Java语言基础

【Day10】

抽象类
abstract class

抽象类的基本概念

- 用abstract关键字修饰的类称为抽象类。抽象类不能实例化，抽象类的意义在于“被继承”。抽象类为其子类“抽象”出了公共的部分，通常也定义了子类所必须具体实现的抽象方法。

```
public abstract class Shape{
    private int x;
    private int y;
    public abstract boolean contains(int x, int y);
}
```

一个类如果定义有抽象方法则必须以abstract关键字声明为抽象类。

用abstract修饰的方法，称之为抽象方法，抽象方法仅仅有方法的定义，而没有方法的实现。

继承抽象类

- 一个类继承抽象类必须实现其抽象方法（除非该类也声明为抽象类）。

```
class Circle extends Shape{
    int r;
    public boolean contains
        (int x, int y) {
        …… ……
    }
}
```

```
class Rect extends Shape {
    int width;
    int height;
    public boolean contains
        (int x, int y) {
        …… ……
    }
}
```

继承抽象类，必须实现抽象方法（不同子类可能有不同实现）

抽象类的应用

- 模板设计模式
- 银行有 定期账户和活期账户。继承自 账户类。账户类中：

```
public class Account{
    private double money;
    public double getLixi(){  获取利率的方法，问题来了：
        定期和活期账户的利率不一样，方法没法写；
        但这个方法必须存在，只能定义成抽象方法，交给子类实现
    }
}
```

接口
interface

接口的基本概念

- 接口可以看成是特殊的抽象类。即只包含有抽象方法的抽象类。

```
interface Runner {
    public static final int DEF_SPEED = 100;

    public void run();
}
```

通过interface关键字定义接口

接口中不可以定义成员变量，但可以定义常量。

接口中只可以定义没有实现的方法（可以省略public abstract）。

接口的实现

- 一个类可以通过implements关键字实现接口。
- 一个类实现了某个接口后必须实现该接口中定义的所有方法。

```
class AmericanCurl implements Runner , … {
    public void run() {
        System.out.println("run...");
    }
}
```

与继承不同，一个类可以实现多个接口，实现的接口直接用逗号分隔。当然，该类需要实现这些接口中定义的所有方法。

```
Runner runner = new AmericanCurl();
```

接口可以作为一种类型声明变量，一个接口类型的变量可以引用实现了该接口的类的对象；通过该变量可以调用该接口中定义的方法（具体的实现类提供了方法的实现）。

接口的继承

- 接口间可以存在继承关系，一个接口可以通过extends关键字继承另外一个接口。子接口继承了父接口中定义的所有方法。

```
interface Runner {
    public void run();
}

interface Hunter
        extends Runner {
    public void hunt();
}
```

```
class AmericanCurl implements Hunter {
    public void run() {… … …}
    public void hunt() {… … …}
}
```

AmericanCurl必须实现Hunter 接口中的hunt方法以及其父接口Runner中的run方法。

匿名内部类

Inner class anonymous

匿名内部类

- 如果在一段程序中需要创建一个类的对象（通常这个类需要实现某个接口或者继承某个类），而且对象创建后这个类的价值也就不存在了，这个类可以不必命名，称之为匿名内部类。

用匿名类所实现的接口或所继承的父类类型声明的引用

```
SuperType obj = new SuperType (…) {
      …… …… ……
};
```

匿名中定义的成员变量或方法

回调模式的概念

- 回调模式是指——如果一个方法的参数是接口类型，则在调用该方法时，需要创建并传递一个实现此接口类型的对象；而该方法在运行时会调用到参数对象中所实现的方法（接口中定义的）。

回调模式的实现

```
interface Action {
    public void doSth();
}
```

```
public static void repeat(int n, Action action) {
    for (int i = 0; i < n; i++) {
        action.doSth();
    }
}
```

repeat方法需要一个Action接口类型的参数，其逻辑为将此参数的doSth方法重复执行n次

```
public static void main(String[] args) {
    repeat(5, new Action() {
        public void doSth() {
            System.out.println("Hello, World");
        }
    });
}
```

通过内部匿名类传递参数，此处的语义可解释为：通过接口回调传递了一个方法给repeat，让repeat将其执行5次。

总结与答疑

技术资料

IT兄弟连

ITXDL.CN

变态严管　让学习成为一种习惯