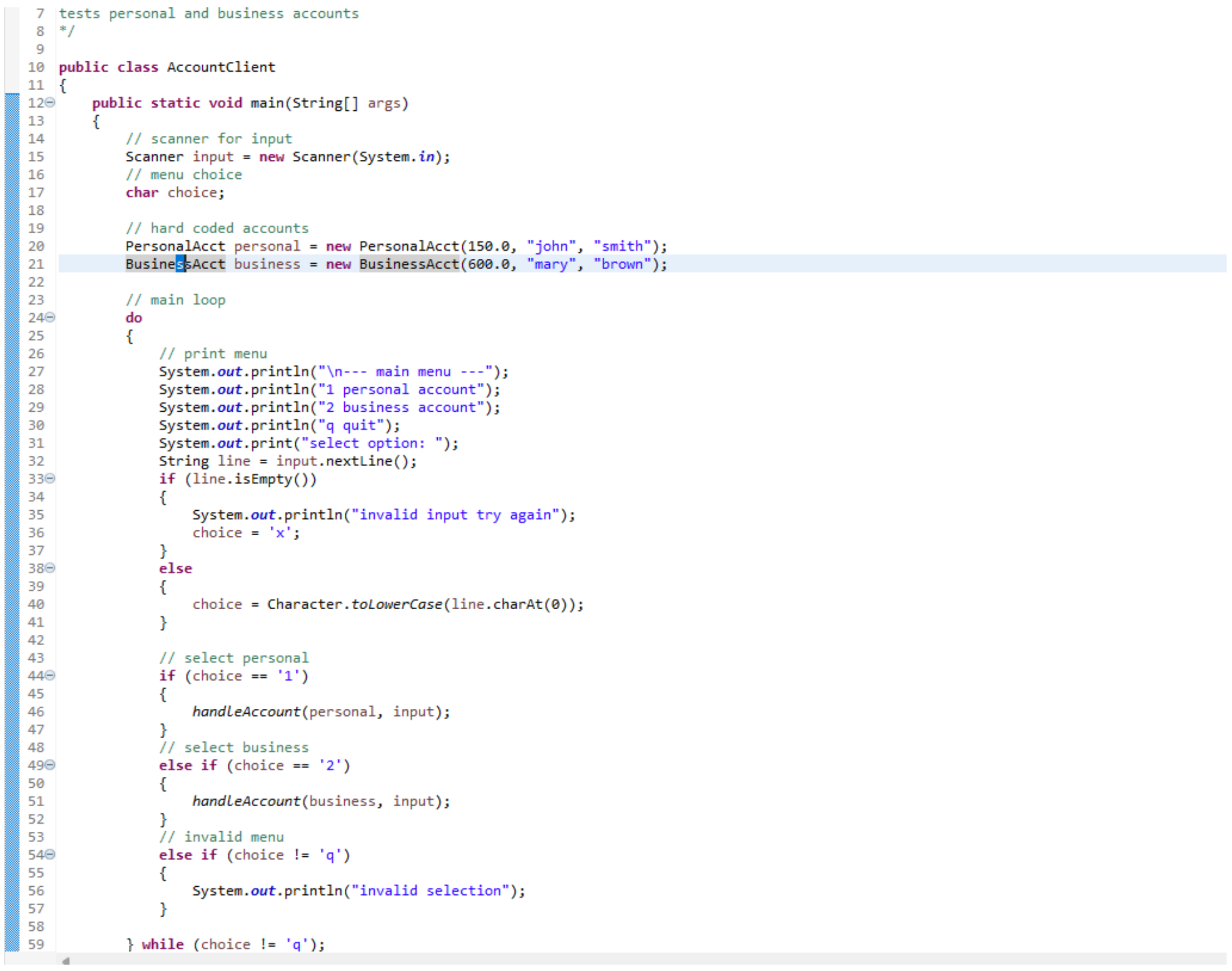
Credit
3130
Assignment
E2 Account

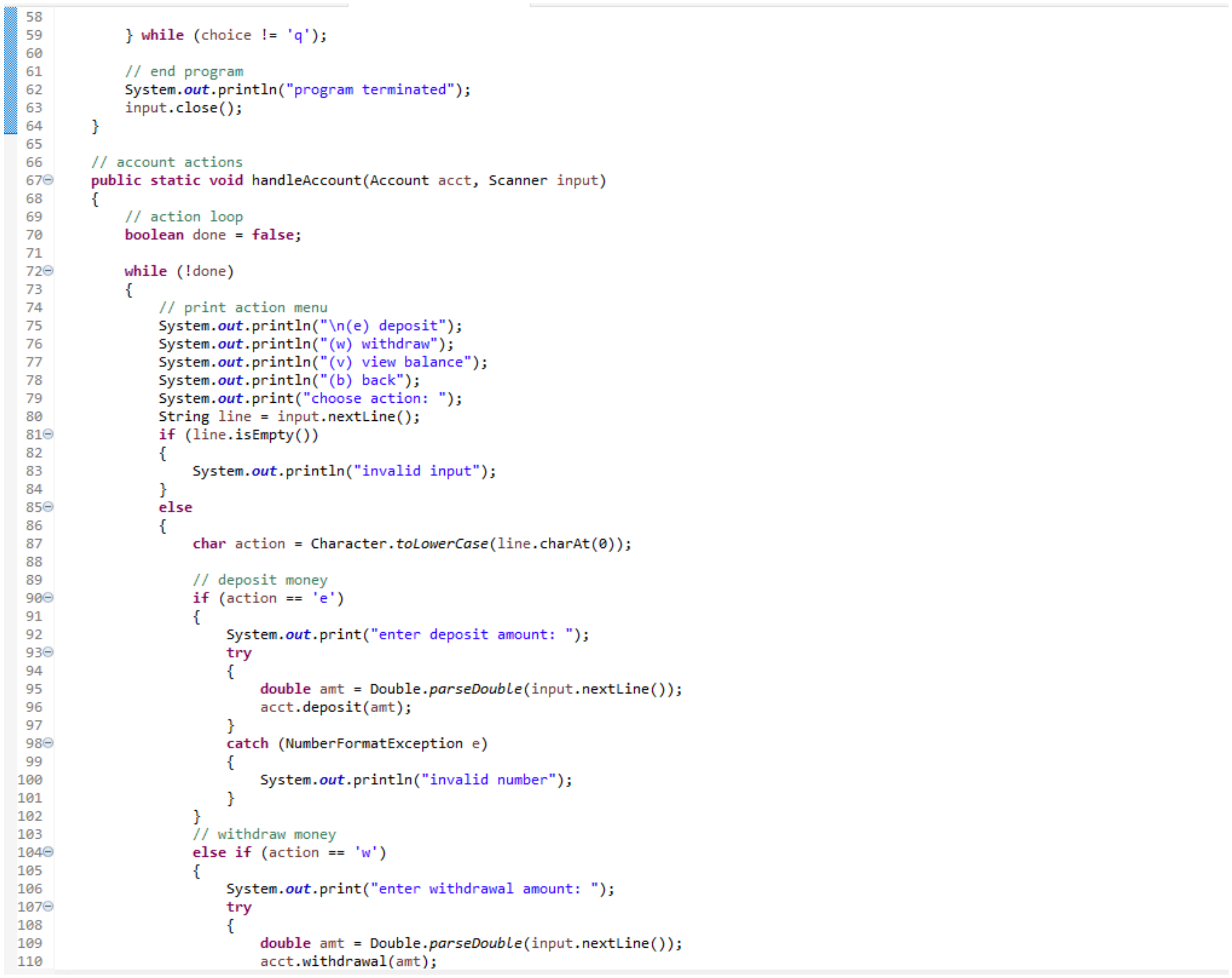
tests personal and business accounts

How has your program changed from planning to coding to now? Please explain?

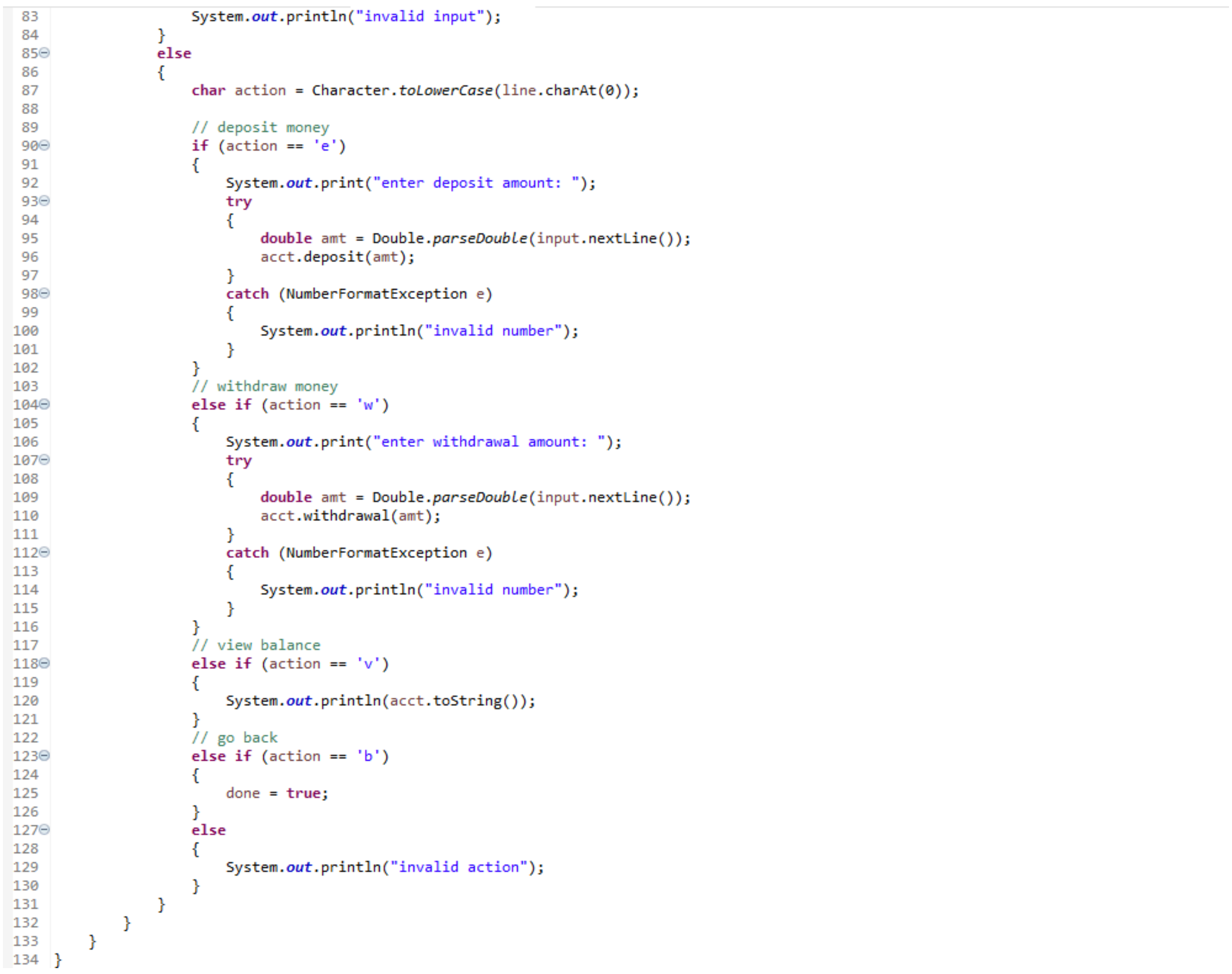
Tester



```java
tests personal and business accounts
*/

public class AccountClient
{
    public static void main(String[] args)
    {
        // scanner for input
        Scanner input = new Scanner(System.in);
        // menu choice
        char choice;

        // hard coded accounts
        PersonalAcct personal = new PersonalAcct(150.0, "john", "smith");
        BusinessAcct business = new BusinessAcct(600.0, "mary", "brown");

        // main loop
        do
        {
            // print menu
            System.out.println("\n--- main menu ---");
            System.out.println("1 personal account");
            System.out.println("2 business account");
            System.out.println("q quit");
            System.out.print("select option: ");
            String line = input.nextLine();
            if (line.isEmpty())
            {
                System.out.println("invalid input try again");
                choice = 'x';
            }
            else
            {
                choice = Character.toLowerCase(line.charAt(0));
            }

            // select personal
            if (choice == '1')
            {
                handleAccount(personal, input);
            }
            // select business
            else if (choice == '2')
            {
                handleAccount(business, input);
            }
            // invalid menu
            else if (choice != 'q')
            {
                System.out.println("invalid selection");
            }

        } while (choice != 'q');
```

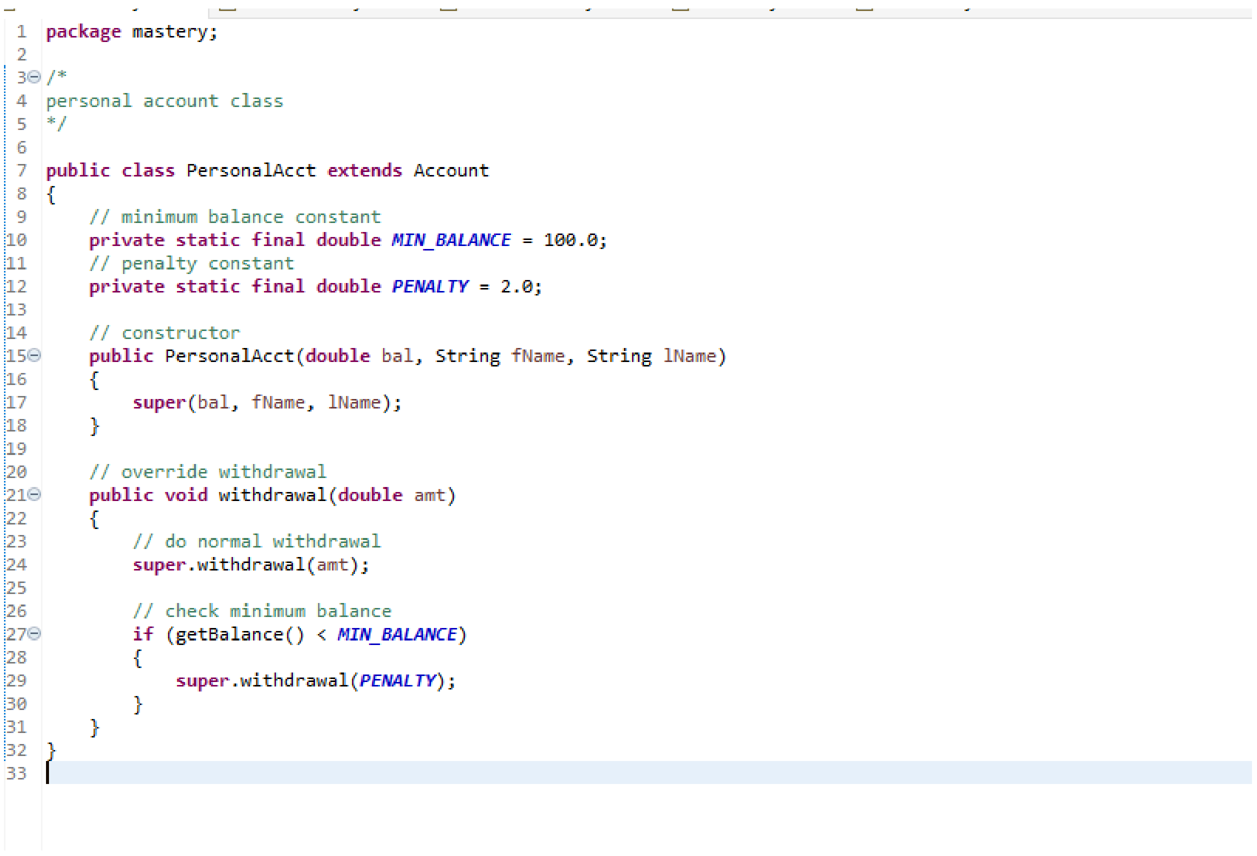


```java

        } while (choice != 'q');

        // end program
        System.out.println("program terminated");
        input.close();
    }

    // account actions
    public static void handleAccount(Account acct, Scanner input)
    {
        // action loop
        boolean done = false;

        while (!done)
        {
            // print action menu
            System.out.println("\n(e) deposit");
            System.out.println("(w) withdraw");
            System.out.println("(v) view balance");
            System.out.println("(b) back");
            System.out.print("choose action: ");
            String line = input.nextLine();
            if (line.isEmpty())
            {
                System.out.println("invalid input");
            }
            else
            {
                char action = Character.toLowerCase(line.charAt(0));

                // deposit money
                if (action == 'e')
                {
                    System.out.print("enter deposit amount: ");
                    try
                    {
                        double amt = Double.parseDouble(input.nextLine());
                        acct.deposit(amt);
                    }
                    catch (NumberFormatException e)
                    {
                        System.out.println("invalid number");
                    }
                }
                // withdraw money
                else if (action == 'w')
                {
                    System.out.print("enter withdrawal amount: ");
                    try
                    {
                        double amt = Double.parseDouble(input.nextLine());
                        acct.withdrawal(amt);
```



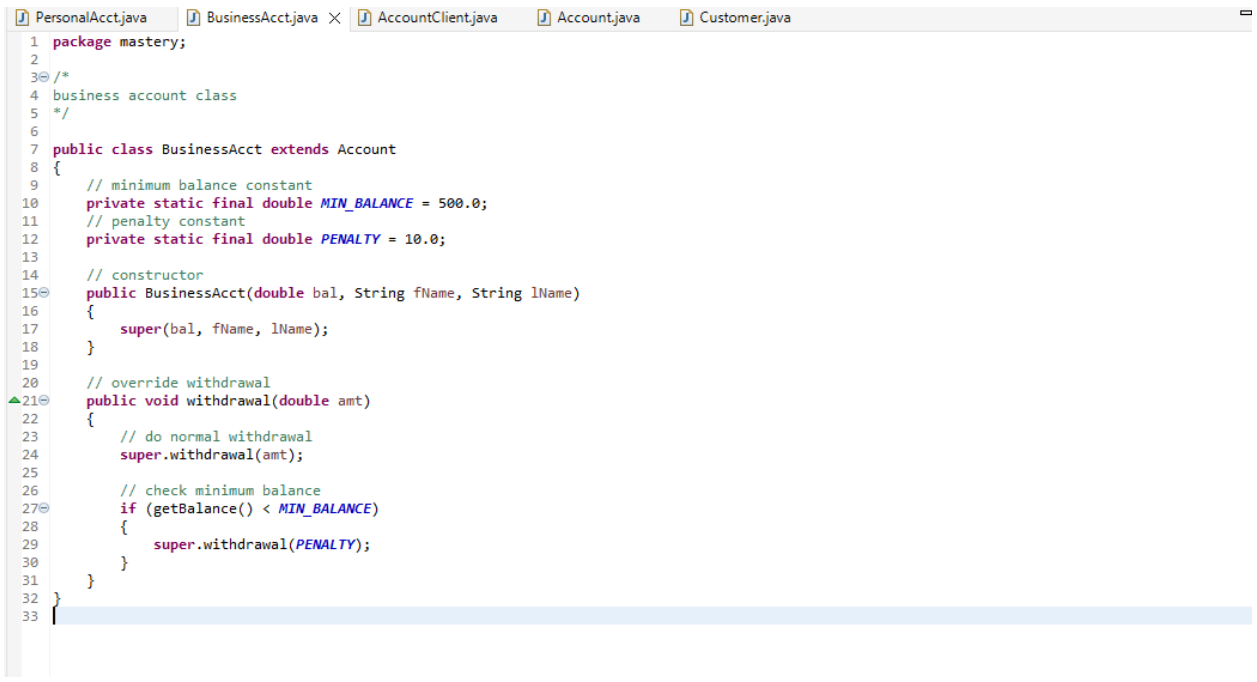
```java
                System.out.println("invalid input");
            }
            else
            {
                char action = Character.toLowerCase(line.charAt(0));

                // deposit money
                if (action == 'e')
                {
                    System.out.print("enter deposit amount: ");
                    try
                    {
                        double amt = Double.parseDouble(input.nextLine());
                        acct.deposit(amt);
                    }
                    catch (NumberFormatException e)
                    {
                        System.out.println("invalid number");
                    }
                }
                // withdraw money
                else if (action == 'w')
                {
                    System.out.print("enter withdrawal amount: ");
                    try
                    {
                        double amt = Double.parseDouble(input.nextLine());
                        acct.withdrawal(amt);
                    }
                    catch (NumberFormatException e)
                    {
                        System.out.println("invalid number");
                    }
                }
                // view balance
                else if (action == 'v')
                {
                    System.out.println(acct.toString());
                }
                // go back
                else if (action == 'b')
                {
                    done = true;
                }
                else
                {
                    System.out.println("invalid action");
                }
            }
        }
    }
}
```

Persona



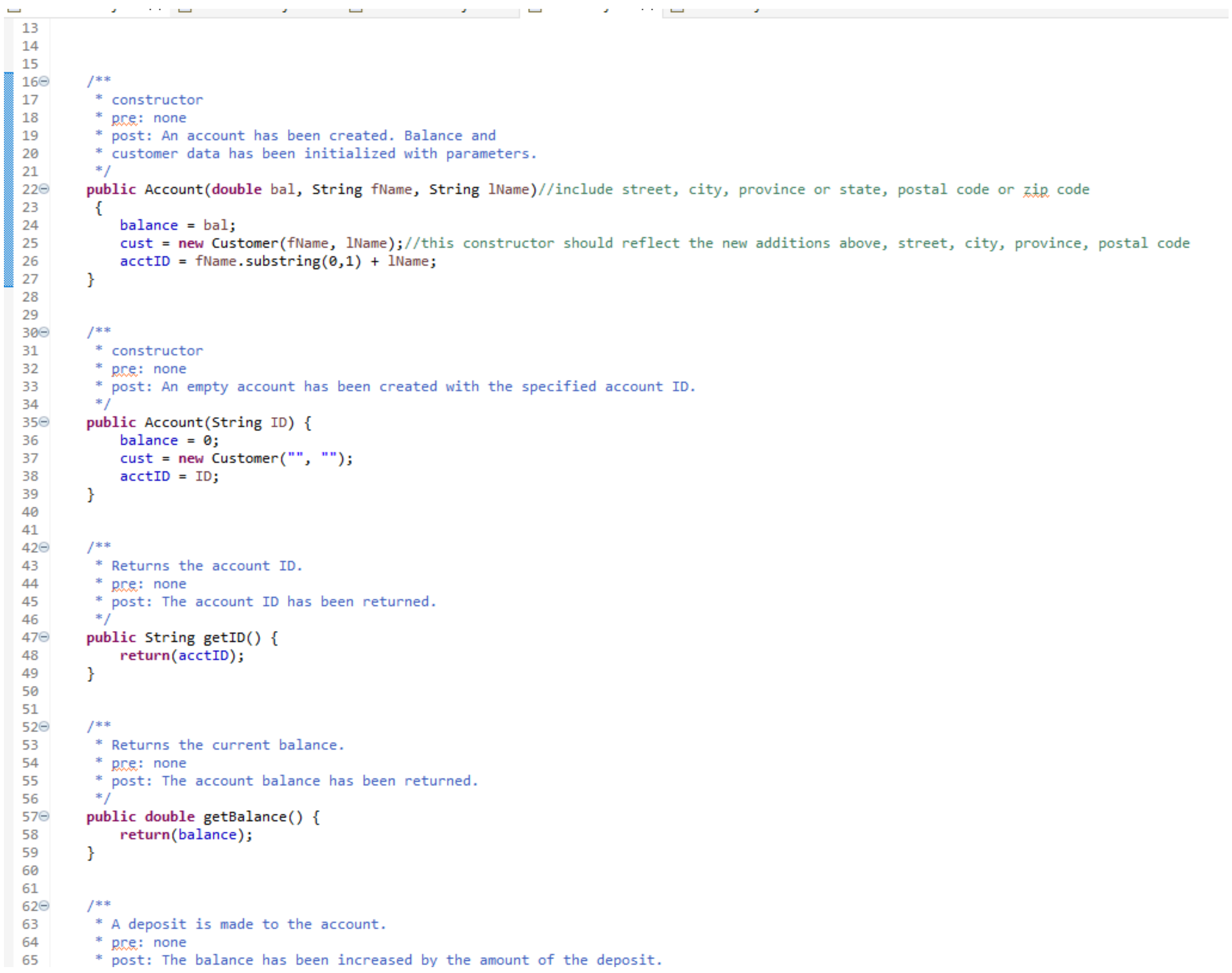
```java
package mastery;

/*
personal account class
*/

public class PersonalAcct extends Account
{
    // minimum balance constant
    private static final double MIN_BALANCE = 100.0;
    // penalty constant
    private static final double PENALTY = 2.0;

    // constructor
    public PersonalAcct(double bal, String fName, String lName)
    {
        super(bal, fName, lName);
    }

    // override withdrawal
    public void withdrawal(double amt)
    {
        // do normal withdrawal
        super.withdrawal(amt);

        // check minimum balance
        if (getBalance() < MIN_BALANCE)
        {
            super.withdrawal(PENALTY);
        }
    }
}
```

Bussienius

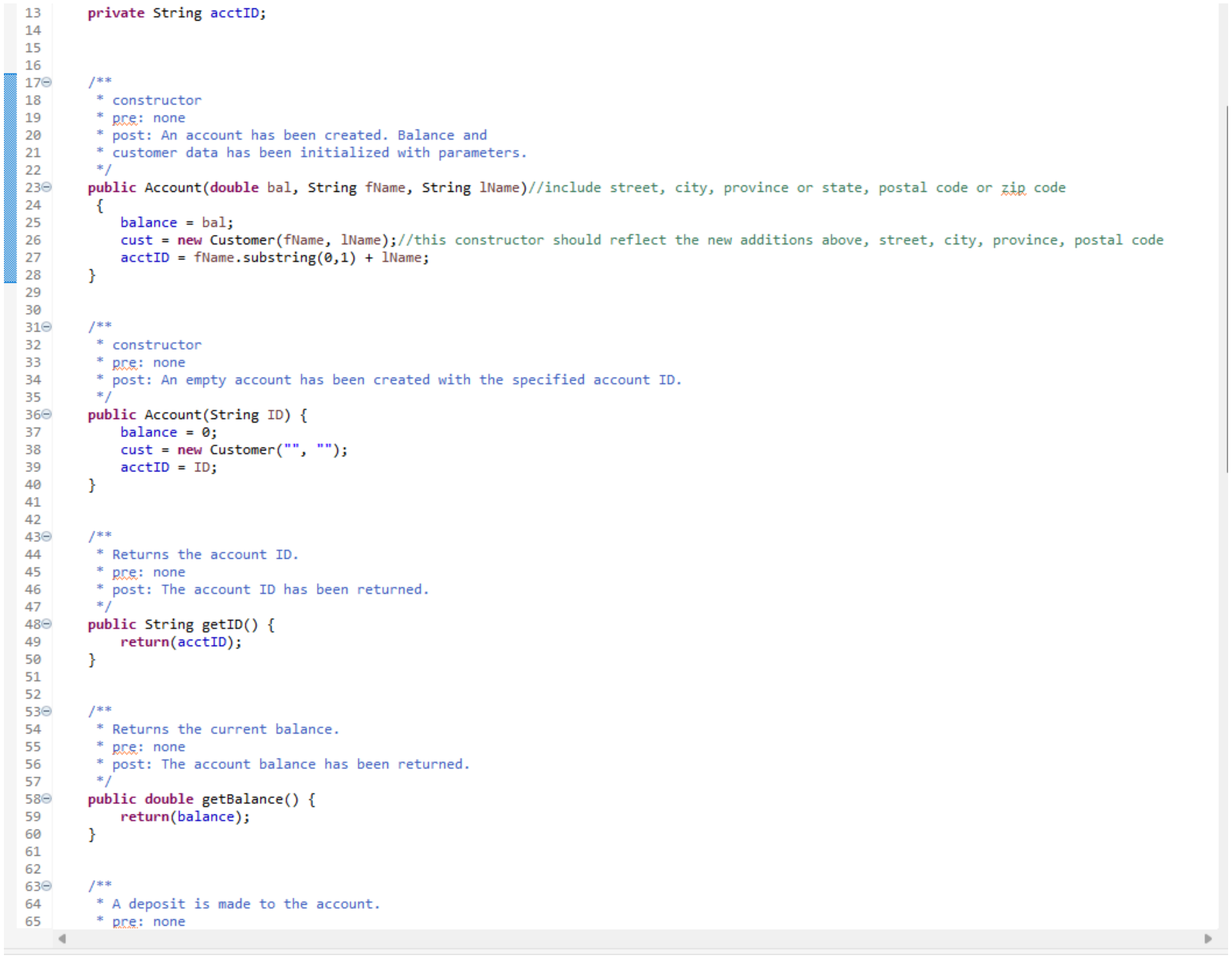


```java
package mastery;

/*
business account class
*/

public class BusinessAcct extends Account
{
    // minimum balance constant
    private static final double MIN_BALANCE = 500.0;
    // penalty constant
    private static final double PENALTY = 10.0;

    // constructor
    public BusinessAcct(double bal, String fName, String lName)
    {
        super(bal, fName, lName);
    }

    // override withdrawal
    public void withdrawal(double amt)
    {
        // do normal withdrawal
        super.withdrawal(amt);

        // check minimum balance
        if (getBalance() < MIN_BALANCE)
        {
            super.withdrawal(PENALTY);
        }
    }
}
```

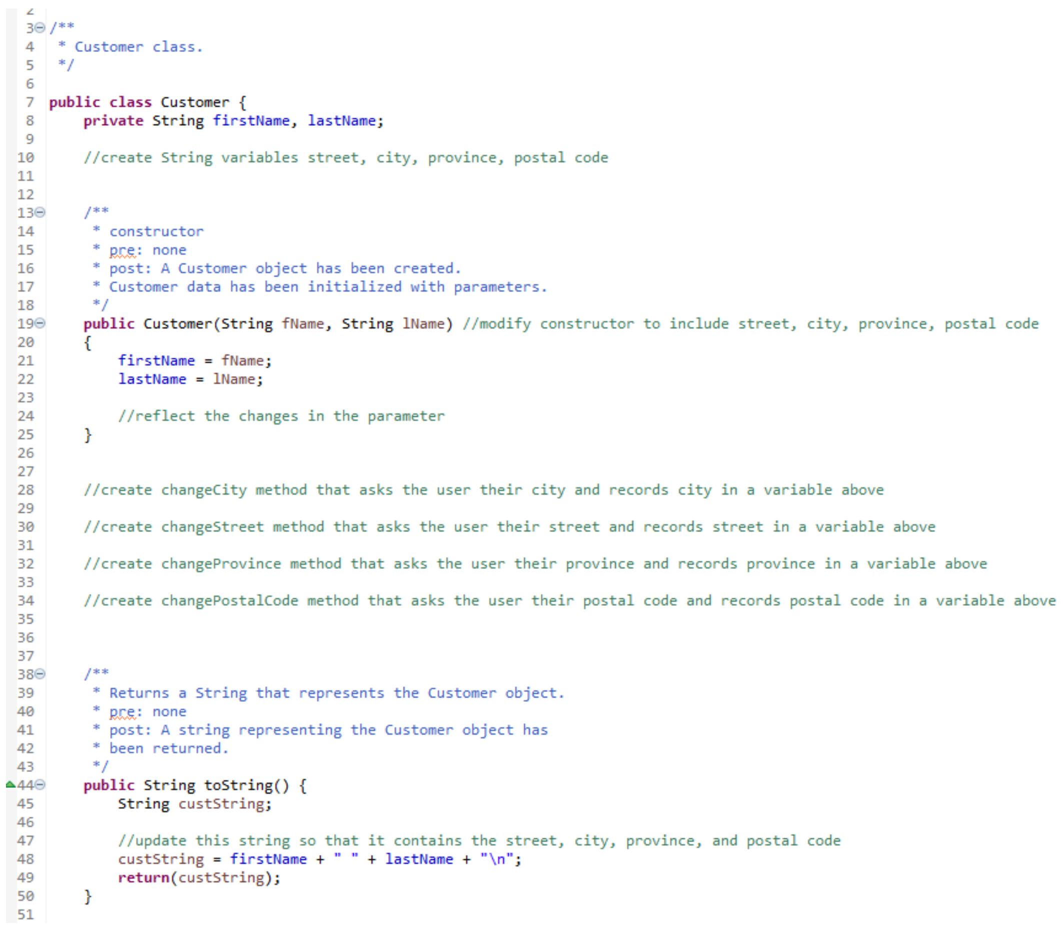
Account



```



    /**
     * constructor
     * pre: none
     * post: An account has been created. Balance and
     * customer data has been initialized with parameters.
     */
    public Account(double bal, String fName, String lName)//include street, city, province or state, postal code or zip code
     {
        balance = bal;
        cust = new Customer(fName, lName);//this constructor should reflect the new additions above, street, city, province, postal code
        acctID = fName.substring(0,1) + lName;
    }


    /**
     * constructor
     * pre: none
     * post: An empty account has been created with the specified account ID.
     */
    public Account(String ID) {
        balance = 0;
        cust = new Customer("", "");
        acctID = ID;
    }


    /**
     * Returns the account ID.
     * pre: none
     * post: The account ID has been returned.
     */
    public String getID() {
        return(acctID);
    }


    /**
     * Returns the current balance.
     * pre: none
     * post: The account balance has been returned.
     */
    public double getBalance() {
        return(balance);
    }


    /**
     * A deposit is made to the account.
     * pre: none
     * post: The balance has been increased by the amount of the deposit.
```


```java
    private String acctID;



    /**
     * constructor
     * pre: none
     * post: An account has been created. Balance and
     * customer data has been initialized with parameters.
     */
    public Account(double bal, String fName, String lName)//include street, city, province or state, postal code or zip code
     {
        balance = bal;
        cust = new Customer(fName, lName);//this constructor should reflect the new additions above, street, city, province, postal code
        acctID = fName.substring(0,1) + lName;
    }


    /**
     * constructor
     * pre: none
     * post: An empty account has been created with the specified account ID.
     */
    public Account(String ID) {
        balance = 0;
        cust = new Customer("", "");
        acctID = ID;
    }


    /**
     * Returns the account ID.
     * pre: none
     * post: The account ID has been returned.
     */
    public String getID() {
        return(acctID);
    }


    /**
     * Returns the current balance.
     * pre: none
     * post: The account balance has been returned.
     */
    public double getBalance() {
        return(balance);
    }


    /**
     * A deposit is made to the account.
     * pre: none
```

Customer one



```java
/**
 * Customer class.
 */

public class Customer {
    private String firstName, lastName;

    //create String variables street, city, province, postal code


    /**
     * constructor
     * pre: none
     * post: A Customer object has been created.
     * Customer data has been initialized with parameters.
     */
    public Customer(String fName, String lName) //modify constructor to include street, city, province, postal code
    {
        firstName = fName;
        lastName = lName;

        //reflect the changes in the parameter
    }


    //create changeCity method that asks the user their city and records city in a variable above

    //create changeStreet method that asks the user their street and records street in a variable above

    //create changeProvince method that asks the user their province and records province in a variable above

    //create changePostalCode method that asks the user their postal code and records postal code in a variable above



    /**
     * Returns a String that represents the Customer object.
     * pre: none
     * post: A string representing the Customer object has
     * been returned.
     */
    public String toString() {
        String custString;

        //update this string so that it contains the street, city, province, and postal code
        custString = firstName + " " + lastName + "\n";
        return(custString);
    }

```