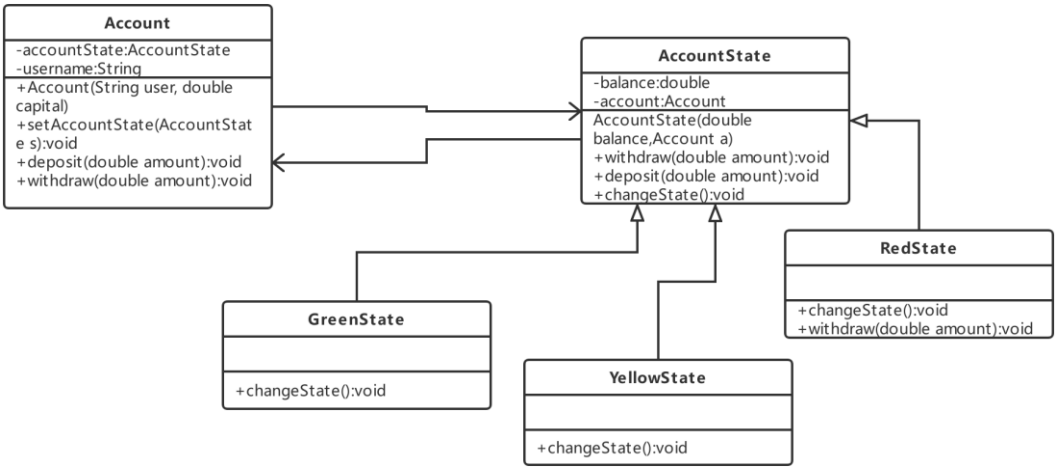



```java
package StatePattern;

public class Account {
    AccountState  accountState;
    public void setAccountState(AccountState accountState) {
        this.accountState = accountState;
    }
    String username;
    Account(String user, double balance ){
        this.username = user;
        this.accountState = new GreenState(balance,this);
    }
    public void withdraw(double amount) {
        //取款函数
        System.out.println(this.username+" 欢迎！！！");
        this.accountState.withdraw(amount);
    }
    public void deposit(double amount) {
        //存款函数
        System.out.println(this.username+" 欢迎！！！");
        this.accountState.deposit(amount);
    }
}
```

```java
package StatePattern;

public abstract class AccountState {
    Account account;
    double balance;
    public AccountState(double balance, Account a ) {
        // TODO 自动生成的构造函数存根
        this.account = a;
        this.balance= balance;
    }
    public abstract void  changeState();
    public void withdraw(double amount) {
        //取款函数
        this.balance -= amount;
        System.out.println("成功取款 "+amount+" 元");
        System.out.println("余额为 "+this.balance+" 元");
        changeState();
    }
    public void deposit(double amount) {
        this.balance += amount;
        System.out.println("成功存款 "+amount+" 元");
        System.out.println("余额为 "+this.balance+" 元");
        changeState();
    }
}


```

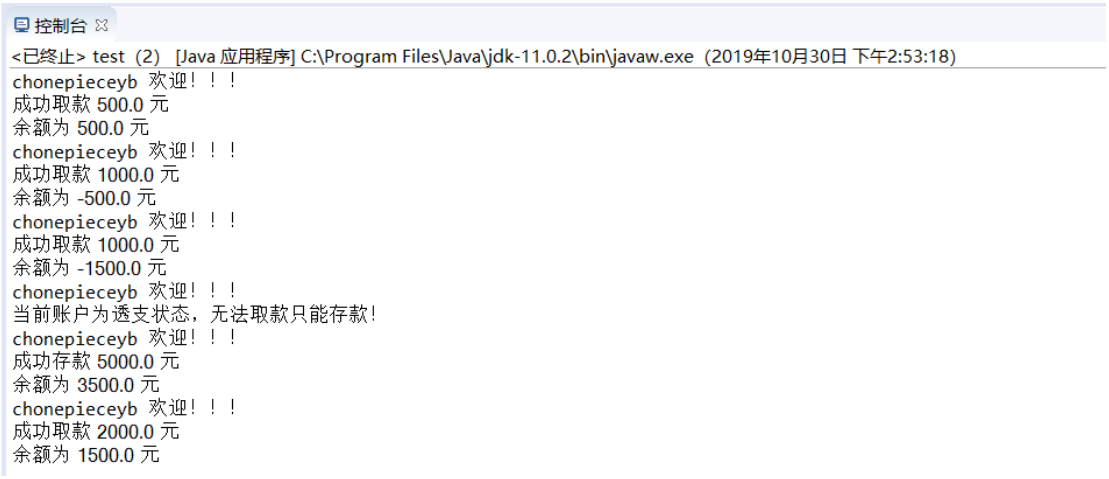
```java
package StatePattern;

public class GreenState extends AccountState {

    public GreenState(double balance, Account a) {
        super(balance, a);
        // TODO 自动生成的构造函数存根
    }

    @Override
    public void changeState() {
        // TODO 自动生成的方法存根
        if (this.balance<0 && this.balance >=-1000) {
            this.account.setAccountState(new YellowState(this.balance,this.account));
        }else if (this.balance<-1000) {
            this.account.setAccountState(new RedState(this.balance,this.account));
        }
    }

}

```

```java
package StatePattern;

public class YellowState extends AccountState {

    public YellowState(double balance, Account a) {
        super(balance, a);
        // TODO 自动生成的构造函数存根
    }

    @Override
    public void changeState() {
        // TODO 自动生成的方法存根
        if (this.balance>=0) {
            this.account.setAccountState(new GreenState(this.balance,this.account));
        }else if (this.balance<-1000) {
            this.account.setAccountState(new RedState(this.balance,this.account));
        }
    }

}

```

```java
package StatePattern;

public class RedState extends AccountState {

    public RedState(double balance, Account a) {
        super(balance, a);
        // TODO 自动生成的构造函数存根
    }

    @Override
    public void changeState() {
        // TODO 自动生成的方法存根
        if (this.balance>=0) {
            this.account.setAccountState(new GreenState(this.balance,this.account));
        }else if (this.balance>=-1000 && this.balance<0) {
            this.account.setAccountState(new YellowState(this.balance,this.account));
        }
    }
    public void withdraw(double amount) {
        System.out.println("当前账户为透支状态，无法取款只能存款！");
    }

}
```

```java
package StatePattern;

public class test {

    public static void main(String[] args) {
        // TODO 自动生成的方法存根
        // 客户端类

        Account myAccount = new Account("chonepieceyb",1000);
        myAccount.withdraw(500);
        myAccount.withdraw(1000);
        myAccount.withdraw(1000);
        myAccount.withdraw(1000);
        myAccount.deposit(5000);
        myAccount.withdraw(2000);
    }

}

```

结果：


控制台
<已终止> test（2） [Java 应用程序] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe（2019年10月30日 下午2:53:18）
chonepieceyb 欢迎！！！
成功取款 500.0 元
余额为 500.0 元
chonepieceyb 欢迎！！！
成功取款 1000.0 元
余额为 -500.0 元
chonepieceyb 欢迎！！！
成功取款 1000.0 元
余额为 -1500.0 元
chonepieceyb 欢迎！！！
当前账户为透支状态，无法取款只能存款！
chonepieceyb 欢迎！！！
成功存款 5000.0 元
余额为 3500.0 元
chonepieceyb 欢迎！！！
成功取款 2000.0 元
余额为 1500.0 元