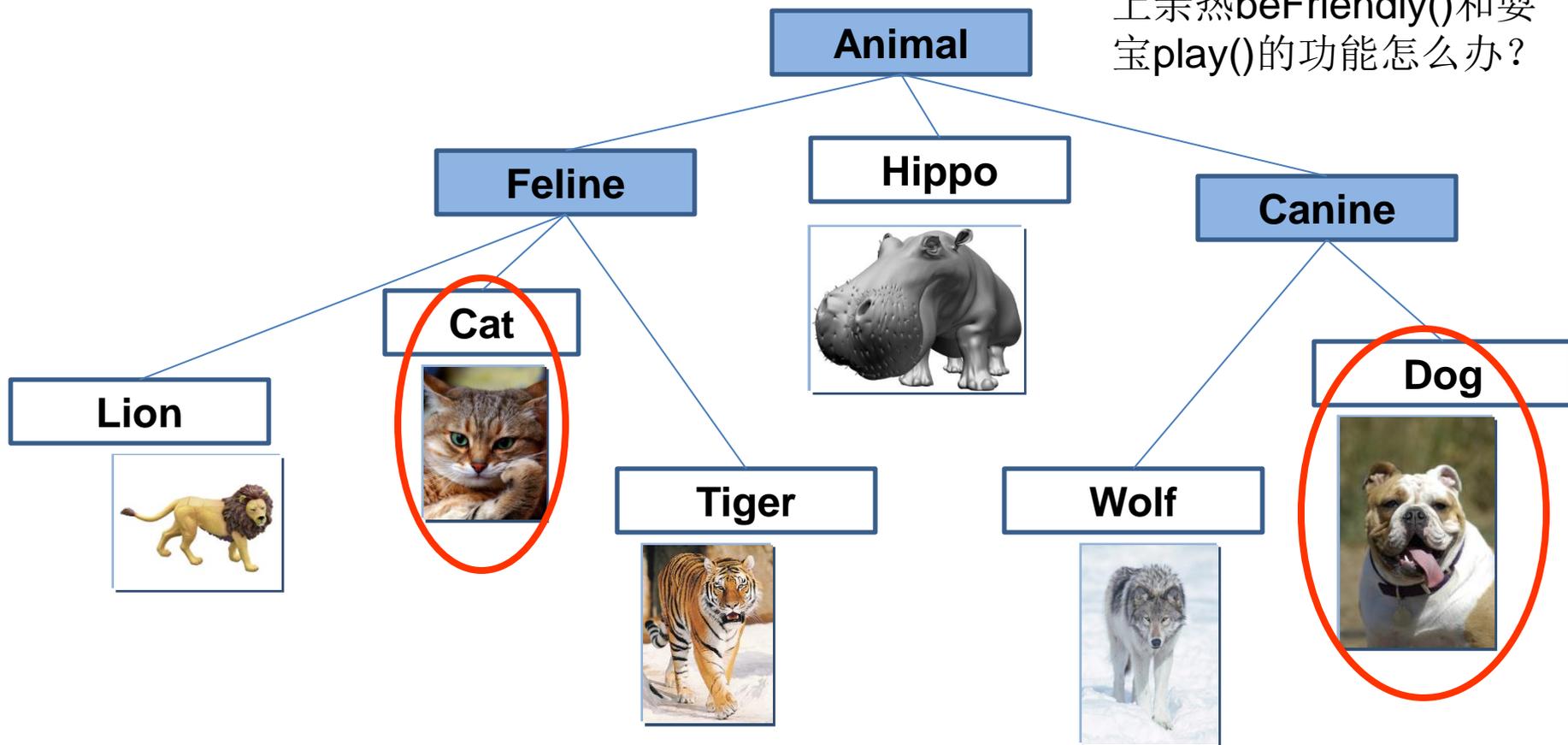
add(Dog d)

大家理解了  
多态的意义  
了吗??



# 问题来了

猫和狗是宠物，若要加上亲热beFriendly()和耍宝play()的功能怎么办？



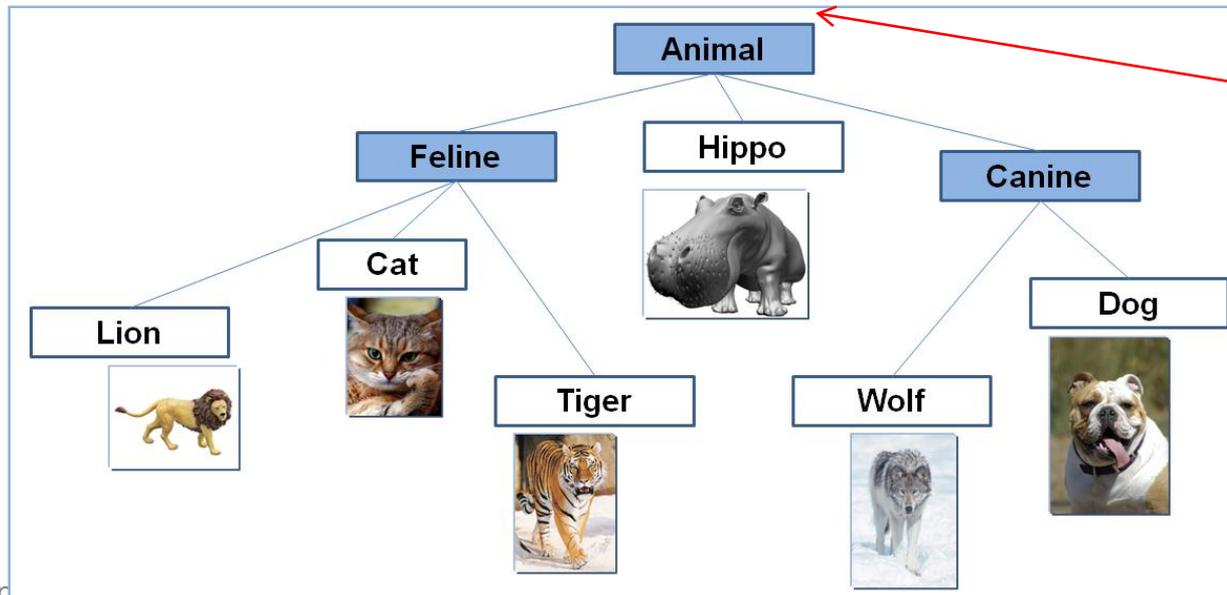
- 怎样让Animal可以选择性的（如猫、狗）带有宠物pet的行为又不会强迫让狮子老虎都表现成宠物？？



# 方法一

- 把宠物方法加进Animal类中

- 优点：所有的动物马上就可以继承宠物的行为，不需要改变所有子类的程序代码。
- 缺点：河马也成了宠物！你见过有人牵着一头河马去公园散步吗？

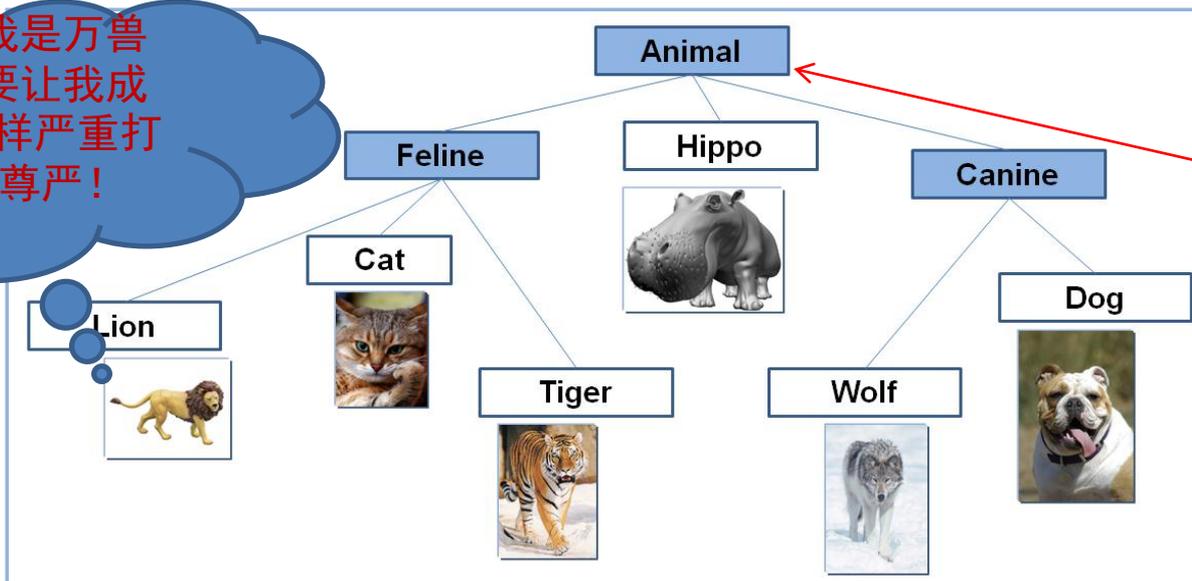


把方法加到这里

# 方法二

- 把宠物方法加进Animal类中，而且把宠物的方法设计成为抽象的，强迫每个动物子类覆盖他们
  - 优点：非宠物类的动物在重写这些方法时，做出合理的动作，或者什么也不做。
  - 缺点：所有具体的动物都得实现宠物的行为，不利于重复利用

狮子：哼，我是万兽之王，干嘛要让我成为宠物。--这样严重打击了狮子滴尊严！



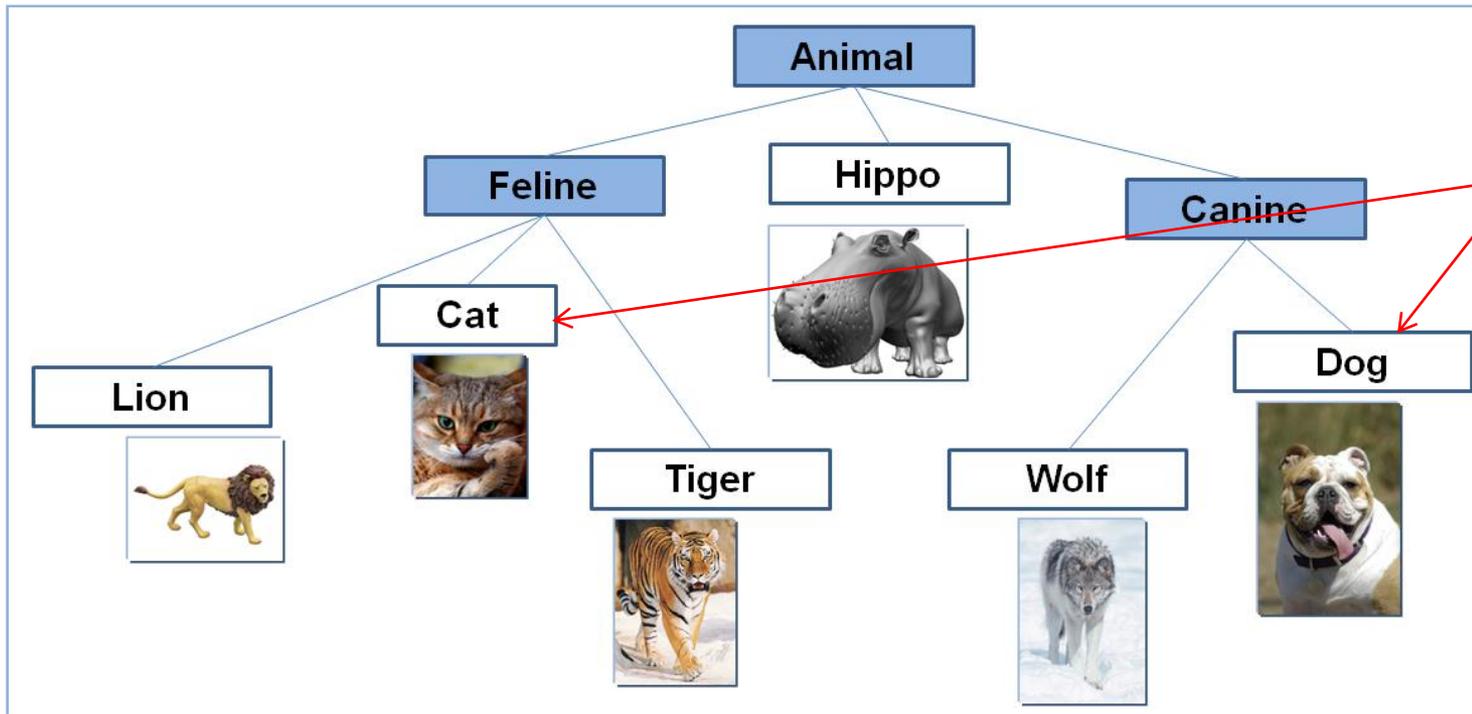
把抽象方法声明在这里，但并不实现



# 方法三

- 把宠物方法加到需要的地方

- 优点：不必担心牵着一头河马散步——只有宠物才有宠物的行为。
- 缺点：多态无法起作用了！

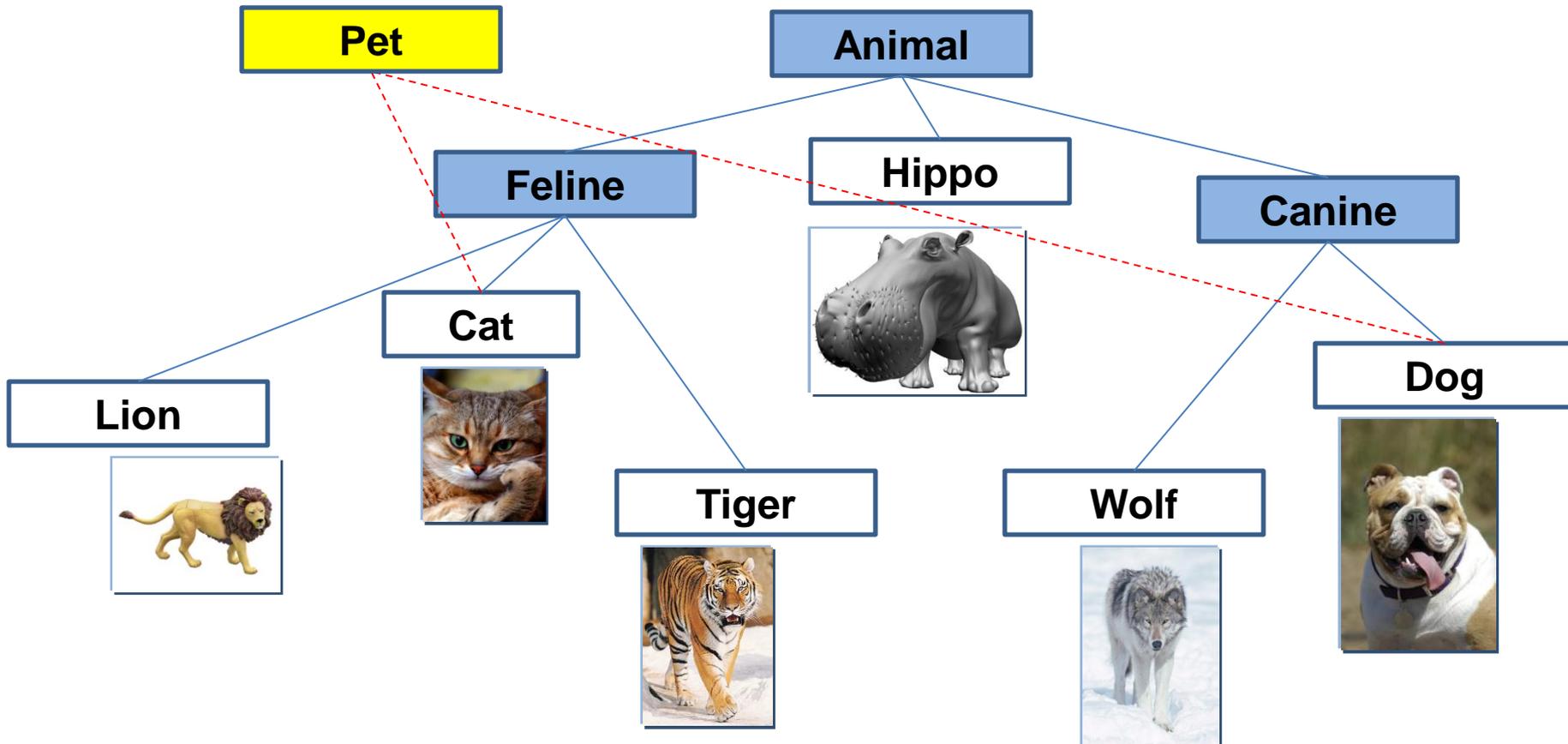


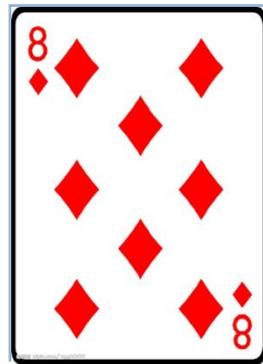
把宠物方法写在需要的类里



# So...how?

- 看起来我们需要两个上层父类。。。。

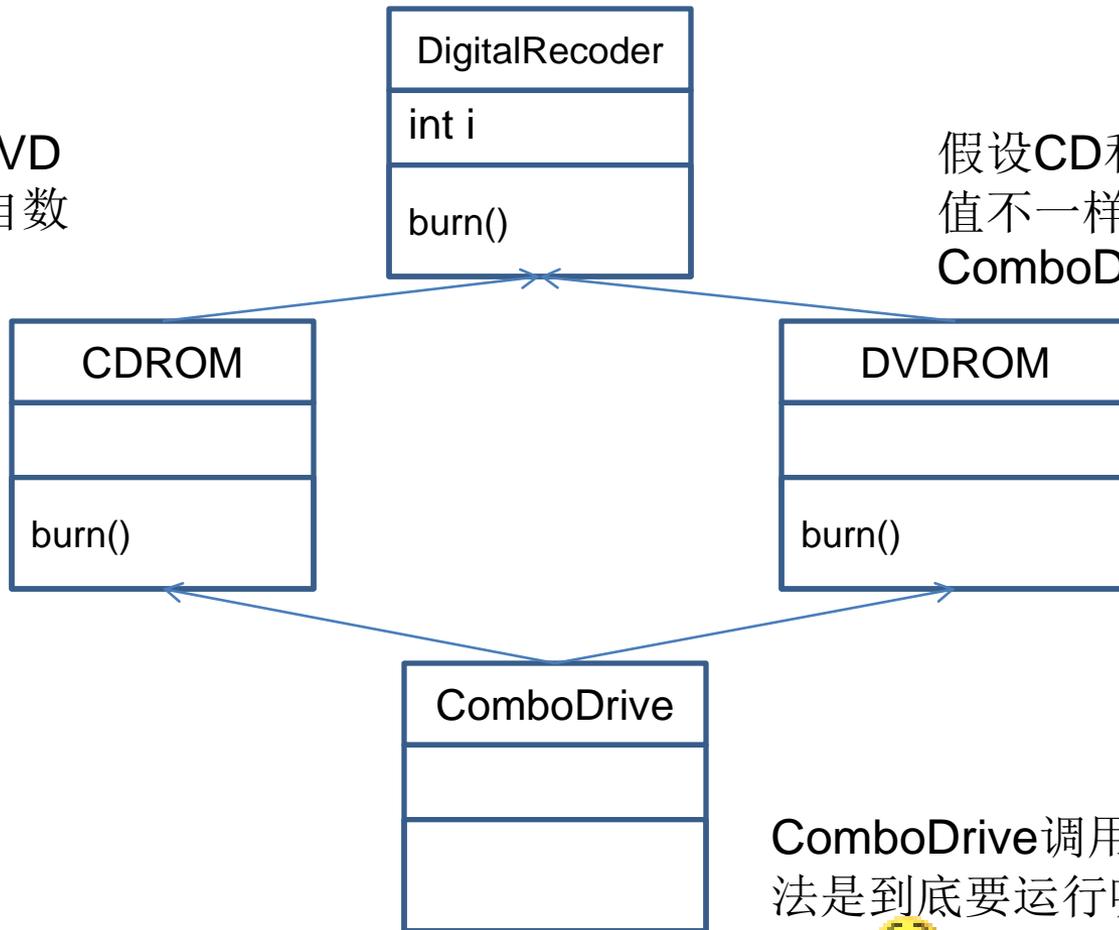




# But.....

- “多重继承”会有“致命方块”问题

CD Burner和DVD Burner都继承自数字记录器



假设CD和DVD都用到i且值不一样，那么ComboDriver怎么办？



ComboDrive调用burn()方法是到底要运行哪个版本呢？

- **允许致命方块的程序语言会产生很糟糕的复杂性问题。C++...**
- **So how...?**



# 接口

- Java的解决方案是接口interface!

- 接口的定义

```
public interface Pet{...}
```

- 接口的实现

```
public class Dog extends Canine  
implements Pet{...}
```



# 设计与实现Pet接口

接口的方法一定是抽象的，必须以分号结尾。记住他们没有内容！！

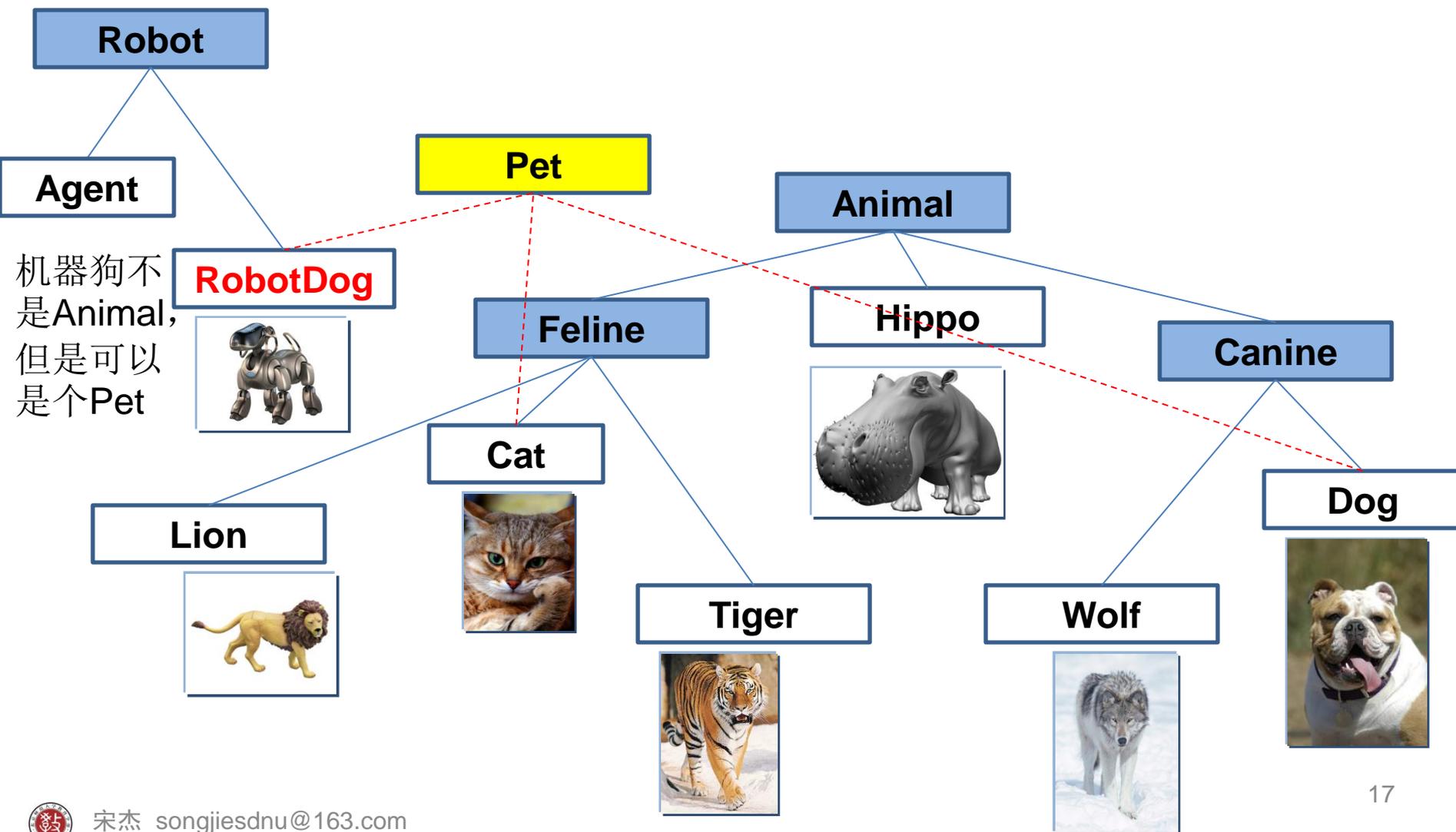
```
public interface Pet{  
    public abstract void beFriendly();  
    public abstract void play();  
}
```

```
public class Dog extends Canine implements Pet{  
    public void beFriendly(){...};  
    public void play(){...};  
}
```

必须实现Pet的方法



# 不同继承树的类也可以实现相同的接口



# 类可以实现多个接口

```
public class Dog extends Animal implements Pet,  
    Food, Swim {...}
```

- **extend只能有一个，implement可以有多个**
- **类来自单亲家庭(superclass)，但可以扮演多重角色(implement)**

# 内部类Inner class

- 内部类方法可以访问该类所在作用域中的数据，包括私有数据
- 内部类可以对同一个包中的其他类隐藏起来
- 内部类的class文件会有一个\$符号（上机时观察）

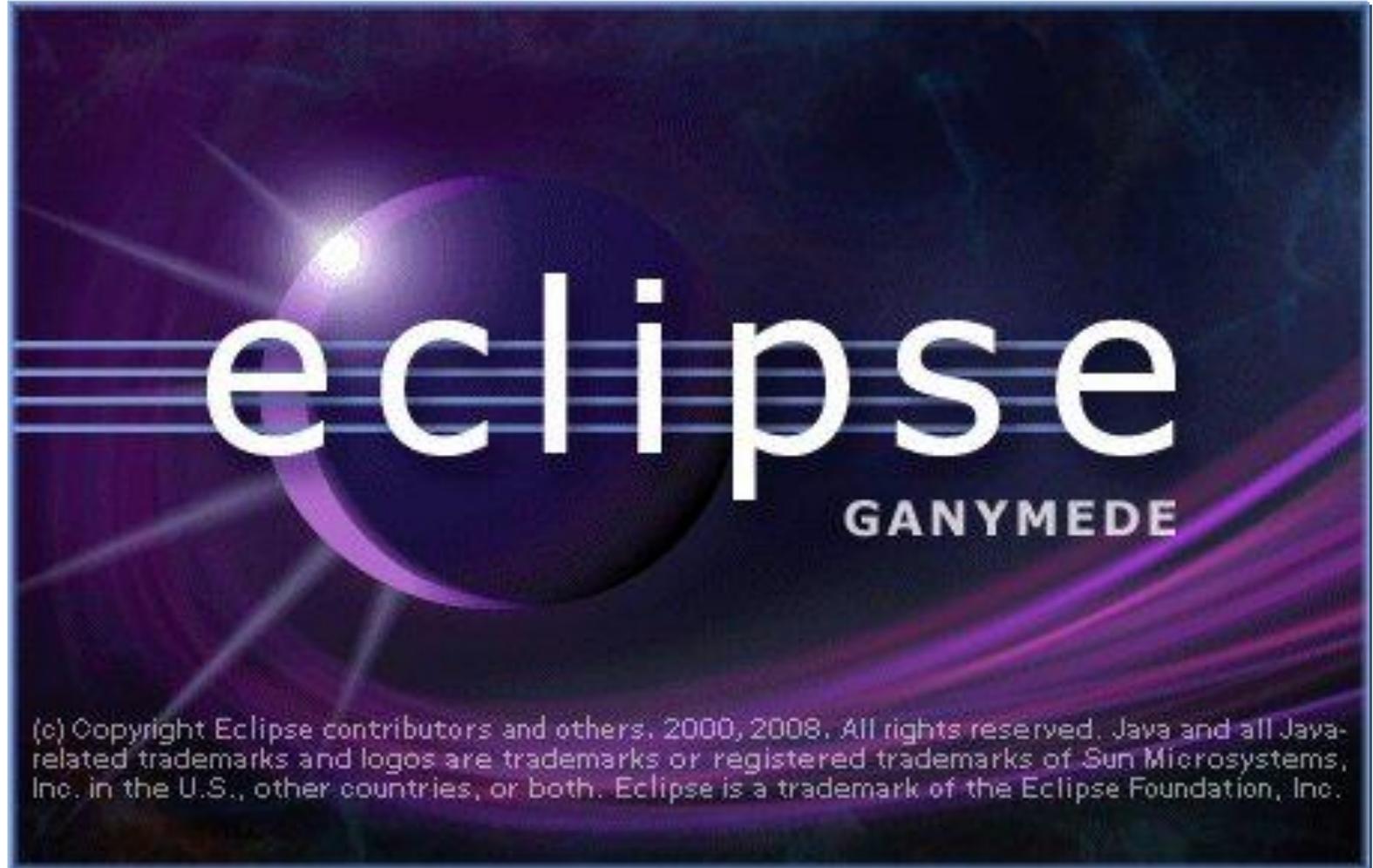


# 小结

- 面向对象的三个特点
  - 封装
  - 继承
  - 多态



# Eclipse



# JDK下载和环境变量设置

- JDK下载

<http://lcell.bnu.edu.cn/do/lcpage?action=view&kold=6735>

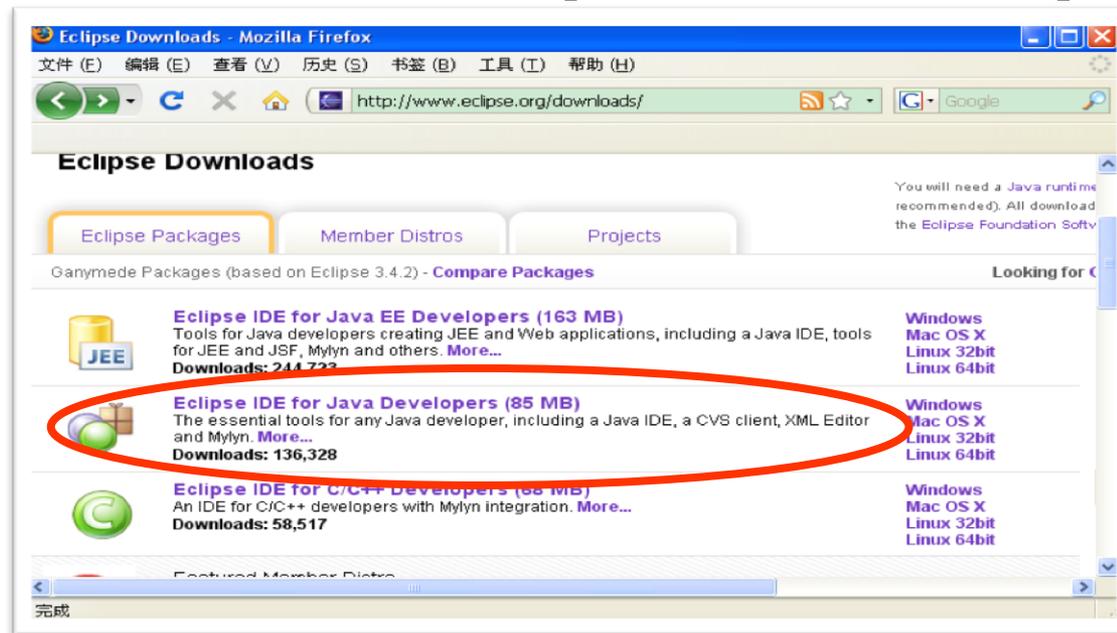
- JDK环境变量配置

<http://lcell.bnu.edu.cn/do/lcpage?action=view&kold=2936>



# Eclipse 安装

- Download it from <http://www.eclipse.org>



- 或者 <http://lcell.bnu.edu.cn/do/lcpage?action=view&kold=6735>
- 绿色软件，解压即可

# Eclipse工程的建立



# Eclipse中的包和工程高级配置

- 参考

[http://lcell.bnu.edu.cn/lc/lc\\_index.jsp?id=1652&courseId=113336](http://lcell.bnu.edu.cn/lc/lc_index.jsp?id=1652&courseId=113336)



# Eclipse调试、自动生成



## • Bug的典故

- 格蕾丝·霍波Grace Hopper(计算机语言之母)在软件设计领域的第一项重大“发明”，竟是创造出一个著名的计算机术语——“bug”。1946年，她在发生故障的Mark II计算机的继电器触点里，找到了一只被夹扁的小飞蛾，正是这只小虫子“卡”住了机器的运行。霍波顺手将飞蛾夹在工作笔记里，并诙谐地把程序故障称为“bug”。bug的意思是“臭虫”，而这一奇怪的称呼，后来演变成计算机行业的专业术语。虽然现代电脑再也不可能夹扁任何飞蛾，大家还是习惯地把排除程序故障叫做Debug（除虫）。

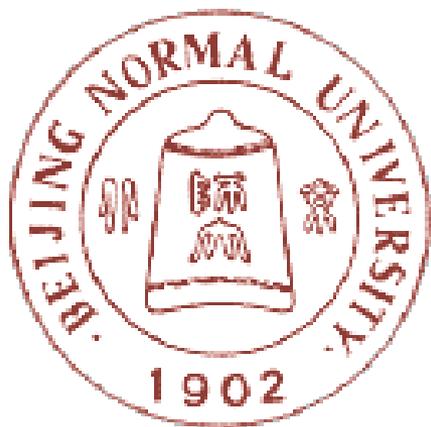
## • 使用eclipse简化工作

- 调试
- 自动生成set和get方法以及构造函数
- 自动生成规范Javadoc



# 04面向对象程序设计思想

## (3) 接口与内部类 & Eclipse



宋杰 硕士研究生  
[songjiesdnu@163.com](mailto:songjiesdnu@163.com)

北京师范大学教育技术学院  
现代教育技术研究所

