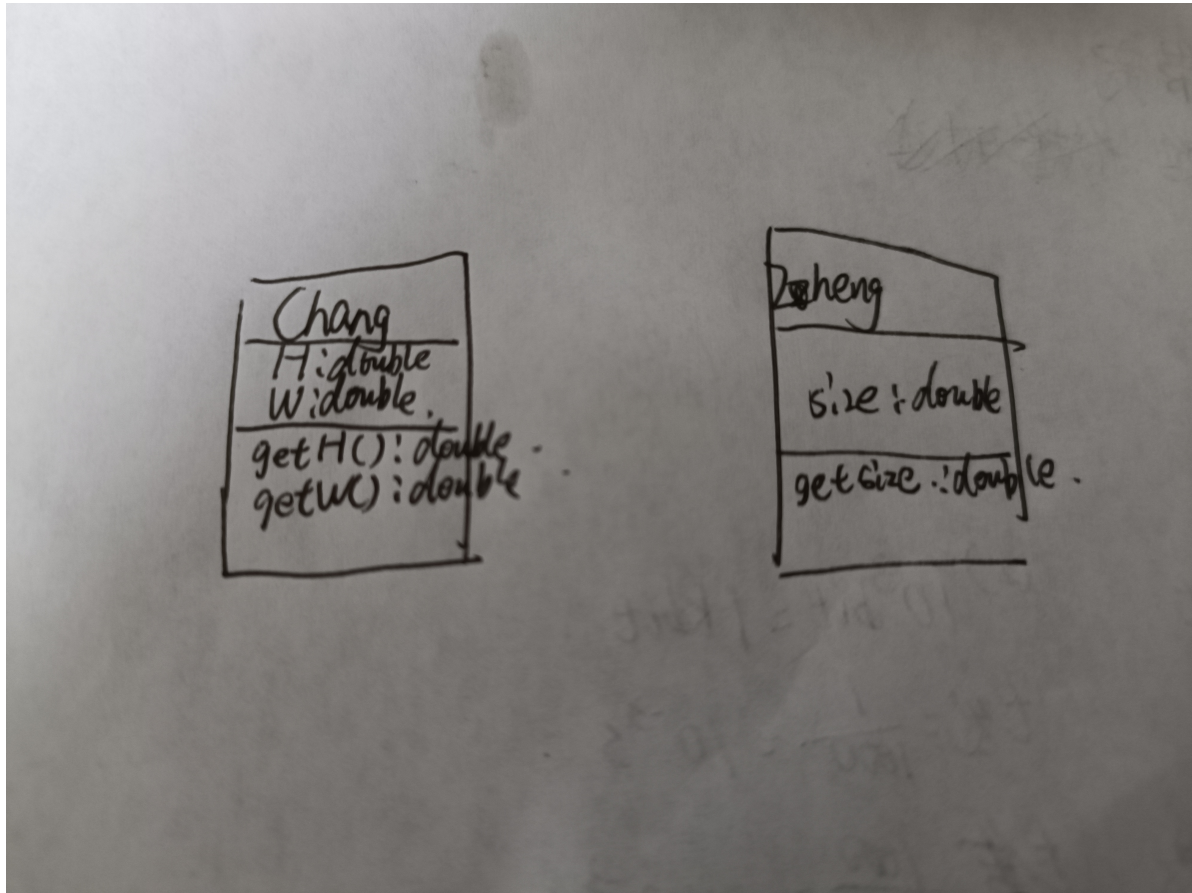


1.2

UML



Chang

```
package work_1_2;

public class Chang {
    private double h;
    private double w;
    public double getH() {
        return h;
    }
    public void setH(double h) {
        this.h = h;
    }
    public double getW() {
        return w;
    }
    public void setW(double w) {
        this.w = w;
    }
}
```

Zheng

```
package work_1_2;

public class Zheng {
    private double size;

    public double getSize() {
        return size;
    }

    public void setSize(double size) {
        this.size = size;
    }
}
```

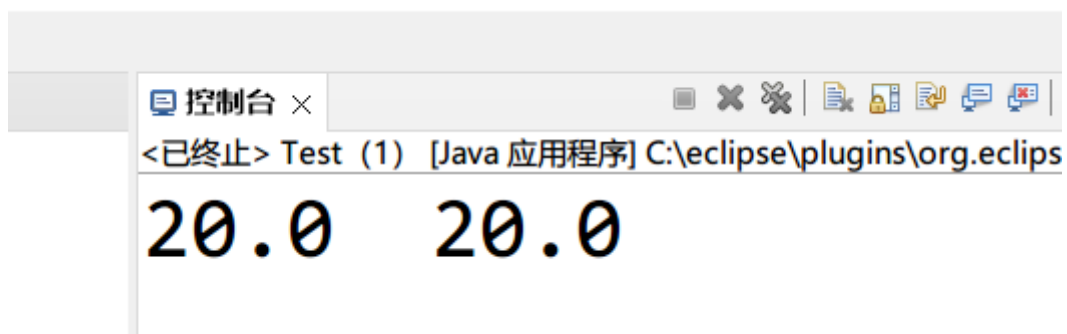
Test

```
package work_1_2;

public class Test {
    public static void resize(Chang chang) {
        while(chang.getH()>chang.getW()) {
            chang.setH(chang.getH()-1);
        }
        System.out.println(chang.getH()+" "+chang.getW());
    }

    public static void main(String []args) {
        Chang chang=new Chang();
        chang.setH(30);
        chang.setW(20);
        resize(chang);
    }
}
```

测试



思考与总结

思考

我们知道，正方形是特殊的长方形，所以在设计中想当然使用继承，但是在本例子中是有问题的，在这个例子中，我们应该是单独把这两个实现，因为不然会违反单一设计原则

总结

我们在平时也要能够实现

- 单一职责原则：一个类只做一件事情
- 开-闭：增加程序的功能时，不应该改动原有代码，只要增加新的代码即可
- 里氏代换：限制变化的影响范围组合优先：优先使用组合，而不是
- 继承依赖倒置：用抽象类来声明变量等
- 迪米特法则：耦合尽量松散
- 接口隔离：客户不应该依赖他们不用到的方法，只给每个客户它所需要的接口