“十三五”职业教育国家规划教材

高等职业教育计算机类课程
MOOC+SPOC 系列教材

Java 程序设计案例教程

主　编 / 张红
副主编 / 胡坚

智慧职教
MOOC+SPOC课程

扫描二维码
了解本书的
配套资源

- 微课视频
- 课程标准
- 授课计划
- 电子教案
- 授课用PPT

+

- 案例源码：案例及实训代码与素材
- 习题答案：经典测试题目及详细解答

高等教育出版社

3）为 DateUtils 增加私有的 0 参构造函数，防止被实例化。

4）创建归还日期计算函数。

【实现代码】 由于案例实现的代码较多，请通过下载查看。

案例：图书借阅系统 …… （Book.rar 案例代码下载）

源代码

5.4 抽象类

5.4.1 抽象类的运用场合

微课 5-10
抽象类的运用场合

在大多数情况下，Java 的类是可以被其他类继承的，也可以被直接实例化使用，但有一种类是专门用类来作父类的，自身也不能实例化，这样的类称为抽象类。抽象类有点类似于“模板”的作用，目的是根据其格式来创建和修改新的类，对象不能由抽象类直接创建，只可以通过抽象类派生出新的子类，再由其子类来创建对象，当一个类被声明为抽象类时，要在这个类前面加上修饰符 abstract，在抽象类中的成员方法可以包括一般方法和抽象方法。

包含了抽象方法的类一律称为抽象类，一个抽象类可以存在一个或多个抽象方法。抽象方法是只定义了方法的声明但不包含具体实现的方法。

抽象类可以帮助用户管理自己的代码。例如，当用户定义一个宠物抽象类时，则希望所有的宠物都具有吃、喝、叫功能，这样就可以在这个宠物类中添加 3 个抽象方法，在要继承这个宠物类时，会提示是否要实现这 3 个抽象，它可以防止用户漏掉一些功能，便于开发。

5.4.2 定义抽象类

微课 5-11
定义抽象类

如前所述，类的抽象方法是指没有具体实现的方法，抽象方法用关键字 abstract 修饰：

```
abstract void fn();
```

正如上面所示，fn()方法不存在方法体。任何含有一个或多个抽象方法的类都必须声明为抽象类；否则，编译器会报告一条出错消息。声明一个抽象类，只需在类声明开始时在关键字 class 前使用关键字 abstract。抽象类定义的具体形式为：

```
abstract  class 类名称
{
    成员变量;
方法(){……};        //一般方法
abstract 方法();  //抽象方法
}
```

笔记

抽象类没有对象。也就是说，一个抽象类不能通过 new 操作符直接实例化。这样的对象是无用的，因为抽象类是不完全定义的，而且不能定义抽象构造方法或抽象静态方法。所有抽象类的子类都必须实现父类中的所有抽象方法或者是它自己也声明成 abstract 类。

如下定义 Person 抽象类：

```
abstract class Person {
public static final String country="中国";
private String name;
public  void setName(String name) {
    this.name=name;
}
public  String getName() {
    return name;
}
public abstract void tell(); //定义抽象方法，没有方法主体
}
```

不能用 Person 类来实例化对象，此时它可以定义一个 Person 类的句柄，但不可以定义一个 Person 对象。例如：

```
Person p; // 正确，可以定义一个抽象类句柄
Person  x = new Person(); //错，不允许定义一个抽象类对象
```

5.4.3 继承抽象类

微课 5-12
继承抽象类

抽象类的实现是指具体化抽象类中的所有抽象方法，这一任务通常交给抽象类的子类去完成，关于抽象类的典型应用如下所示。

```
/** AbstractTest.java: 抽象类测试*/
abstract class Person {
 public static final String country="中国";
 private String name;
 public  void setName(String name) {
     this.name=name;
 }
 public  String getName() {   return name; }
 public abstract void tell(); //定义抽象方法，没有方法主体
}
class Student extends Person {
 public void tell() {//重写抽象类的所有抽象方法
 System.out.println("我叫"+getName()+", 是一名"+country+"人！");
 }
}
public class AbstractTest {
 public static void main(String[] args) {
```

```
        Student stu=new Student();    //创建 Student 类的实例
        stu.setName("小强");
        stu.tell();  //调用子类 Student 的 tell()
    }
}
```

程序运行结果如下图 5-8 所示。

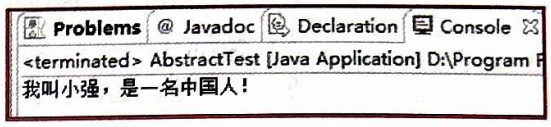

图 5-8 继承抽象类

子类对抽象类中的抽象方法继承时，必须实现抽象类中的所有抽象方法，否则该子类仍必须被定义为抽象类。若子类没有完全实现抽象类的抽象方法且未被定义为抽象类，则在编译时会报错。

5.5 案例：工程师房屋建造

当我们学习了这部分知识，大家实际上已经掌握了面向对象中的抽象类与抽象方法，以及 static 关键字和类的继承等问题。

微课 5-13
案例：工程师房屋建造

5.5.1 案例要求

【案例目标】 以建筑师建造房屋为业务背景，体验“抽象类”与子类的应用关系和特性。

【相关解释】 本实验以建筑师建造房屋为业务背景。房屋建造的施工规范和流程：先建基础设施（如地基），再建主体设施。基础设施必须在计划完工时间前完成施工，随后开始主体设施建设。若实际完工时间超过计划完工时间，则不能进行主体设施建设。

【案例效果】 本案例程序运行的结果如图 5-9 所示。

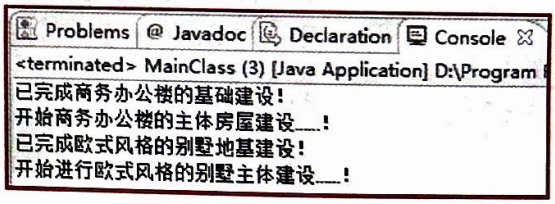

图 5-9 工程师房屋建造案例程序运行结果

【具体要求】 本案例的实现过程应满足以下要求。

1. 创建工程并配置环境

1）限制 1：工程取名 SE_JAVA_EXP_E035。

2）限制 2：创建包，取名 cn.campsg.java.experiment。

3）限制 3：创建建造者抽象类，类名为 Builder。

2. 为 Builder 类创建抽象方法（二）

1）限制 1：方法名为 buildBody（抽象方法）。

2）作用：用于描述房屋的主体设施建设，返回建造进度。

3）限制 2：buildBody 方法设定为 0 参公共方法，无返回。

3. 为 Builder 类创建实体方法（三）

1）限制 1：方法名为 construct（实体方法）。