UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

SOFTWARE ENGINEERING

# OOP-LAB 7

SUBMITTED TO: ENGR.SIDRA SHAFFI

SUBMITTED BY: MUHAMMAD ARSALAN

REG NO: 20-SE-56

COURSE: OOP-LAB

DATED: October 6, 2021

# TASK

## Part 1

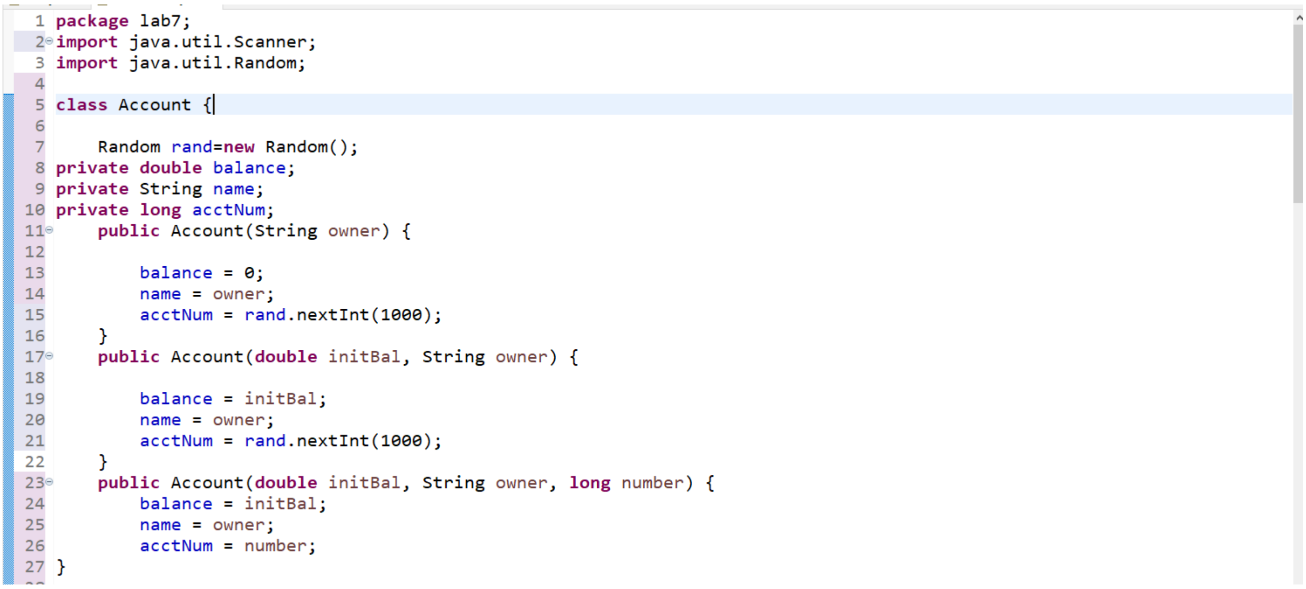
File Account.java contains a definition for a simple bank account class with methods to withdraw, deposit, get the balance and account number, and return a String representation. Note that the constructor for this class creates a random account number. Save this class to your directory and study it to see how it works. Then modify it as follows:

1. Overload the constructor as follows:

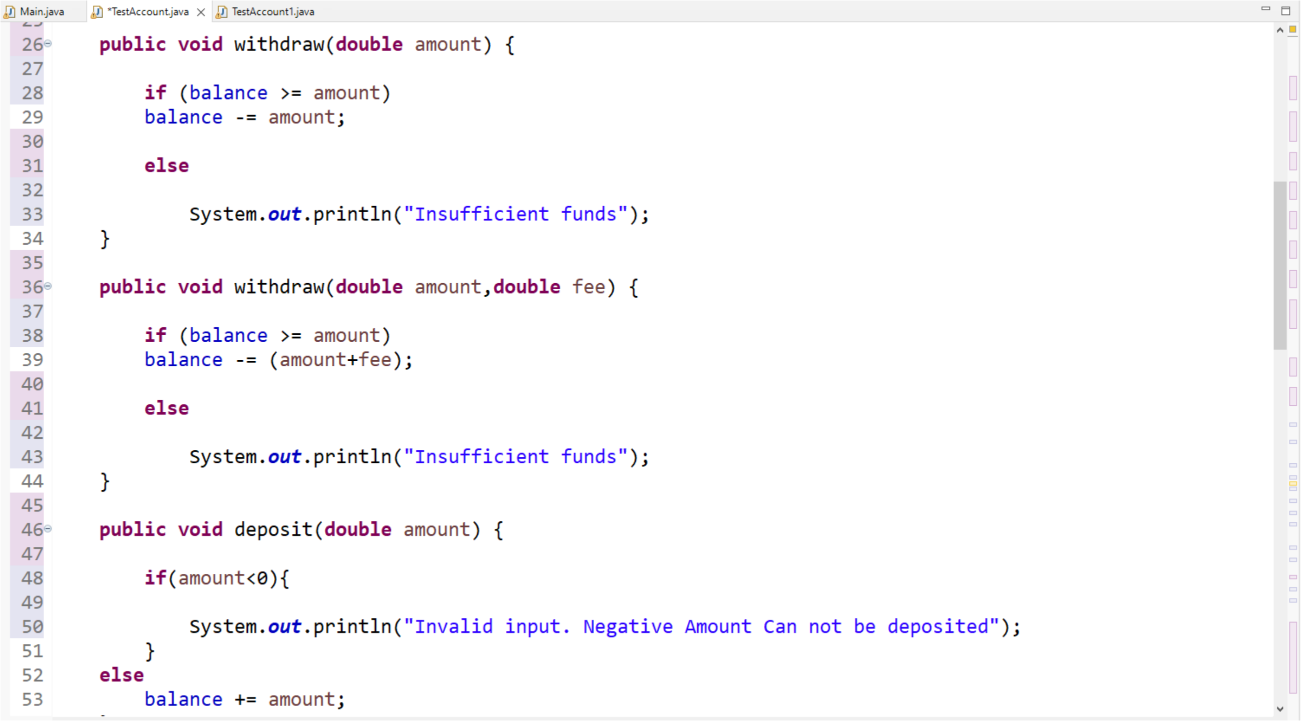
- public Account (double initBal, String owner, long number) – initializes the balance, owner, and account number as specified
- public Account (double initBal, String owner) – initializes the balance and owner as specified; randomly generates the account number.
- public Account (String owner) – initializes the owner as specified; sets the initial balance to 0 and randomly generates the account number.

## Code:

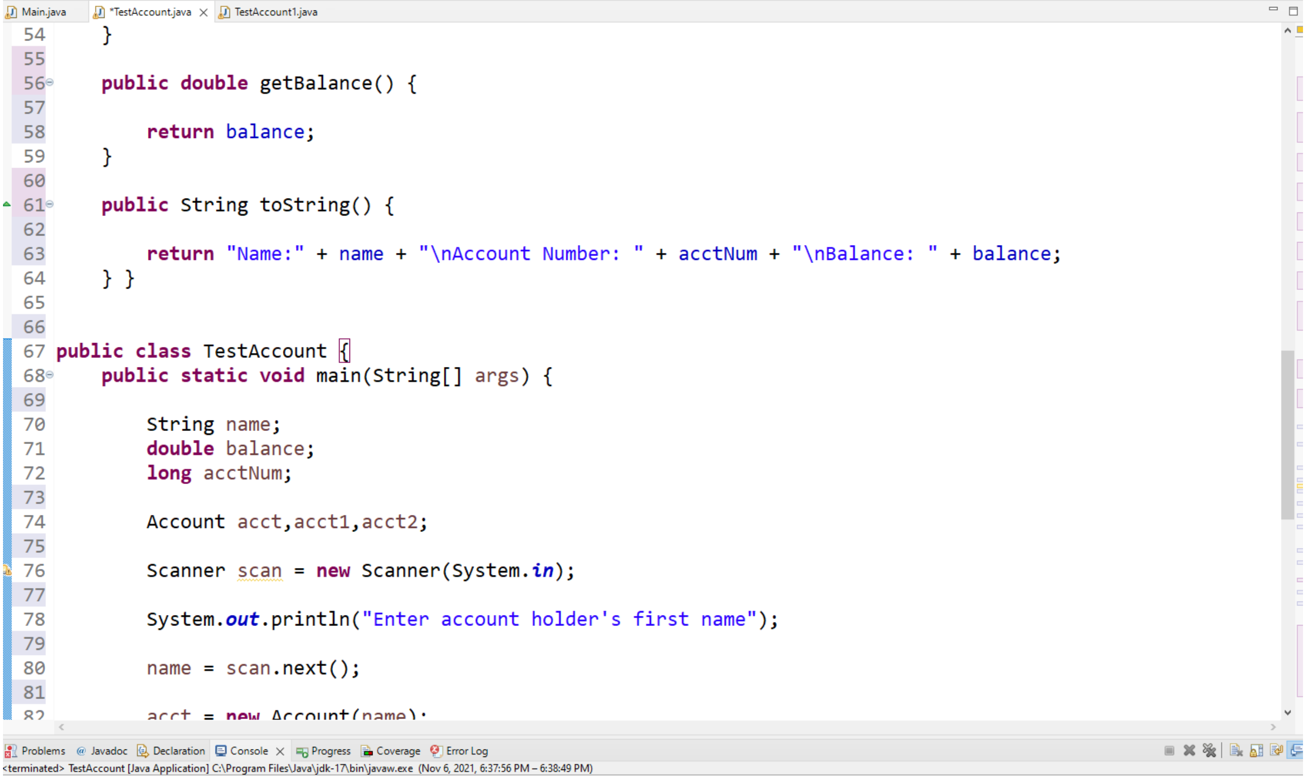
```java
package lab7;
import java.util.Scanner;
import java.util.Random;

class Account {

  Random rand=new Random();
private double balance;
private String name;
private long acctNum;
      public Account(String owner) {

            balance = 0;
            name = owner;
            acctNum = rand.nextInt(1000);
      }
      public Account(double initBal, String owner) {

            balance = initBal;
            name = owner;
            acctNum = rand.nextInt(1000);
      }
      public Account(double initBal, String owner, long number) {
            balance = initBal;
            name = owner;
            acctNum = number;
}

      public void withdraw(double amount) {

            if (balance >= amount)
            balance -= amount;

            else

                  System.out.println("Insufficient funds");
      }

      public void withdraw(double amount,double fee) {

            if (balance >= amount)
            balance -= (amount+fee);

            else

                  System.out.println("Insufficient funds");
      }

      public void deposit(double amount) {

            if(amount<0){

                  System.out.println("Invalid input. Negative Amount Can not be 
deposited");
            }
      else
            balance += amount;
      }

      public double getBalance() {

            return balance;
      }

      public String toString() {

            return "Name:" + name + "\nAccount Number: " + acctNum + "\nBalance: 
" + balance;
      } }


public class TestAccount {
      public static void main(String[] args) {

            String name;
            double balance;
            long acctNum;

            Account acct,acct1,acct2;

            Scanner scan = new Scanner(System.in);

            System.out.println("Enter account holder's first name");

            name = scan.next();

            acct = new Account(name);

            System.out.println("Account for " + name + ":");
            System.out.println(acct);
            System.out.println("\nEnter initial balance");

            balance = scan.nextDouble();

            acct1 = new Account(balance,name);
            System.out.println("Account for " + name + ":");
            System.out.println(acct1);
            System.out.println("\nEnter account number");

            acctNum = scan.nextLong();

            acct2 = new Account(balance,name,acctNum);
            System.out.println("Account for " + name + ":");
            System.out.println(acct2);
            System.out.print("\nDepositing 100 into account, balance is now ");

            acct2.deposit(100);
            System.out.println(acct2.getBalance());
            System.out.print("\nWithdrawing $25, balance is now ");

            acct2.withdraw(25);
            System.out.println(acct2.getBalance());
            System.out.print("\nWithdrawing $25 with $2 fee, balance is now ");

            acct2.withdraw(25,2);
            System.out.println(acct2.getBalance());
            System.out.println("\nBye!");
} }
```



```
package lab7;
import java.util.Scanner;
import java.util.Random;

class Account {

    Random rand=new Random();
private double balance;
private String name;
private long acctNum;
    public Account(String owner) {

        balance = 0;
        name = owner;
        acctNum = rand.nextInt(1000);
    }
    public Account(double initBal, String owner) {

        balance = initBal;
        name = owner;
        acctNum = rand.nextInt(1000);
    }
    public Account(double initBal, String owner, long number) {
        balance = initBal;
        name = owner;
        acctNum = number;
}
```

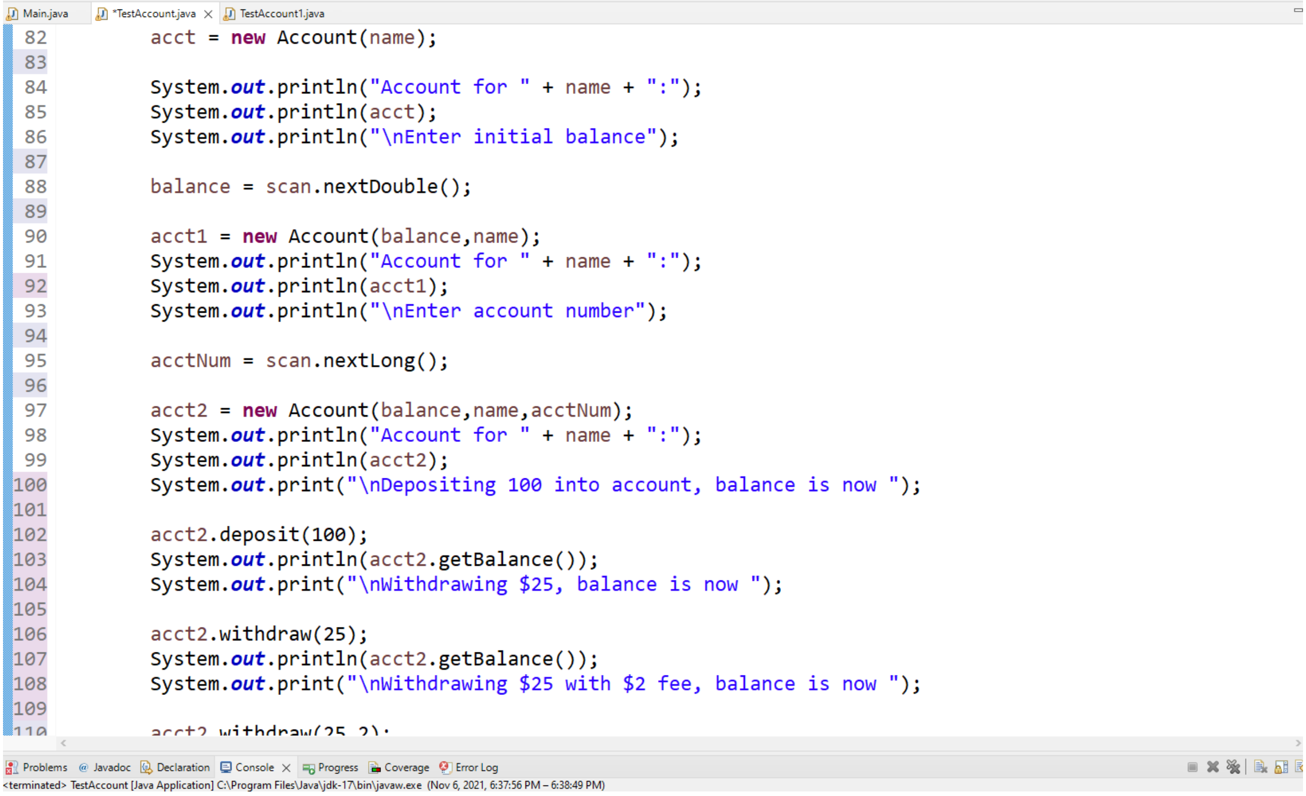


```
  public void withdraw(double amount) {

    if (balance >= amount)
    balance -= amount;

    else

      System.out.println("Insufficient funds");
  }

  public void withdraw(double amount,double fee) {

    if (balance >= amount)
    balance -= (amount+fee);

    else

      System.out.println("Insufficient funds");
  }

  public void deposit(double amount) {

    if(amount<0){

      System.out.println("Invalid input. Negative Amount Can not be deposited");
    }
  else
    balance += amount;
```



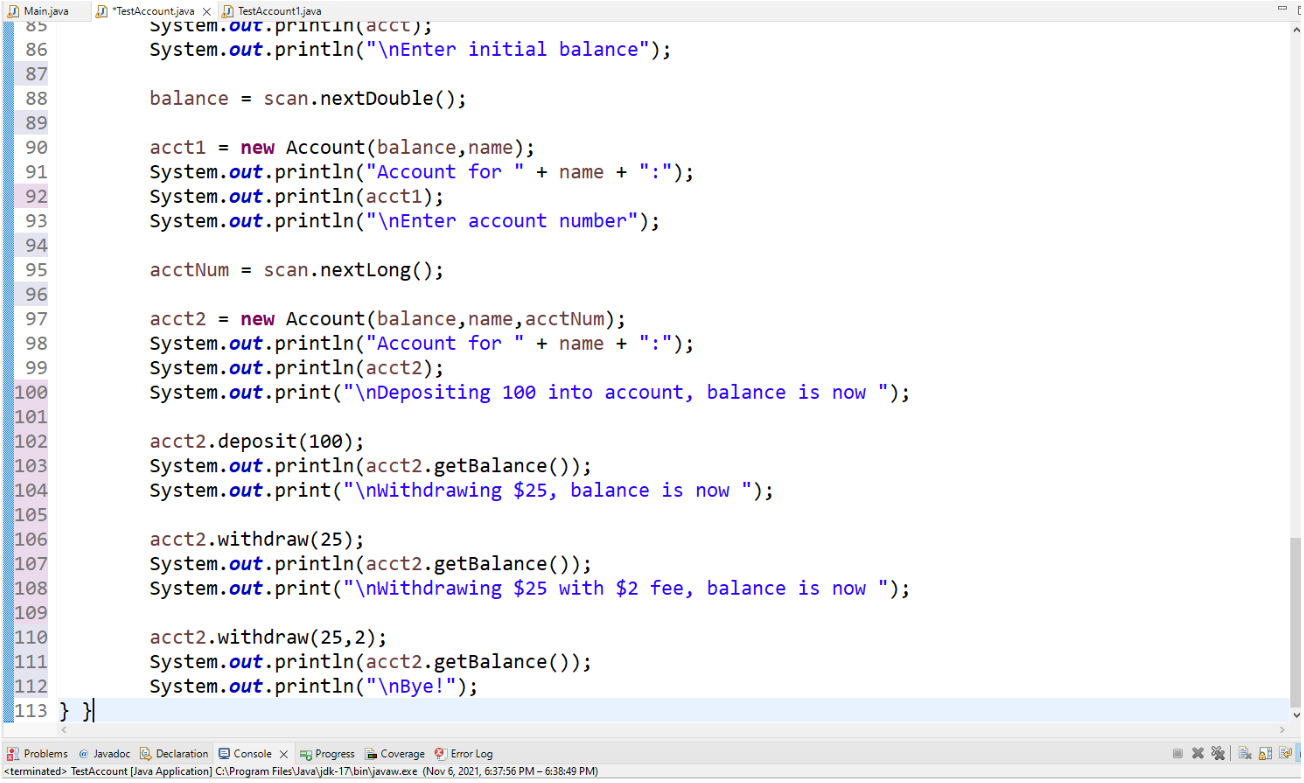
```
  }

  public double getBalance() {

    return balance;
  }

  public String toString() {

    return "Name:" + name + "\nAccount Number: " + acctNum + "\nBalance: " + balance;
  } }

public class TestAccount {
  public static void main(String[] args) {

    String name;
    double balance;
    long acctNum;

    Account acct,acct1,acct2;

    Scanner scan = new Scanner(System.in);

    System.out.println("Enter account holder's first name");

    name = scan.next();

    acct = new Account(name);
```

```
<terminated> TestAccount [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Nov 6, 2021, 6:37:56 PM – 6:38:49 PM)
```



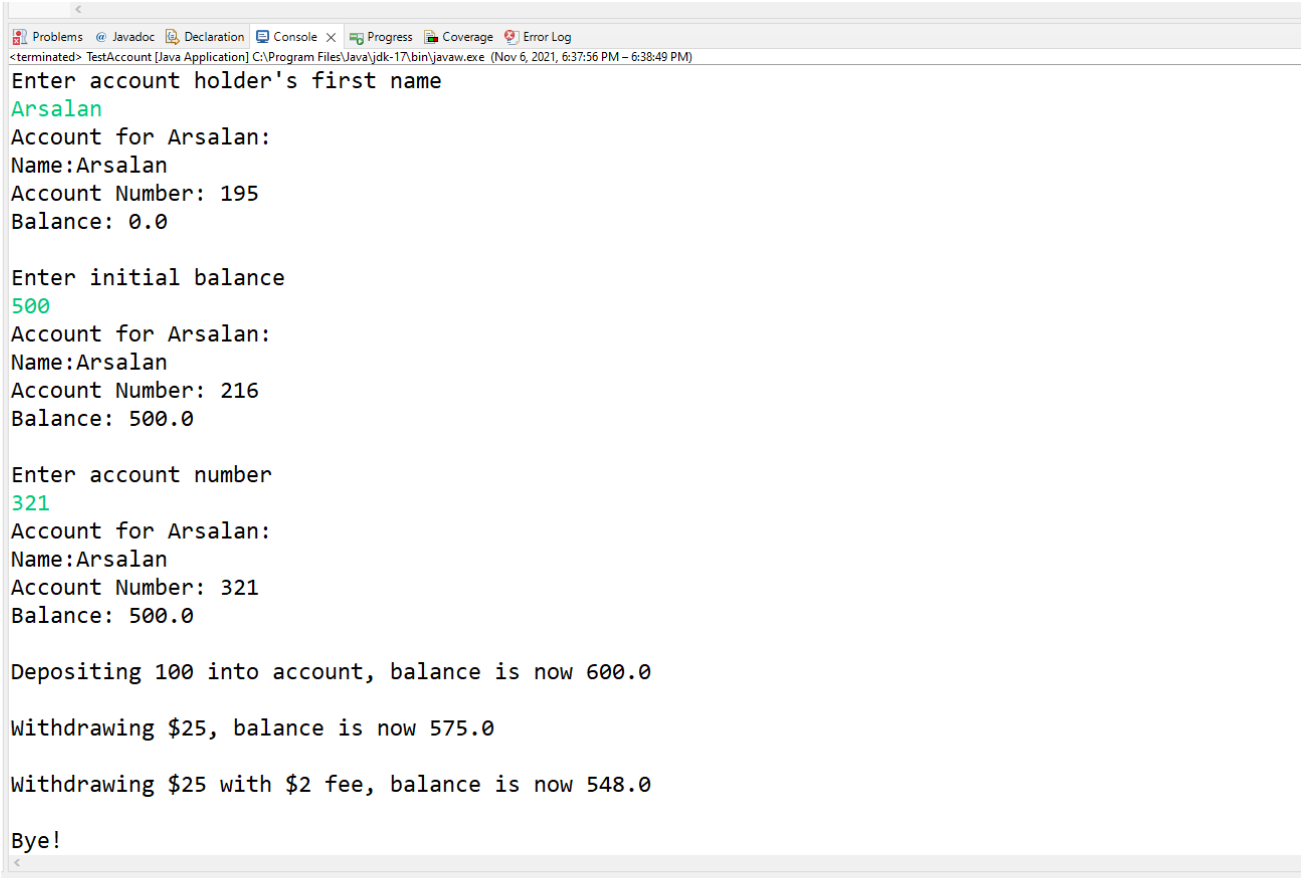
```
        acct = new Account(name);

        System.out.println("Account for " + name + ":");
        System.out.println(acct);
        System.out.println("\nEnter initial balance");

        balance = scan.nextDouble();

        acct1 = new Account(balance,name);
        System.out.println("Account for " + name + ":");
        System.out.println(acct1);
        System.out.println("\nEnter account number");

        acctNum = scan.nextLong();

        acct2 = new Account(balance,name,acctNum);
        System.out.println("Account for " + name + ":");
        System.out.println(acct2);
        System.out.print("\nDepositing 100 into account, balance is now ");

        acct2.deposit(100);
        System.out.println(acct2.getBalance());
        System.out.print("\nWithdrawing $25, balance is now ");

        acct2.withdraw(25);
        System.out.println(acct2.getBalance());
        System.out.print("\nWithdrawing $25 with $2 fee, balance is now ");

        acct2.withdraw(25,2);
```

```
<terminated> TestAccount [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Nov 6, 2021, 6:37:56 PM – 6:38:49 PM)
```



```
        System.out.println(acct);
        System.out.println("\nEnter initial balance");

        balance = scan.nextDouble();

        acct1 = new Account(balance,name);
        System.out.println("Account for " + name + ":");
        System.out.println(acct1);
        System.out.println("\nEnter account number");

        acctNum = scan.nextLong();

        acct2 = new Account(balance,name,acctNum);
        System.out.println("Account for " + name + ":");
        System.out.println(acct2);
        System.out.print("\nDepositing 100 into account, balance is now ");

        acct2.deposit(100);
        System.out.println(acct2.getBalance());
        System.out.print("\nWithdrawing $25, balance is now ");

        acct2.withdraw(25);
        System.out.println(acct2.getBalance());
        System.out.print("\nWithdrawing $25 with $2 fee, balance is now ");

        acct2.withdraw(25,2);
        System.out.println(acct2.getBalance());
        System.out.println("\nBye!");
} }
```

```
<terminated> TestAccount [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Nov 6, 2021, 6:37:56 PM – 6:38:49 PM)
```



```
<terminated> TestAccount [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (Nov 6, 2021, 6:37:56 PM – 6:38:49 PM)
Enter account holder's first name
Arsalan
Account for Arsalan:
Name:Arsalan
Account Number: 195
Balance: 0.0

Enter initial balance
500
Account for Arsalan:
Name:Arsalan
Account Number: 216
Balance: 500.0

Enter account number
321
Account for Arsalan:
Name:Arsalan
Account Number: 321
Balance: 500.0

Depositing 100 into account, balance is now 600.0

Withdrawing $25, balance is now 575.0

Withdrawing $25 with $2 fee, balance is now 548.0

Bye!
```

## PART-2

### TestAccount1

```
package lab7;
	//TestAccounts1
	// A simple program to test the numAccts method of the
	// Account class.
	import java.util.Scanner;
	public class TestAccount1 {
	public static void main(String[] args) {
	Account testAcct;
	Scanner scan = new Scanner(System.in);
	System.out.println("How many accounts would you like to create?");
	int num = scan.nextInt();
	for (int i=1; i<=num; i++) {
	testAcct = new Account(100, "Name" + i);
	System.out.println("\nCreated account " + testAcct);
	System.out.println("Now there are " + Account.numAccounts() + "
accounts");
	}
	}
	}
```

## PART-3

**Write a test program that prompts for and reads in three names and creates an account with an initial balance of $100 for each. Print the three accounts, then close the first account and try to consolidate the second and third into a new account. Now print the accounts again, including the consolidated one if it was created.**

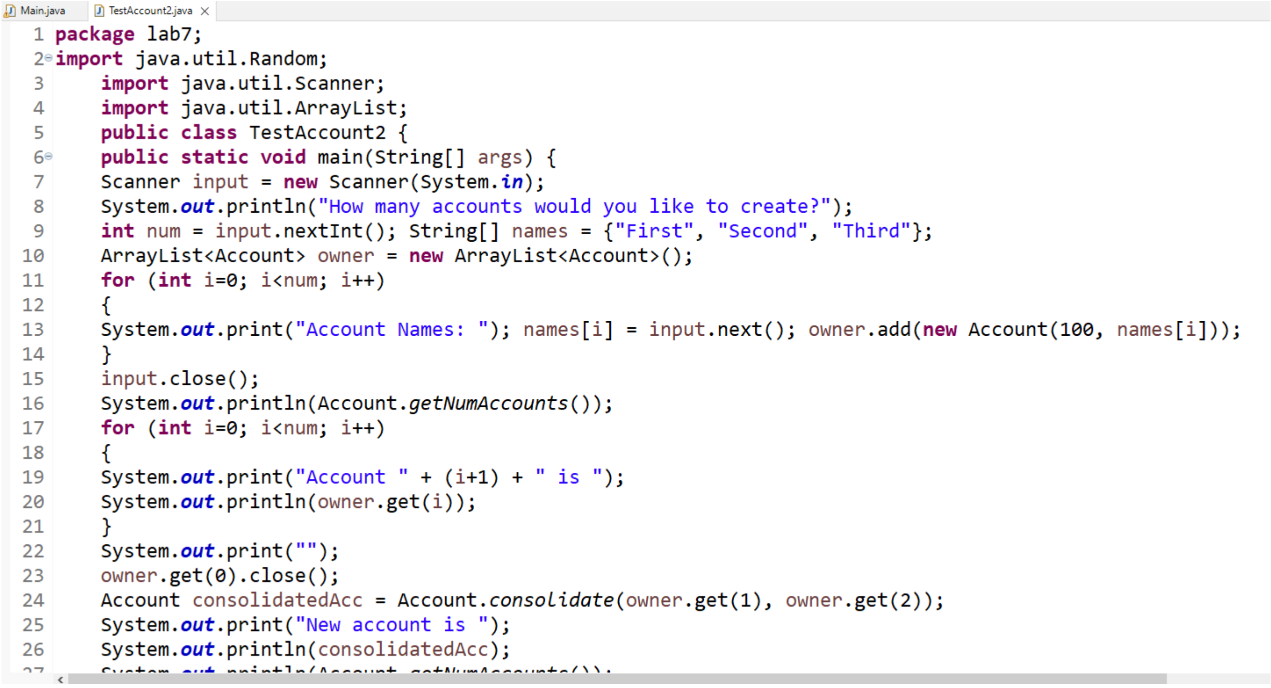
**CODE:**

```
package lab7;
import java.util.Random;
      import java.util.Scanner;
      import java.util.ArrayList;
      public class TestAccount2 {
      public static void main(String[] args) {
      Scanner input = new Scanner(System.in);
      System.out.println("How many accounts would you like to create?");
      int num = input.nextInt(); String[] names = {"First", "Second", "Third"};
      ArrayList<Account> owner = new ArrayList<Account>();
      for (int i=0; i<num; i++)
      {
      System.out.print("Account Names: "); names[i] = input.next();
owner.add(new Account(100, names[i]));
      }
      input.close();
      System.out.println(Account.getNumAccounts());
      for (int i=0; i<num; i++)
      {
      System.out.print("Account " + (i+1) + " is ");
      System.out.println(owner.get(i));
      }
      System.out.print("");
      owner.get(0).close();
      Account consolidatedAcc = Account.consolidate(owner.get(1), owner.get(2));
      System.out.print("New account is ");
      System.out.println(consolidatedAcc);
      System.out.println(Account.getNumAccounts());
      for (int i=0; i<num; i++)
      {
      System.out.print("Account " + (i+1) + " is ");
      System.out.println(owner.get(i));
      }
      System.out.print("New account is ");
      System.out.println(consolidatedAcc);
      }
      }
       class Account
      {
      private static int numAccounts;
      private double balance;
      private String Name;
      private long accountNum;
      // Overloaded Constructors
      public Account(double initBal, String owner, long number)
      {
```

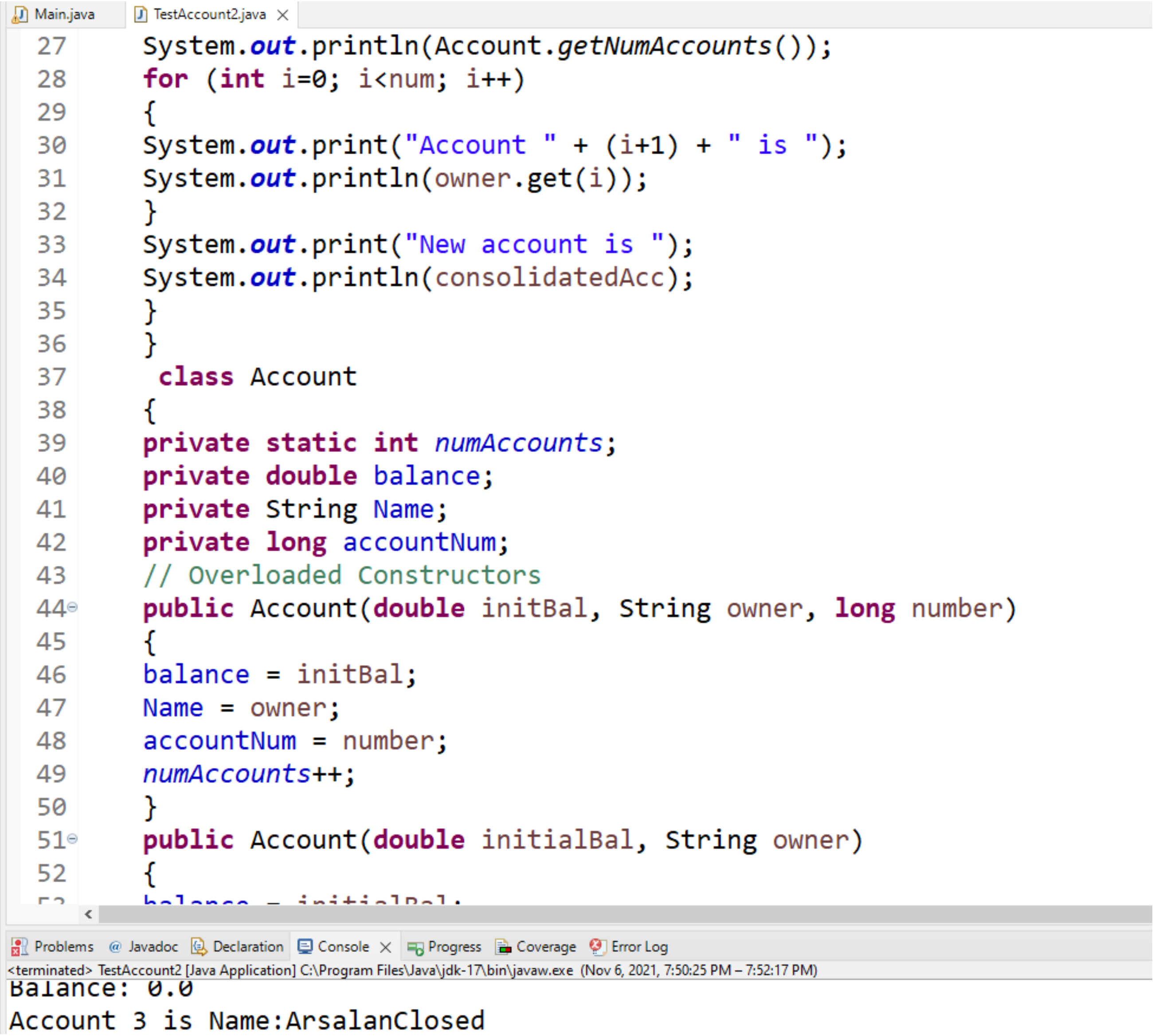
```java
balance = initBal;
Name = owner;
accountNum = number;
numAccounts++;
}
public Account(double initialBal, String owner)
{
balance = initialBal;
Name = owner;
Random rand = new Random();
accountNum = rand.nextInt(1001);
numAccounts++;
}
public Account(String owner)
{
Name = owner;
balance = 0;
Random rand = new Random();
accountNum = rand.nextInt(1001);
numAccounts++;
}
public Account()
{
balance = 0;
Name = " ";
accountNum = 0;
numAccounts++;
}
// overloaded Method to withdraw Money
public void withdraw(double amount)
{
if (balance >= amount)
balance -= amount;
else
System.out.println("Insufficient funds");
}
public void withdraw(double amount, double fee)
{
if (amount <= 0 || fee<=0)
System.out.println("Error! Amount or fee can not be less than zero.");
else
{
double totalAmount = amount + fee;
if (balance >= totalAmount)
balance -= totalAmount;
else
System.out.println("Insufficient funds");
}
}
//Method to Deposit Money
public void deposit(double amount)
{
balance += amount;
}
```

```
// Method to check balance
public double getBalance()
{
return balance;
}
// Method to check number of accounts
public static int getNumAccounts()
{
System.out.print("No. of Accounts created: ");
return numAccounts;
}
// Method to close account
public void close()
{
Name += "Closed";
balance = 0;
numAccounts--;
}
// Method to Consolidate Account
public static Account consolidate(Account acct1, Account acct2)
{
if (acct1.Name.equals(acct2.Name) && acct1.accountNum != acct2.accountNum)
{
Account acct3 = new Account();
acct3.Name = acct1.Name;
acct3.balance = acct1.balance + acct2.balance;
acct3.accountNum = acct2.accountNum + 1;
acct1.close();
acct2.close();
return acct3;
}
else
{
System.out.println("Error! For consolidation Account owner must be the same and must have two different accounts.");
return null;
}
}
public String toString()
{
return "Name:" + Name +
"\nAccount Number: " + accountNum +
"\nBalance: " + balance;
}
public static String numAccounts() {

return null;
}
```

```
    }
```

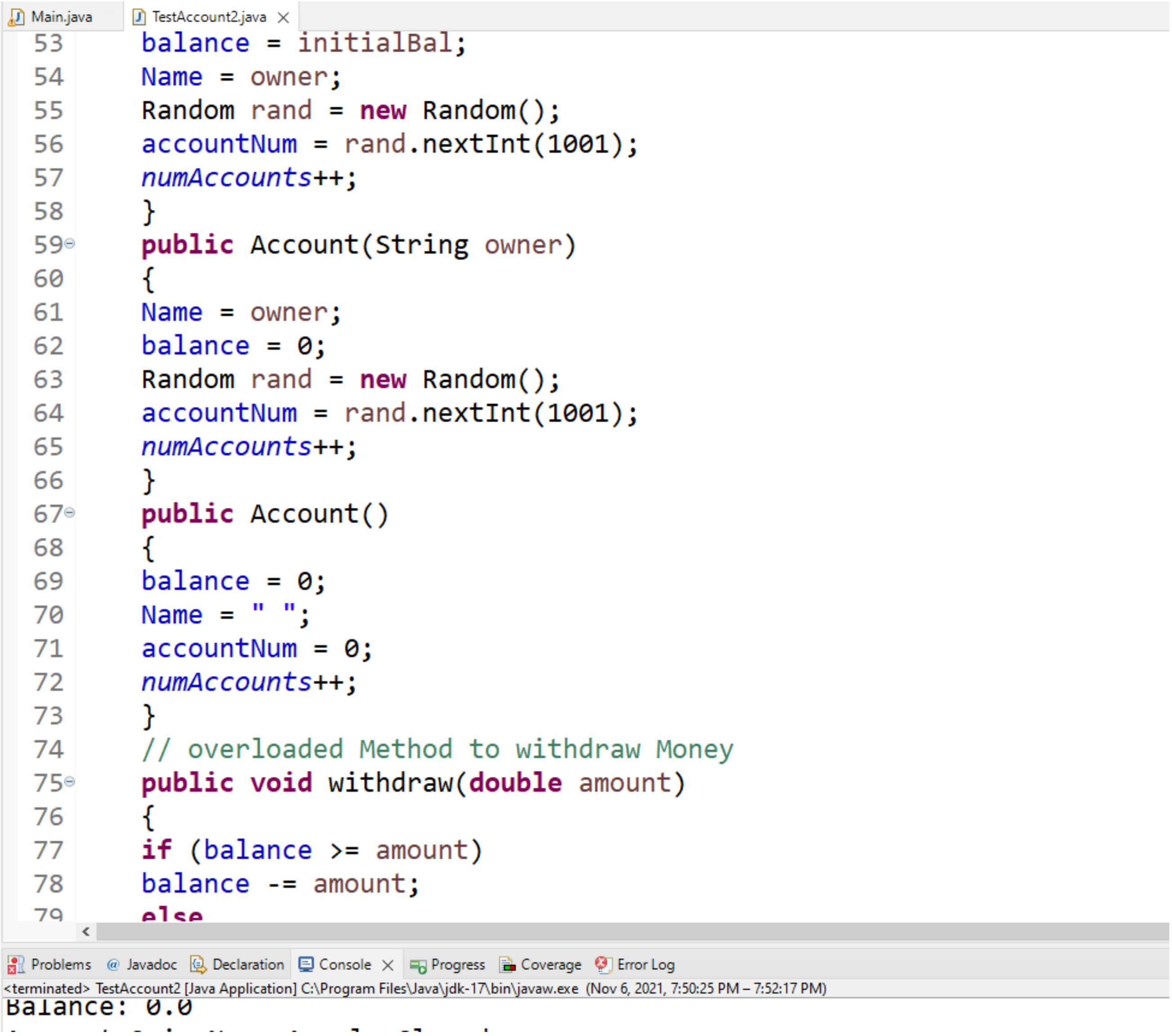


```java
package lab7;
import java.util.Random;
    import java.util.Scanner;
    import java.util.ArrayList;
    public class TestAccount2 {
    public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    System.out.println("How many accounts would you like to create?");
    int num = input.nextInt(); String[] names = {"First", "Second", "Third"};
    ArrayList<Account> owner = new ArrayList<Account>();
    for (int i=0; i<num; i++)
    {
    System.out.print("Account Names: "); names[i] = input.next(); owner.add(new Account(100, names[i]));
    }
    input.close();
    System.out.println(Account.getNumAccounts());
    for (int i=0; i<num; i++)
    {
    System.out.print("Account " + (i+1) + " is ");
    System.out.println(owner.get(i));
    }
    System.out.print("");
    owner.get(0).close();
    Account consolidatedAcc = Account.consolidate(owner.get(1), owner.get(2));
    System.out.print("New account is ");
    System.out.println(consolidatedAcc);
    System.out.println(Account.getNumAccounts());
```



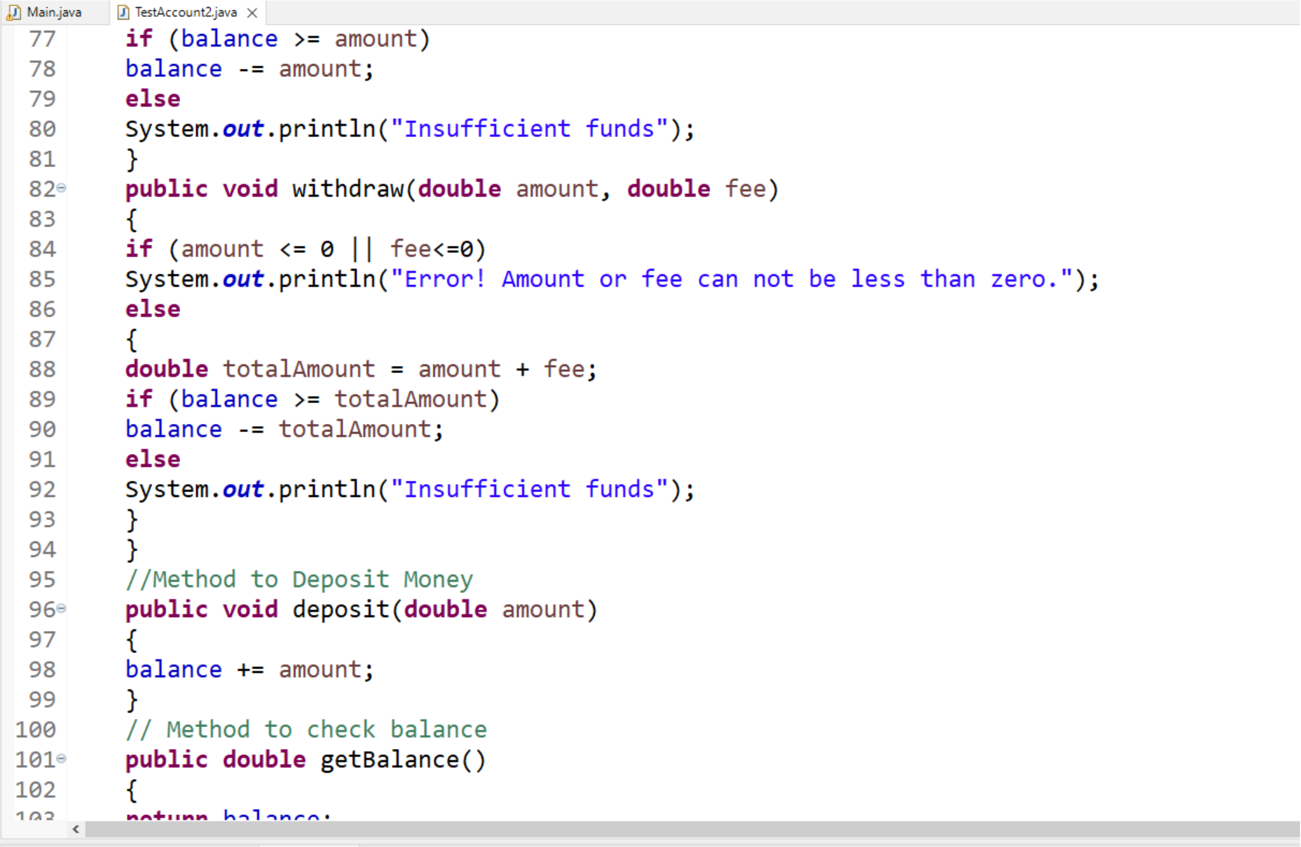
```java
System.out.println(Account.getNumAccounts());
for (int i=0; i<num; i++)
{
System.out.print("Account " + (i+1) + " is ");
System.out.println(owner.get(i));
}
System.out.print("New account is ");
System.out.println(consolidatedAcc);
}
}
 class Account
{
private static int numAccounts;
private double balance;
private String Name;
private long accountNum;
// Overloaded Constructors
public Account(double initBal, String owner, long number)
{
balance = initBal;
Name = owner;
accountNum = number;
numAccounts++;
}
public Account(double initialBal, String owner)
{
balance = initialBal;
```

```
<terminated> TestAccount2 [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (Nov 6, 2021, 7:50:25 PM – 7:52:17 PM)
Balance: 0.0
Account 3 is Name:ArsalanClosed
```

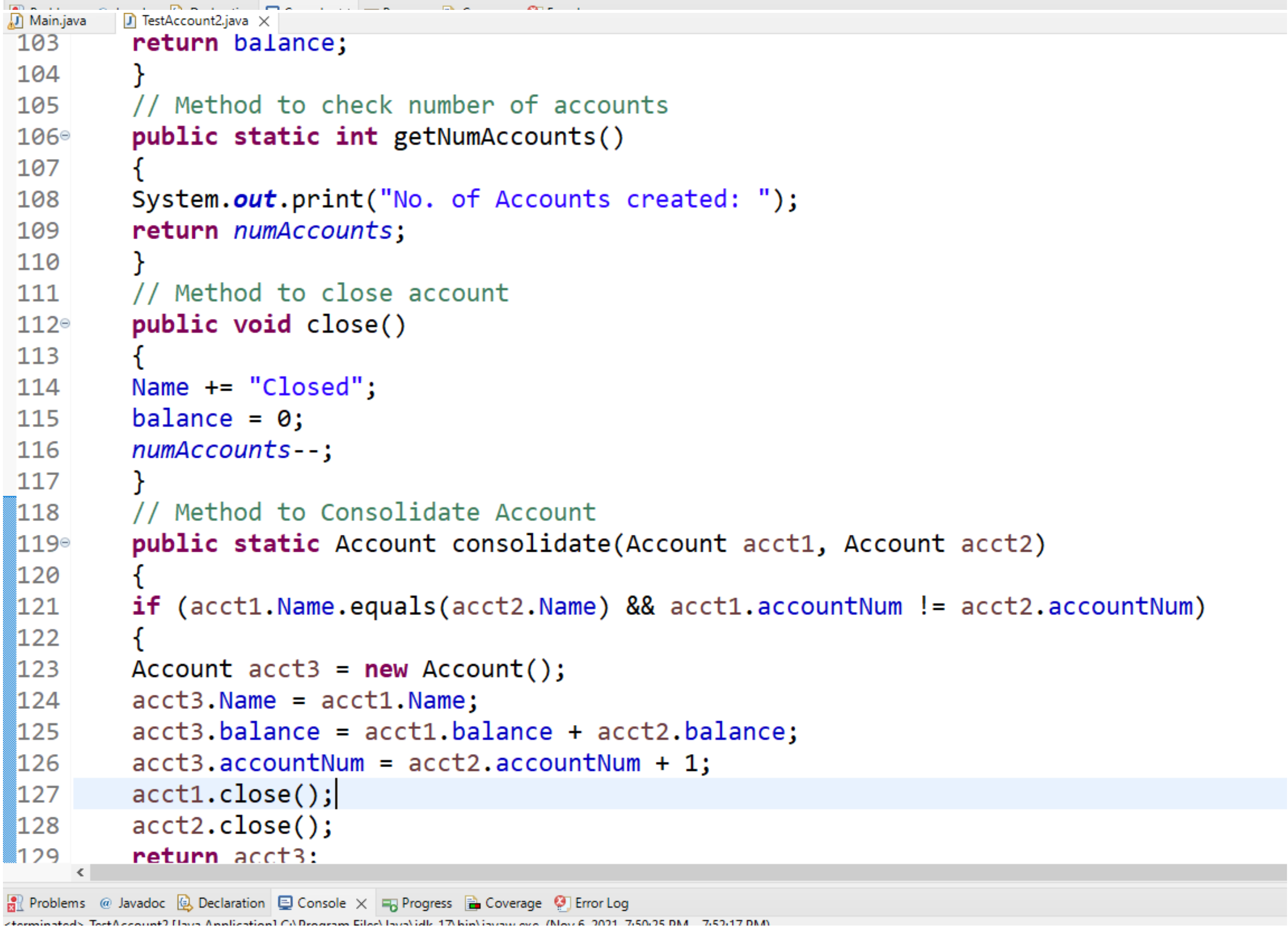


```
balance = initialBal;
Name = owner;
Random rand = new Random();
accountNum = rand.nextInt(1001);
numAccounts++;
}
public Account(String owner)
{
Name = owner;
balance = 0;
Random rand = new Random();
accountNum = rand.nextInt(1001);
numAccounts++;
}
public Account()
{
balance = 0;
Name = " ";
accountNum = 0;
numAccounts++;
}
// overloaded Method to withdraw Money
public void withdraw(double amount)
{
if (balance >= amount)
balance -= amount;
else
```

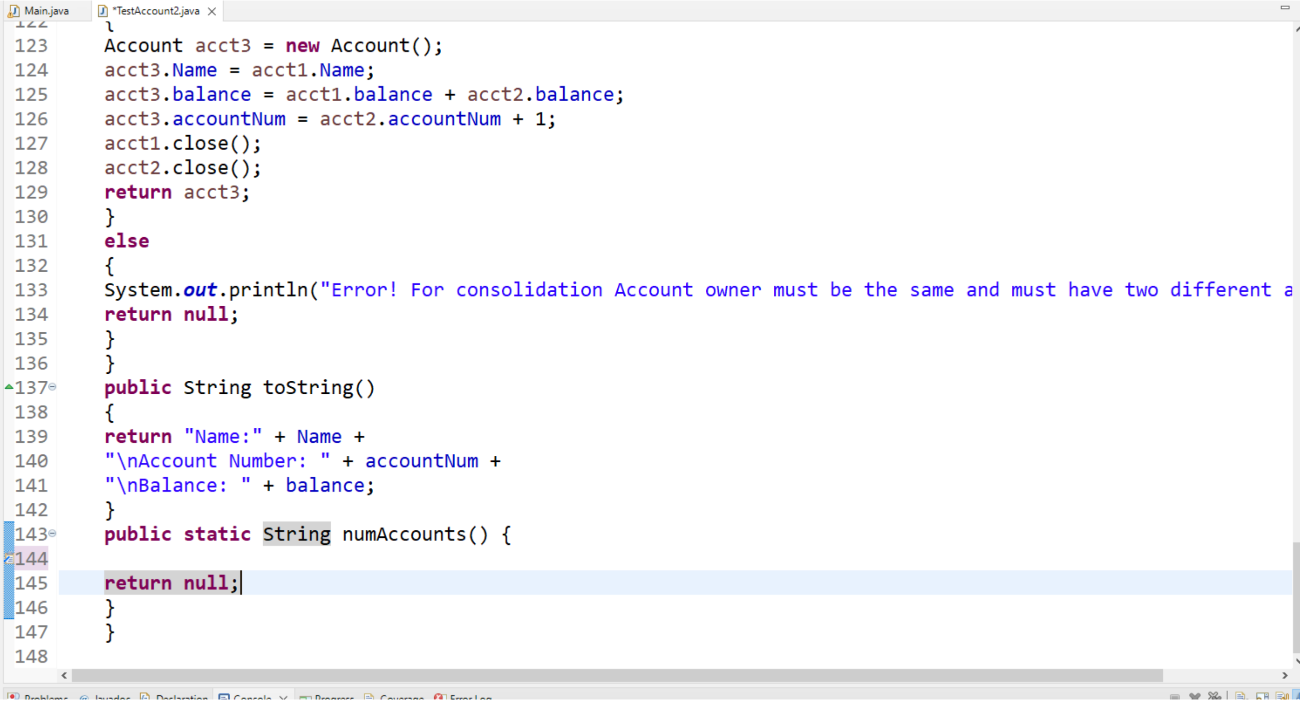
```
<terminated> TestAccount2 [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (Nov 6, 2021, 7:50:25 PM – 7:52:17 PM)
Balance: 0.0
```



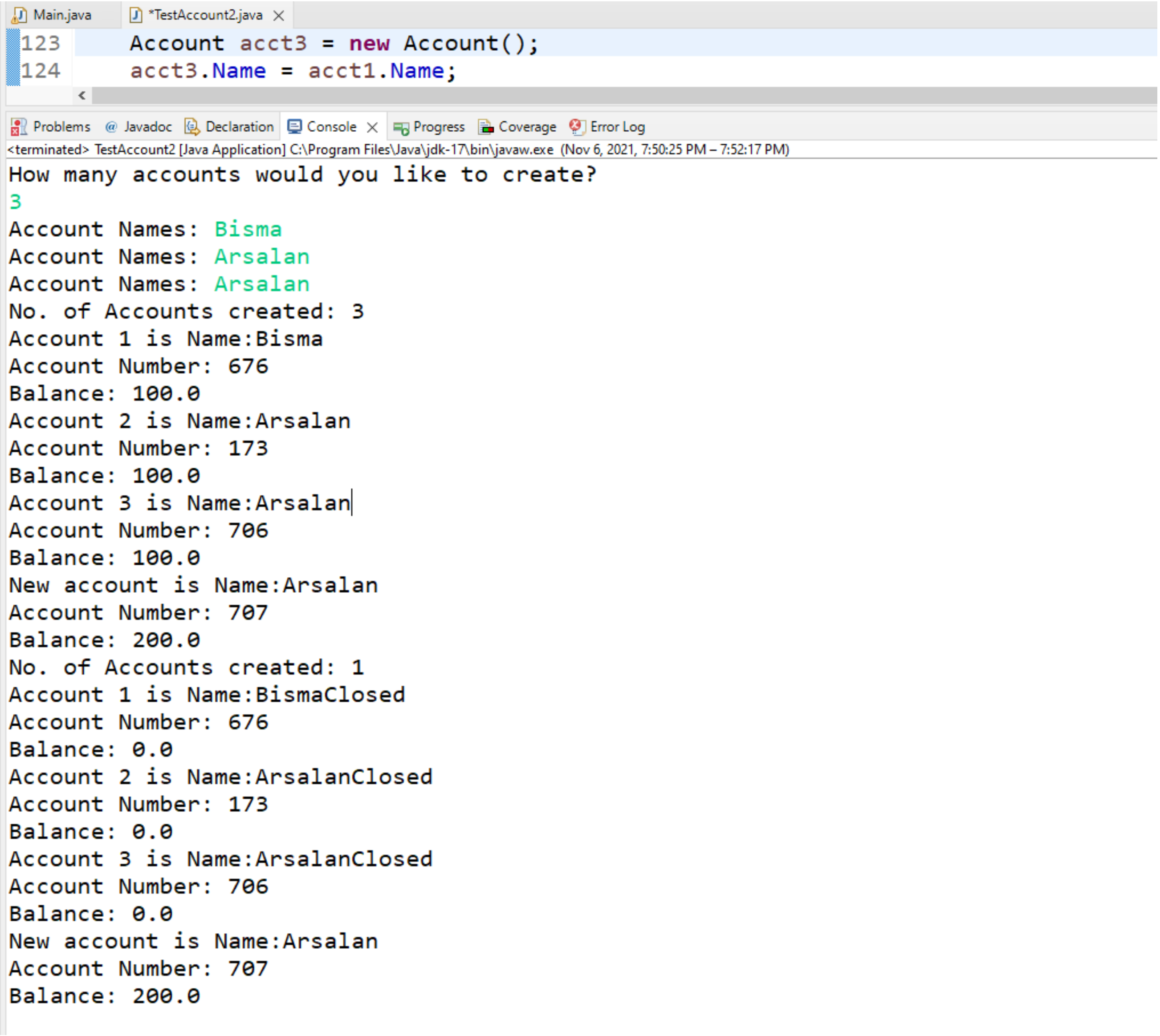
```
if (balance >= amount)
balance -= amount;
else
System.out.println("Insufficient funds");
}
public void withdraw(double amount, double fee)
{
if (amount <= 0 || fee<=0)
System.out.println("Error! Amount or fee can not be less than zero.");
else
{
double totalAmount = amount + fee;
if (balance >= totalAmount)
balance -= totalAmount;
else
System.out.println("Insufficient funds");
}
}
//Method to Deposit Money
public void deposit(double amount)
{
balance += amount;
}
// Method to check balance
public double getBalance()
{
```



```
return balance;
}
// Method to check number of accounts
public static int getNumAccounts()
{
System.out.print("No. of Accounts created: ");
return numAccounts;
}
// Method to close account
public void close()
{
Name += "Closed";
balance = 0;
numAccounts--;
}
// Method to Consolidate Account
public static Account consolidate(Account acct1, Account acct2)
{
if (acct1.Name.equals(acct2.Name) && acct1.accountNum != acct2.accountNum)
{
Account acct3 = new Account();
acct3.Name = acct1.Name;
acct3.balance = acct1.balance + acct2.balance;
acct3.accountNum = acct2.accountNum + 1;
acct1.close();
acct2.close();
return acct3;
```



```
{
Account acct3 = new Account();
acct3.Name = acct1.Name;
acct3.balance = acct1.balance + acct2.balance;
acct3.accountNum = acct2.accountNum + 1;
acct1.close();
acct2.close();
return acct3;
}
else
{
System.out.println("Error! For consolidation Account owner must be the same and must have two different a
return null;
}
}
public String toString()
{
return "Name:" + Name +
"\nAccount Number: " + accountNum +
"\nBalance: " + balance;
}
public static String numAccounts() {

return null;
}
}
```



```
    Account acct3 = new Account();
    acct3.Name = acct1.Name;
```

```
<terminated> TestAccount2 [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (Nov 6, 2021, 7:50:25 PM – 7:52:17 PM)
How many accounts would you like to create?
3
Account Names: Bisma
Account Names: Arsalan
Account Names: Arsalan
No. of Accounts created: 3
Account 1 is Name:Bisma
Account Number: 676
Balance: 100.0
Account 2 is Name:Arsalan
Account Number: 173
Balance: 100.0
Account 3 is Name:Arsalan
Account Number: 706
Balance: 100.0
New account is Name:Arsalan
Account Number: 707
Balance: 200.0
No. of Accounts created: 1
Account 1 is Name:BismaClosed
Account Number: 676
Balance: 0.0
Account 2 is Name:ArsalanClosed
Account Number: 173
Balance: 0.0
Account 3 is Name:ArsalanClosed
Account Number: 706
Balance: 0.0
New account is Name:Arsalan
Account Number: 707
Balance: 200.0
```