|  |  |
| --- | --- |
| DESIGN PATTERNS  Lab Mid Term | Mubeen Khalid  SP21-BSE-015 |

Source Code:

//interface

1. Account.java

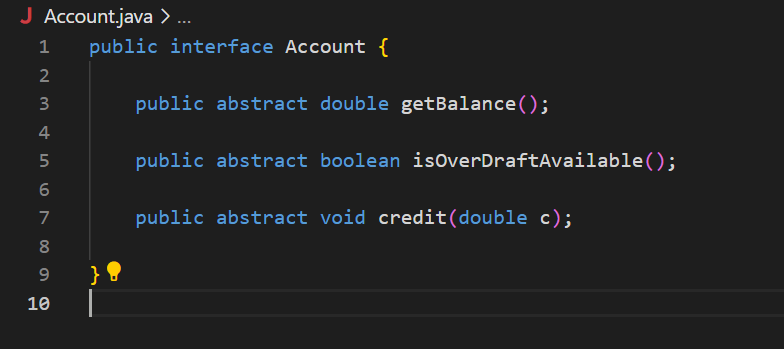
public interface Account {

    public abstract double getBalance();

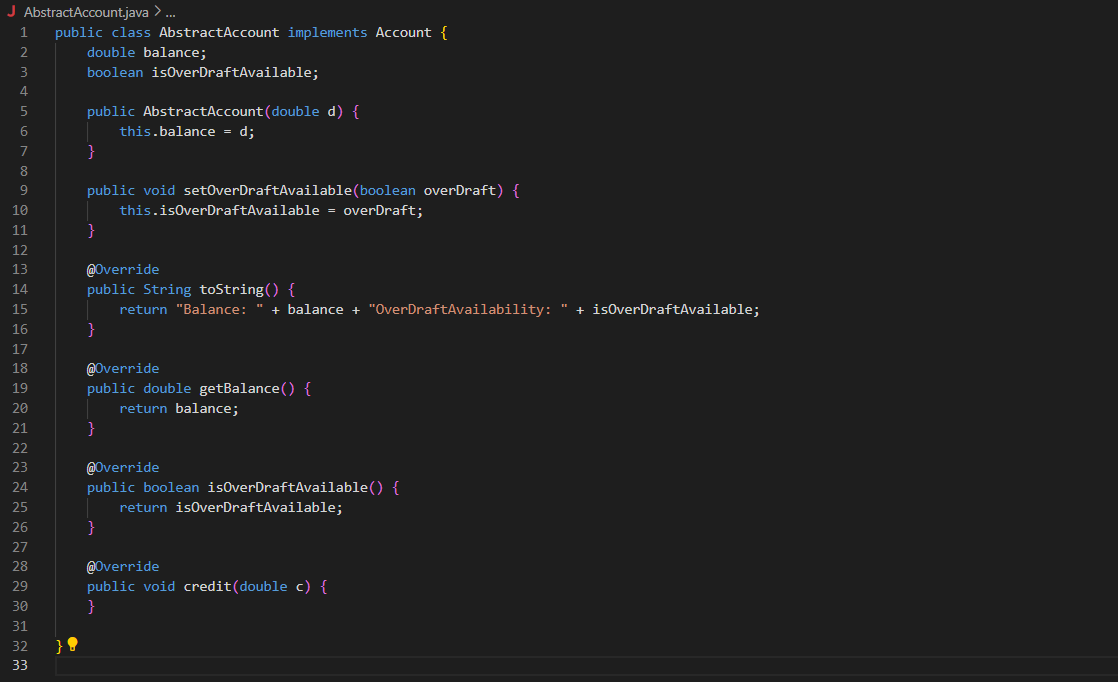
    public abstract boolean isOverDraftAvailable();

    public abstract void credit(double c);

}



1. AbstractAccount.java
2. public class AbstractAccount implements Account {
3. double balance;
4. boolean isOverDraftAvailable;
5. public AbstractAccount(double d) {
6. this.balance = d;
7. }
8. public void setOverDraftAvailable(boolean overDraft) {
9. this.isOverDraftAvailable = overDraft;
10. }
11. @Override
12. public String toString() {
13. return "Balance: " + balance + "OverDraftAvailability: " + isOverDraftAvailable;
14. }
15. @Override
16. public double getBalance() {
17. return balance;
18. }
19. @Override
20. public boolean isOverDraftAvailable() {
21. return isOverDraftAvailable;
22. }
23. @Override
24. public void credit(double c) {
25. }
26. }



3. PlatinumAccount.java

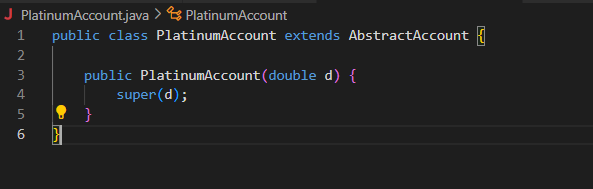
public class PlatinumAccount extends AbstractAccount {

    public PlatinumAccount(double d) {

        super(d);

    }

}



4. StandardAccount.java

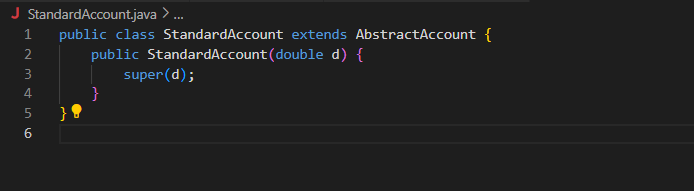
public class StandardAccount extends AbstractAccount {

    public StandardAccount(double d) {

        super(d);

    }

}



5. OffshoreAccount.java

public class OffshoreAccount {

    double balance;

    double TAX\_RATE = 2;

    public OffshoreAccount(double b) {

        this.balance = b;

    }

    public double getTaxRate() {

        return TAX\_RATE;

    }

    public double getOffshoreBalance() {

        return balance;

    }

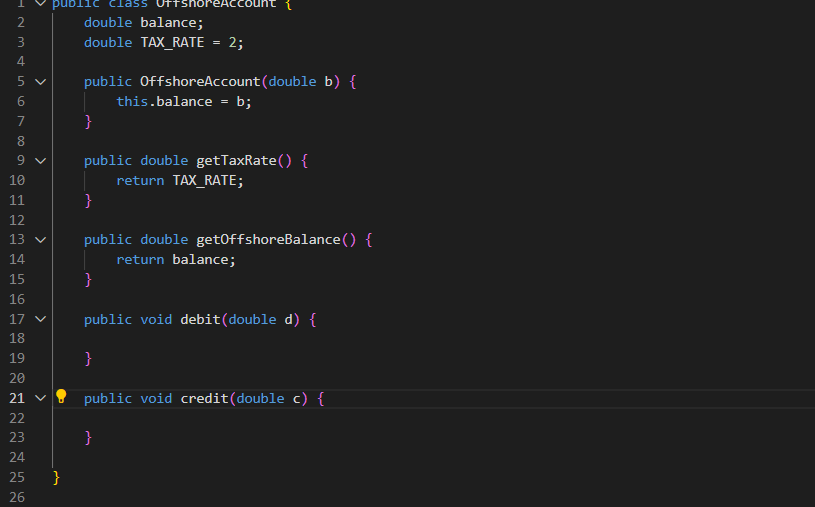
    public void debit(double d) {

    }

    public void credit(double c) {

    }

}



6. AccountAdapter.java

public class AccountAdapter extends AbstractAccount {

    @SuppressWarnings("unused")

    private OffshoreAccount offshoreAccount;

    public AccountAdapter(double d, OffshoreAccount offshoreAccount) {

        super(d);

        this.offshoreAccount = offshoreAccount;

    }

    @Override

    public double getBalance() {

        return balance;

    }

    @Override

    public boolean isOverDraftAvailable() {

        return false;

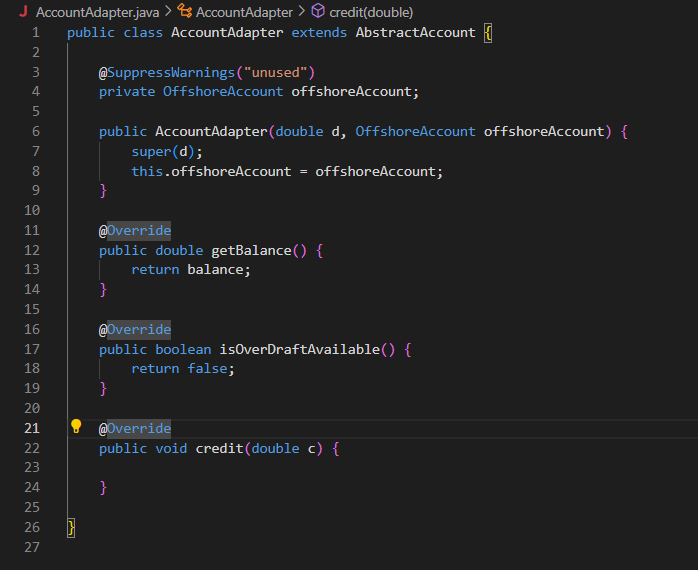
    }

    @Override

    public void credit(double c) {

    }

}



7. Main.java

public class Main {

    public static void main(String[] args) {

        Account StandardAccount = new StandardAccount(23);

        Account PlatinumAccount = new StandardAccount(53);

        System.out.println("Balance: " + StandardAccount.getBalance());

        System.out.println("Balance: " + PlatinumAccount.getBalance());

        OffshoreAccount offshoreAccount = new OffshoreAccount(42);

        System.out.println(offshoreAccount.getTaxRate());

        AccountAdapter adapter = new AccountAdapter(50, offshoreAccount);

        System.out.println(adapter.getBalance(););

    }

}

