

## 什么是面向对象？

是一种将事物抽象表示的思想

对象：拥有自己的属性

类：将对象分类，相当于身份，定义类之间共有的属性

继承：一个类继承另一个类，例如学生类继承人类

多态：一个方法在不同对象的不同实现方法

封装：将数据赋予一个对象

## 静态方法和构造方法有什么区别？

构造方法：初始化对象的属性，在创建对象的时候会自动调用

分为空参构造方法和有参构造方法

空参构造方法不用写电脑也会默认有

例如

```
public class Dish {

    private String name;
    private double price;
    private String introduction;

    public Dish_0() {

    }
    //空参构造

    public Dish_0(String name, double price, String introduction) {
        this.name = name;
        this.price = price;
        this.introduction = introduction;
    }
    //有参构造

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
//因为都是私有化变量，所以还要加上set, get方法
```

静态方法：直接通过类名调用，有static标志，不属于某一个类的实例

## 在Java中修饰符有哪些？什么情况下使用？使用对象是什么？

### 访问修饰符

public 对象：类，方法，变量

希望能被其它类使用时用

private 对象：类，方法

将类内部隐藏起来，保护作用，要用get, set访问 protected 对象：方法、变量

能在不同包使用 没有 对象：类、方法、变量

只能在同一包内使用

### 其他

static 对象：方法、变量、内部类

表示属于这个类本身

abstract 对象：类，方法

为子类提供一个模板，强迫子类必须去重写

final：类、方法、变量

修饰类时，该类不能被继承；修饰方法时，该方法不能被重写；修饰变量时，该变量一旦被赋值就不能再改变（相当于常量）

还有synchronized volatile等

检查三角形

```
package yichang;

class NotTriangle extends Exception {
    public NotTriangle(String message) {
        super(message);
    }
}

class Triangle {
    private double side1;
    private double side2;
    private double side3;

    public Triangle(double side1, double side2, double side3) {
        this.side1 = side1;
        this.side2 = side2;
        this.side3 = side3;
    }

    public double area() throws NotTriangle {
        if (side1 <= 0 || side2 <= 0 || side3 <= 0 ||
```

```

        side1 + side2 <= side3 || side1 + side3 <= side2 || side2 + side3 <= side1) {
            throw new NotTriangle("这不是一个有效的三角形");
        }
        double s = (side1 + side2 + side3) / 2;
        return Math.sqrt(s * (s - side1) * (s - side2) * (s - side3));
    }
}

public class jiancha {
    public static void main(String[] args) {
        Triangle triangle1 = new Triangle(3.0, 4.0, 5.0);
        Triangle triangle2 = new Triangle(1.0, 1.0, 2.0);

        try {
            double area1 = triangle1.area();
            System.out.println("边长为 3.0, 4.0, 5.0 的三角形面积为: " + area1);
        } catch (NotTriangle e) {
            System.out.println(e.getMessage());
        }

        try {
            double area2 = triangle2.area();
            System.out.println("边长为 1.0, 1.0, 2.0 的三角形面积为: " + area2);
        } catch (NotTriangle e) {
            System.out.println(e.getMessage());
        }
    }
}

```

取款

```

package yichang;

class InsufficientBalanceException extends Exception {
    public InsufficientBalanceException(String accountNumber, double amount) {
        super("账户号码 " + accountNumber + " 余额不足, 无法提取 " + amount + " 元。");
    }
}

class BankAccount {
    private String accountNumber;
    private double balance;

    public BankAccount(String accountNumber, double balance) {
        this.accountNumber = accountNumber;
        this.balance = balance;
    }

    public String getAccountNumber() {
        return accountNumber;
    }

    public void setAccountNumber(String accountNumber) {
        this.accountNumber = accountNumber;
    }

    public double getBalance() {
        return balance;
    }

    public void setBalance(double balance) {
        this.balance = balance;
    }

    public void withdraw(double amount) throws InsufficientBalanceException {
        if (amount > balance) {
            throw new InsufficientBalanceException(accountNumber, amount);
        }
        balance -= amount;
        System.out.println("成功从账户 " + accountNumber + " 提取 " + amount + " 元, 当前余额: " + balance + " 元。");
    }
}

class jiancha{
    public static void main(String[] args) {
        BankAccount account = new BankAccount("123456", 1000);
        try {
            account.withdraw(500);
            account.withdraw(2000);
        } catch (InsufficientBalanceException e) {
            System.out.println(e.getMessage());
        }
    }
}

```

}