1

java不支持多继承主要是因为菱形继承问题，即： 若有B类和C类同时继承了A类的同一方法，如果D类同时继承B类和C类时，在调用该方法时就不知道是该调用B还是C，出现歧义，导致代码难以维护和理解。 所以JAVA的设计者只采取了单继承。

2

代码如下

```
public class text {
    abstract class Shape{
        public abstract double C();
        public abstract double S();
        public abstract String name();

        public void printInfo(){
            System.out.println("图形:"+name());
            System.out.println("周长:"+String.format("%.2f",C()));
            System.out.println("面积:"+String.format("%.2f",S()));
        }
    }
    class Circle extends Shape{
        private String name="圆形";
        @Override
        public String name(){
            return name;
        }
        private double r;
        public Circle(double r){
            if(r<=0){
                throw new IllegalArgumentException("半径必须大于0");
        }
            this.r=r;
        }
        @Override
        public double C(){
            return 2*Math.PI*r;
        }
        public double S(){
            return Math.PI*r*r;
        }
        public double getR(){
            return r;
        }
    }
    class Rectangle extends Shape{
        private String name="矩形";
        @Override
```

```
        public String name(){
            return name;
        }
        private double a;
        private double b;
        public Rectangle(double a,double b){
            if(a<=0||b<=0){
                throw new IllegalArgumentException("边长必须大于0");
            }
            this.a=a;
            this.b=b;
        }
        @Override
        public double C(){
            return 2*(a+b);
        }
        public double S(){
            return a*b;
        }
        public double getA(){
            return a;
        }
        public double getB(){
            return b;
        }
    }
    class Triangle extends Shape{
        private String name="三角形";
        @Override
        public String name(){
            return name;
        }
        private double a,b,c;
        public Triangle(double a,double b,double c){
            if(!isValidTriangle(a,b,c)){
                throw new IllegalArgumentException("不能构成三角形");
            }
            this.a=a;
            this.b=b;
            this.c=c;
        }
        private boolean isValidTriangle(double a,double b,double c){
            return a+b>c && a+c>b && b+c>a;
        }
        @Override
        public double C(){
            return a+b+c;
        }
        public double S(){
            double p=(a+b+c)/2;
            return Math.sqrt(p*(p-a)*(p-b)*(p-c));
        }
        public double getA(){
            return a;
```

```
        }
        public double getB(){
            return b;
        }
        public double getC(){
            return c;
        }
    }
    public static void main(String[] args) {
        text t=new text();
        Shape cir=t.new Circle(5);
        cir.printInfo();
        Shape rec=t.new Rectangle(4,5);
        rec.printInfo();
        Shape tri=t.new Triangle(3,4,5);
        tri.printInfo();
    }
}
```

```
PS D:\IT\VS\Cuse> javac text.java
PS D:\IT\VS\Cuse> java text.java
图形:圆形
周长:31.42
面积:78.54
图形:矩形
周长:18.00
面积:20.00
图形:三角形
周长:12.00
面积:6.00
PS D:\IT\VS\Cuse>
```

使用接口完成

```
public class text {
    abstract static class Shape{
        public abstract double C();
        public abstract double S();
        public abstract String name();

        public void printInfo(){
            System.out.println("图形:"+name());
            System.out.println("周长:"+String.format("%.2f",C()));
            System.out.println("面积:"+String.format("%.2f",S()));
        }
    }
    static class Circle extends Shape{
        private String name="圆形";
        @Override
        public String name(){
            return name;
```

```
        }
        private double r;
        public Circle(double r){
            if(r<=0){
                throw new IllegalArgumentException("半径必须大于0");
        }
            this.r=r;
        }
        @Override
        public double C(){
            return 2*Math.PI*r;
        }
        public double S(){
            return Math.PI*r*r;
        }
        public double getR(){
            return r;
        }
    }
    static class Rectangle extends Shape{
        private String name="矩形";
        @Override
        public String name(){
            return name;
        }
        private double a;
        private double b;
        public Rectangle(double a,double b){
            if(a<=0||b<=0){
                throw new IllegalArgumentException("边长必须大于0");
            }
            this.a=a;
            this.b=b;
        }
        @Override
        public double C(){
            return 2*(a+b);
        }
        public double S(){
            return a*b;
        }
        public double getA(){
            return a;
        }
        public double getB(){
            return b;
        }
    }
    static class Triangle extends Shape{
        private String name="三角形";
        @Override
        public String name(){
            return name;
        }
```

```
        private double a,b,c;
        public Triangle(double a,double b,double c){
            if(!isValidTriangle(a,b,c)){
                throw new IllegalArgumentException("不能构成三角形");
            }
            this.a=a;
            this.b=b;
            this.c=c;
        }
        private boolean isValidTriangle(double a,double b,double c){
            return a+b>c && a+c>b && b+c>a;
        }
        @Override
        public double C(){
            return a+b+c;
        }
        public double S(){
            double p=(a+b+c)/2;
            return Math.sqrt(p*(p-a)*(p-b)*(p-c));
        }
        public double getA(){
            return a;
        }
        public double getB(){
            return b;
        }
        public double getC(){
            return c;
        }
    }
    public static void main(String[] args) {
        Shape[] shapes= {
            new Circle(5),
            new Rectangle(4,5),
            new Triangle(3,4,5)
        };

        for(Shape shape:shapes){
            shape.printInfo();
        }
    }
}
```

```
PS D:\IT\VS\Cuse> javac text.java
PS D:\IT\VS\Cuse> java text.java
图形:圆形
周长:31.42
面积:78.54
图形:矩形
周长:18.00
面积:20.00
图形:三角形
周长:12.00
面积:6.00
PS D:\IT\VS\Cuse>
```

3

private：用于封装内部实现细节，提高安全性 default：用于包内协作，限制包外访问 protected：用于继承体系，允许子类访问父类功能 public：用于定义对外的API接口 代码补充

```
public class BankAccount {
    // TODO 修改属性的可见性
    String accountNumber;
    String accountHolder;
    double balance;
    String password; // 敏感信息，需要严格保护

   public BankAccount(String accountNumber, String accountHolder, double
initialBalance, String password) {
        this.accountNumber = accountNumber;
        this.accountHolder = accountHolder;
        this.balance = initialBalance;
        this.password = password;    //TODO
    }

    public void deposit(double amount) {
        if (validateAmount(amount)) {
            balance += amount;
            System.out.println("存款成功！存入金额：" + amount + "，当前余额：" +
balance);
        } else {
            System.out.println("存款失败！金额必须大于0");
        }    //TODO
    }

    public boolean withdraw(double amount, String inputPassword) {
        if (!validatePassword(inputPassword)) {
            System.out.println("取款失败！密码错误");
            return false;
            }
        if (!validateAmount(amount)) {
```

```
            System.out.println("取款失败！金额必须大于0");
            return false;
        }
        if (amount > balance) {
            System.out.println("取款失败！余额不足");
            return false;
        }

        balance -= amount;
        System.out.println("取款成功！取出金额：" + amount + "，当前余额：" +
balance);
        return true;    //TODO
    }

    public boolean transfer(BankAccount recipient, double amount, String
inputPassword) {
        if (!validatePassword(inputPassword)) {
            System.out.println("转账失败！密码错误");
            return false;
        }
        if (!validateAmount(amount)) {
            System.out.println("转账失败！金额必须大于0");
            return false;
        }
        if (amount > balance) {
            System.out.println("转账失败！余额不足");
            return false;
        }
        if (recipient == null) {
            System.out.println("转账失败！收款账户不存在");
            return false;
        }

        this.balance -= amount;
        recipient.balance += amount;

        System.out.println("转账成功！向账户" + recipient.accountNumber + "转账" +
amount + "元");
        System.out.println("当前余额：" + balance);
        return true;    //TODO
    }

    public double getBalance() {
        return balance;    //TODO
    }

    public String getAccountInfo() {
        return "账户号码：" + accountNumber + "，户主：" + accountHolder + "，余
额：" + balance;    //TODO
    }
    // 只需修改可见性
    private boolean validatePassword(String inputPassword) {
        return true;
    }
```

```
    // 只需修改可见性
    private boolean validateAmount(double amount) {
        return true;
    }
}
```