1. 抽象类的定义

抽象，即没有具体的实现，仅仅是抽象的声明。没有方法主体的方法称为抽象方法，包含抽象方法的类称为抽象类。

抽象类只是将不同对象的共性提取出来，放到一个类中，仅仅声明这些抽象的方法，而没有具体实现这些方法。注意：抽象类中既有抽象方法，也有普通的成员方法。

1. 抽象类的定义格式
2. 抽象方法

使用 abstract 关键字修饰方法，该方法就成了抽象方法，抽象方法只包含一个方法名，而没有方法体。

定义格式：

代码举例：

public abstract void run()；

1. 抽象类

用abstract关键字修饰

格式：

abstract class 类名字 {

}

1. 抽象方法和抽象类的使用

抽象类是不能直接创建对象的，只能通过子类继承抽象父类，然后创建子类对象。抽象父类中定义的抽象方法都需要在子类中重写。重写方法时需要去掉abstract关键字。

若子类没有实现抽象父类的所有抽象方法，那么该子类还是一个抽象类，不能创建对象，只有实现了父类的所有抽象方法，该子类才能创建对象。

代码实现：

public abstract class Animal {

    public abstract void run()；

}

public class Cat extends Animal {

    public void run (){

       System.out.println("小猫在墙头走~~~")；

    }

}

public class CatTest {

   public static void main(String[] args) {

        // 创建子类对象

        Cat c = new Cat();

        // 调用run方法

        c.run();

   }

}

1. 注意事项
2. 抽象类中，可以有构造方法，用于抽象类的子类创建对象，初始化父类成员是使用的。
3. 抽象类中，不一定包含抽象方法，但是有抽象方法的类一定是抽象类。没有抽象方法的抽象类，目的就是不让调用者创建该类对象。