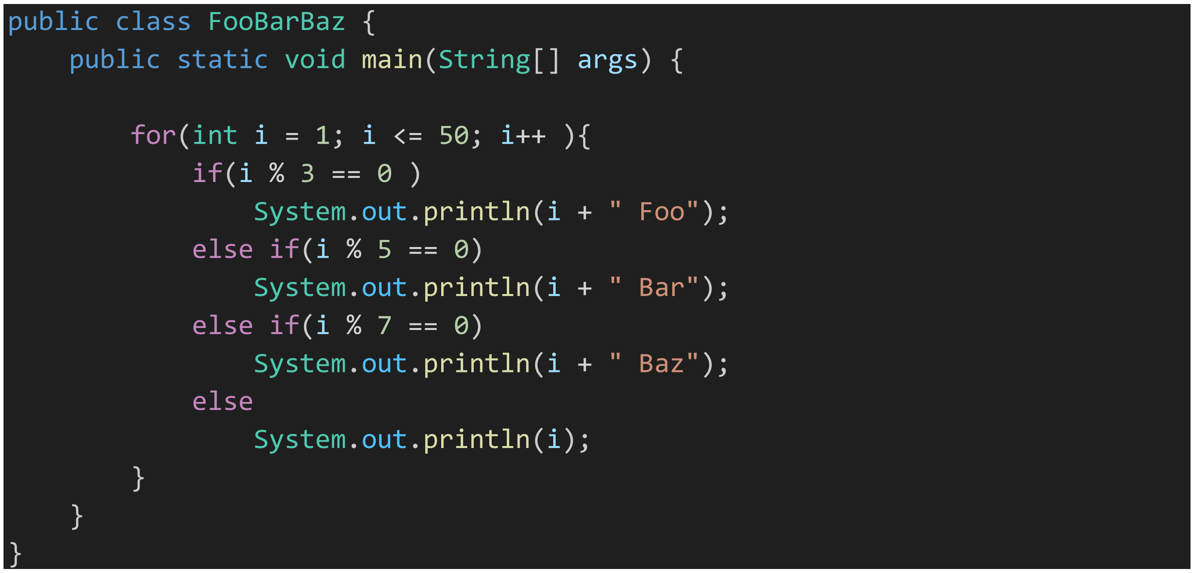
LAB 2 Expressions andFlowControl

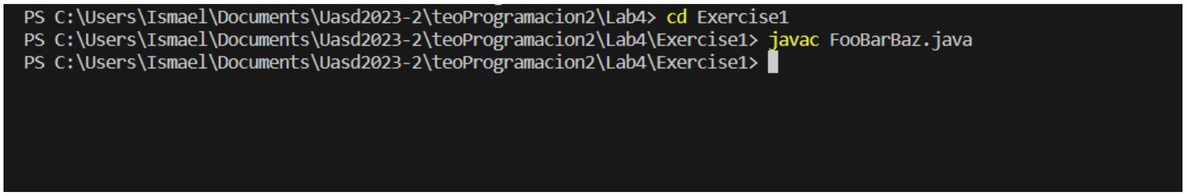
Exercise 1: Using Loops and Branching Statements

Task 1 – Creating the FooBarBaz Class



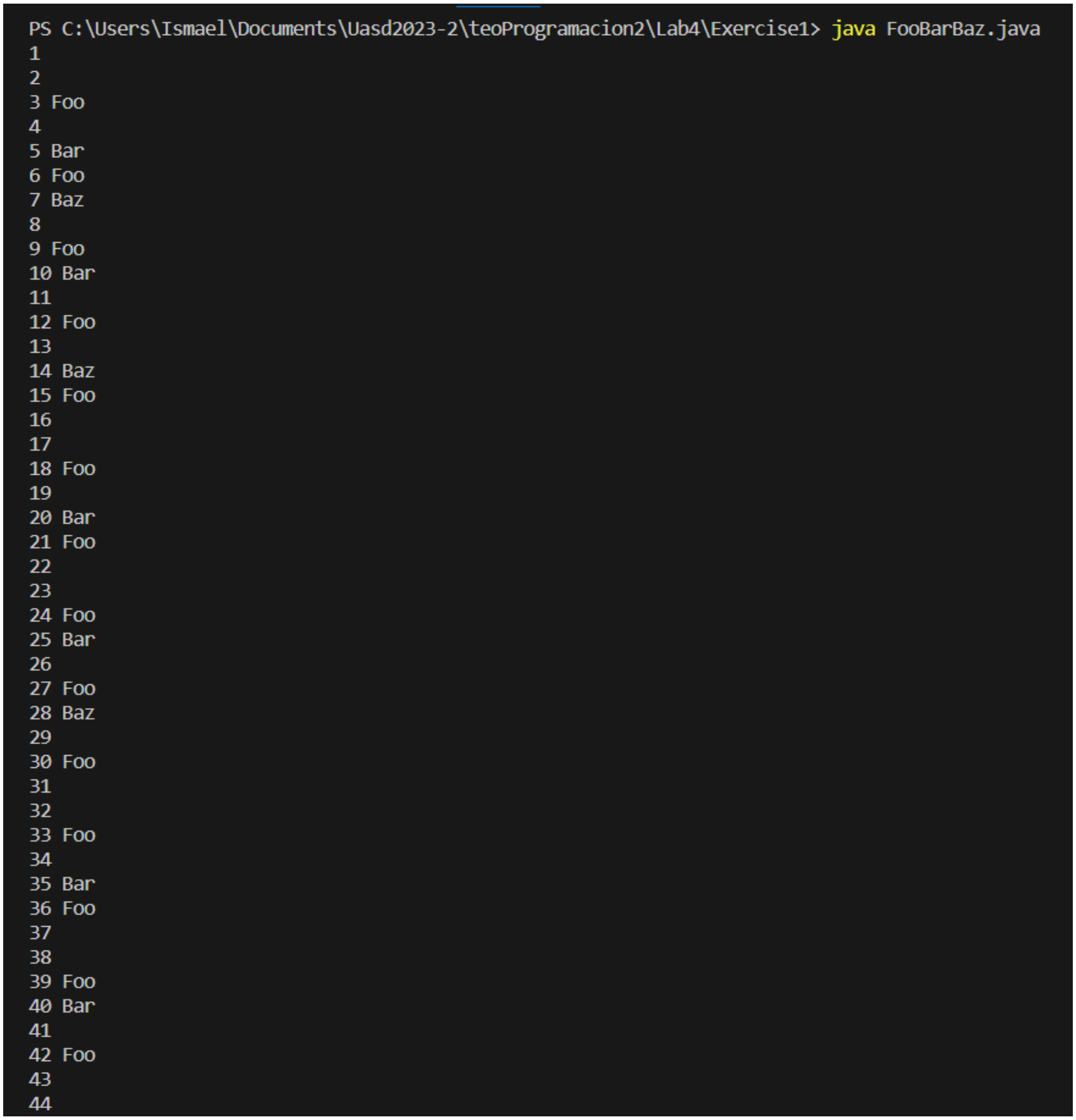
```java
public class FooBarBaz {
    public static void main(String[] args) {

        for(int i = 1; i <= 50; i++ ){
            if(i % 3 == 0 )
                System.out.println(i + " Foo");
            else if(i % 5 == 0)
                System.out.println(i + " Bar");
            else if(i % 7 == 0)
                System.out.println(i + " Baz");
            else
                System.out.println(i);
        }
    }
}
```

Task 2 – Compiling the FooBarBaz Class



```
PS C:\Users\Ismael\Documents\Uasd2023-2\teoProgramacion2\Lab4> cd Exercise1
PS C:\Users\Ismael\Documents\Uasd2023-2\teoProgramacion2\Lab4\Exercise1> javac FooBarBaz.java
PS C:\Users\Ismael\Documents\Uasd2023-2\teoProgramacion2\Lab4\Exercise1> 
```

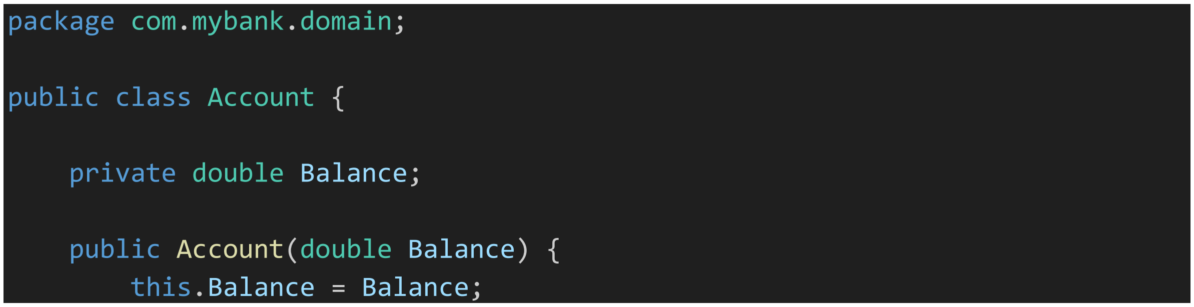
Task 3 – Running the FooBarBaz Program



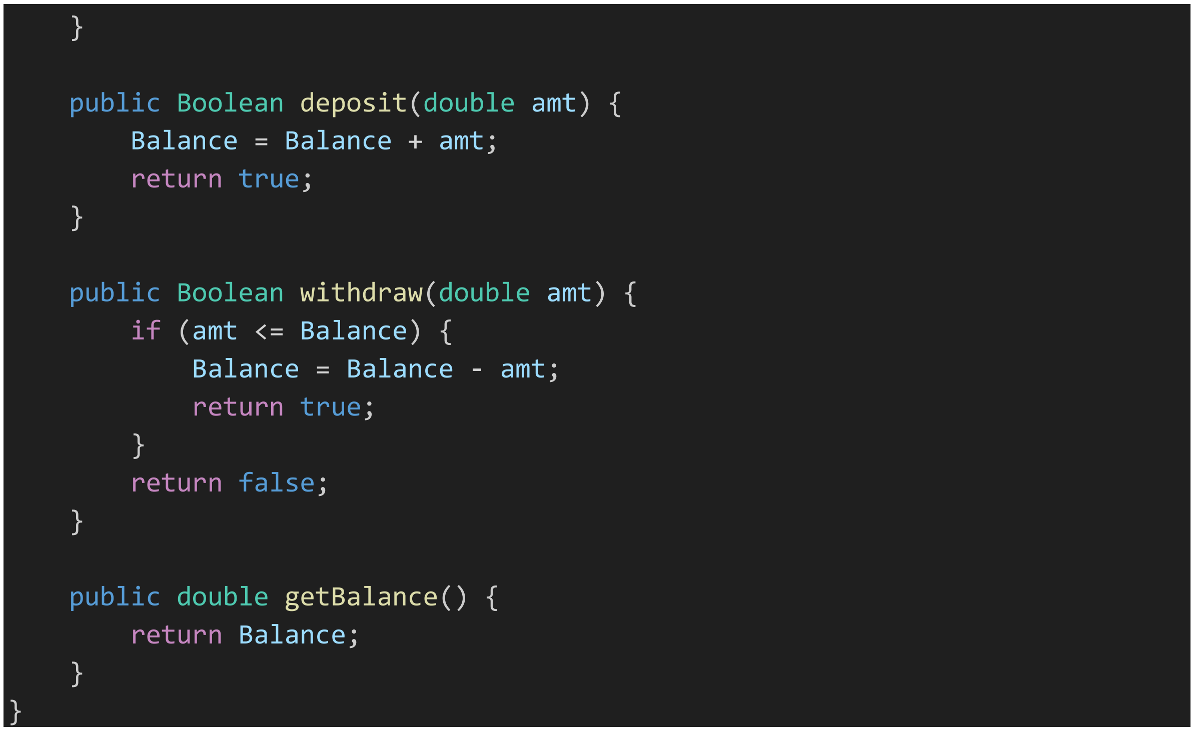
```
PS C:\Users\Ismael\Documents\Uasd2023-2\teoProgramacion2\Lab4\Exercise1> java FooBarBaz.java
1
2
3 Foo
4
5 Bar
6 Foo
7 Baz
8
9 Foo
10 Bar
11
12 Foo
13
14 Baz
15 Foo
16
17
18 Foo
19
20 Bar
21 Foo
22
23
24 Foo
25 Bar
26
27 Foo
28 Baz
29
30 Foo
31
32
33 Foo
34
35 Bar
36 Foo
37
38
39 Foo
40 Bar
41
42 Foo
43
44
```

Exercise 2: Using Conditional Statements in the Account Class

Task 1 – Modifying the Account Class

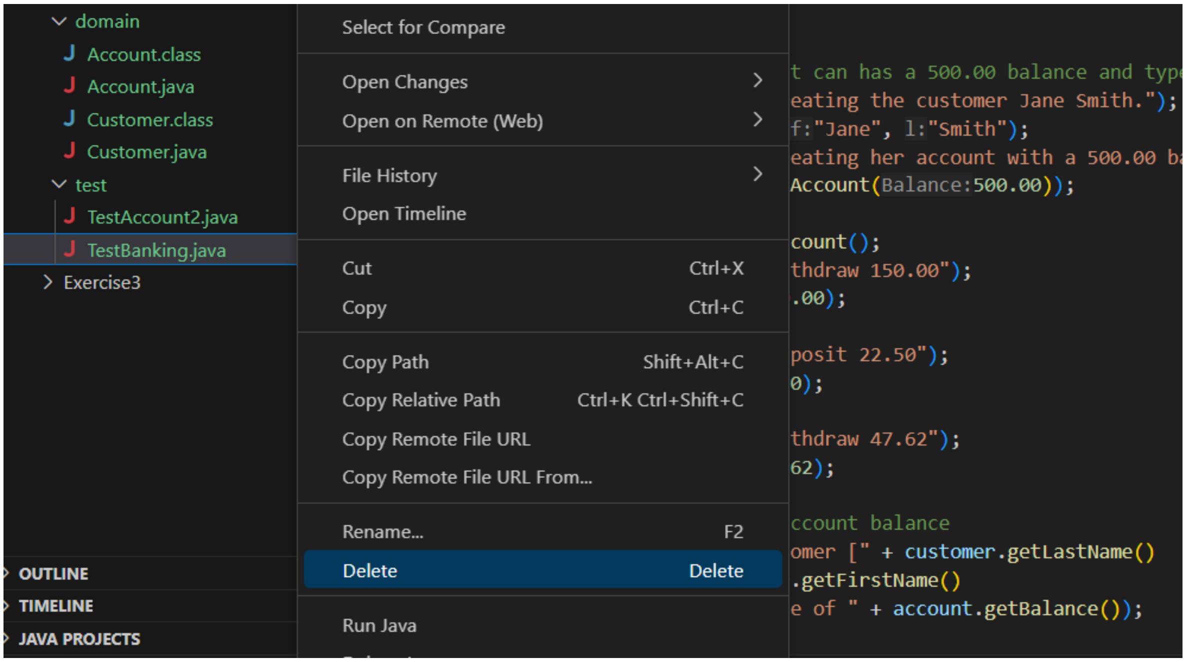


```
package com.mybank.domain;

public class Account {

    private double Balance;

    public Account(double Balance) {
        this.Balance = Balance;
```



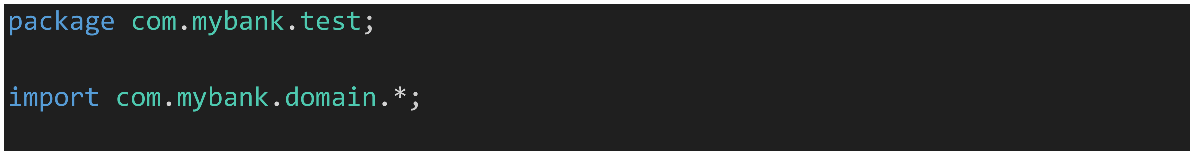
```
    }

    public Boolean deposit(double amt) {
        Balance = Balance + amt;
        return true;
    }

    public Boolean withdraw(double amt) {
        if (amt <= Balance) {
            Balance = Balance - amt;
            return true;
        }
        return false;
    }

    public double getBalance() {
        return Balance;
    }
}
```

Task 2 – Deleting the Current TestBanking Class

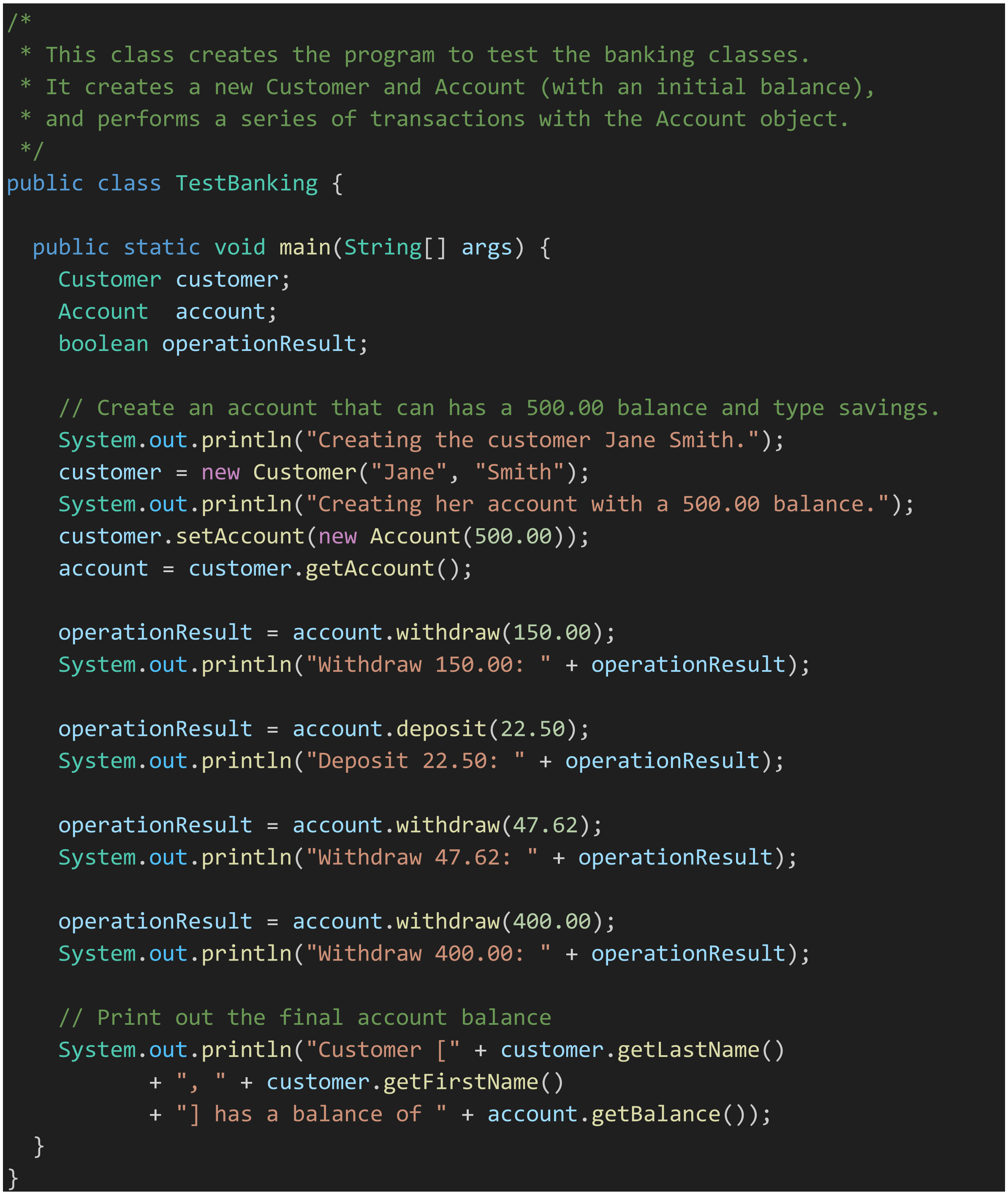


```
t can has a 500.00 balance and typ
eating the customer Jane Smith.");
f:"Jane", l:"Smith");
eating her account with a 500.00 ba
Account(Balance:500.00));

count();
thdraw 150.00");
.00);

posit 22.50");
0);

thdraw 47.62");
62);

ccount balance
omer [" + customer.getLastName()
.getFirstName()
e of " + account.getBalance());
```

Task 3 – Copying the TestBanking Class

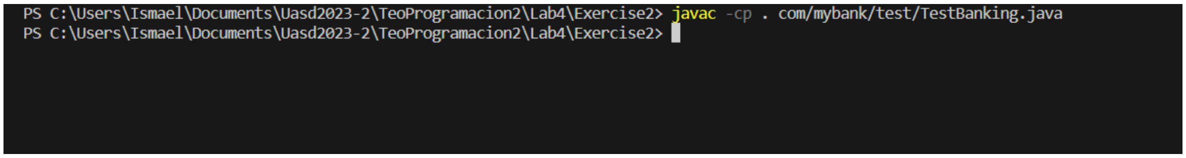


```
package com.mybank.test;

import com.mybank.domain.*;
```



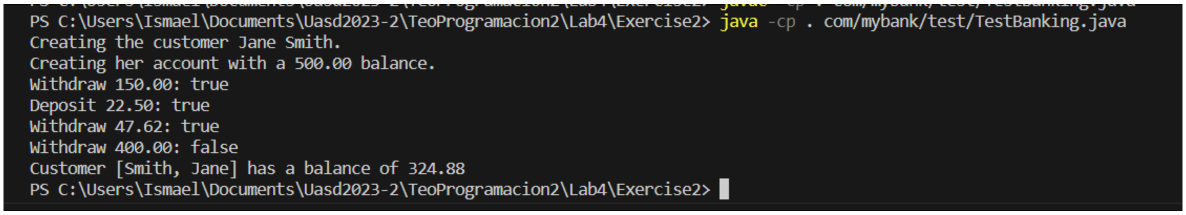
```java
/*
 * This class creates the program to test the banking classes.
 * It creates a new Customer and Account (with an initial balance),
 * and performs a series of transactions with the Account object.
 */
public class TestBanking {

  public static void main(String[] args) {
    Customer customer;
    Account  account;
    boolean operationResult;

    // Create an account that can has a 500.00 balance and type savings.
    System.out.println("Creating the customer Jane Smith.");
    customer = new Customer("Jane", "Smith");
    System.out.println("Creating her account with a 500.00 balance.");
    customer.setAccount(new Account(500.00));
    account = customer.getAccount();

    operationResult = account.withdraw(150.00);
    System.out.println("Withdraw 150.00: " + operationResult);

    operationResult = account.deposit(22.50);
    System.out.println("Deposit 22.50: " + operationResult);

    operationResult = account.withdraw(47.62);
    System.out.println("Withdraw 47.62: " + operationResult);

    operationResult = account.withdraw(400.00);
    System.out.println("Withdraw 400.00: " + operationResult);

    // Print out the final account balance
    System.out.println("Customer [" + customer.getLastName()
           + ", " + customer.getFirstName()
           + "] has a balance of " + account.getBalance());
  }
}
```

Task 4 – Compiling the TestBanking Class



```
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab4\Exercise2> javac -cp . com/mybank/test/TestBanking.java
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab4\Exercise2> 
```

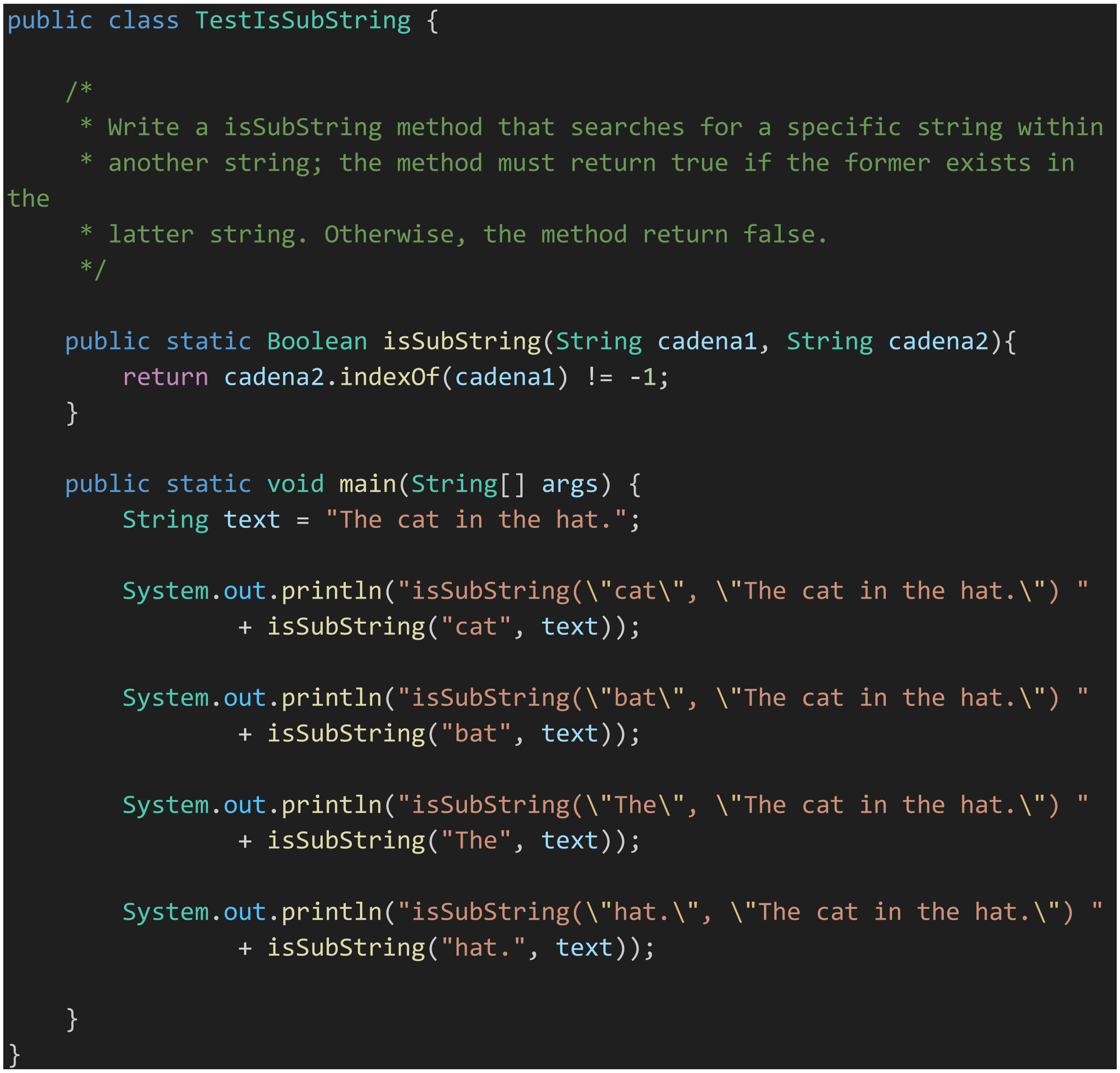
Task 5 – Running the TestBanking Program



```
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab4\Exercise2> java -cp . com/mybank/test/TestBanking.java
Creating the customer Jane Smith.
Creating her account with a 500.00 balance.
Withdraw 150.00: true
Deposit 22.50: true
Withdraw 47.62: true
Withdraw 400.00: false
Customer [Smith, Jane] has a balance of 324.88
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab4\Exercise2> 
```

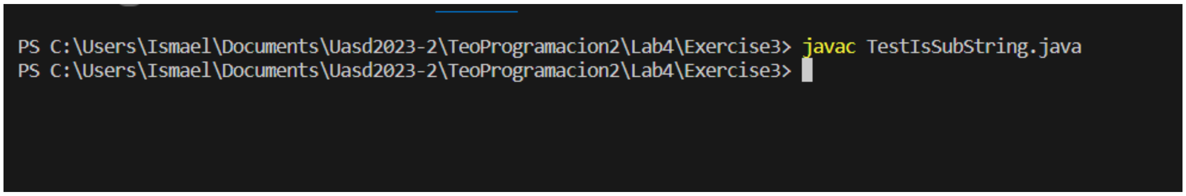
Exercise 3: Using Nested Loops

Task 1 – Writing the isSubString Method



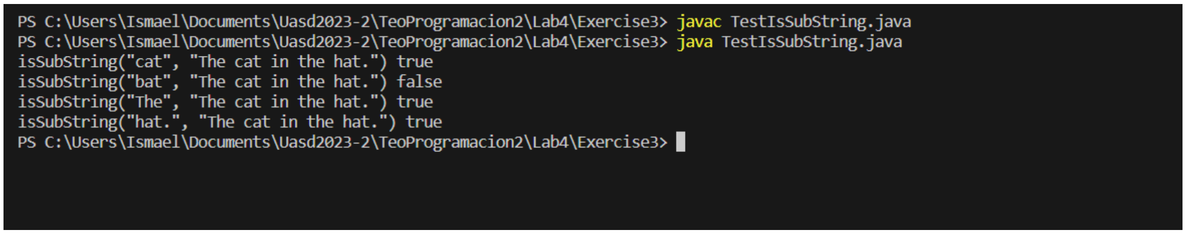
```java
public class TestIsSubString {

    /*
     * Write a isSubString method that searches for a specific string within
     * another string; the method must return true if the former exists in
the
     * latter string. Otherwise, the method return false.
     */

    public static Boolean isSubString(String cadena1, String cadena2){
        return cadena2.indexOf(cadena1) != -1;
    }

    public static void main(String[] args) {
        String text = "The cat in the hat.";

        System.out.println("isSubString(\"cat\", \"The cat in the hat.\") "
                + isSubString("cat", text));

        System.out.println("isSubString(\"bat\", \"The cat in the hat.\") "
                + isSubString("bat", text));

        System.out.println("isSubString(\"The\", \"The cat in the hat.\") "
                + isSubString("The", text));

        System.out.println("isSubString(\"hat.\", \"The cat in the hat.\") "
                + isSubString("hat.", text));

    }
}
```

Task 2 – Compiling the TestIsSubString Class



```
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab4\Exercise3> javac TestIsSubString.java
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab4\Exercise3>
```

Task 3 – Running the TestIsSubString Program



```
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab4\Exercise3> javac TestIsSubString.java
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab4\Exercise3> java TestIsSubString.java
isSubString("cat", "The cat in the hat.") true
isSubString("bat", "The cat in the hat.") false
isSubString("The", "The cat in the hat.") true
isSubString("hat.", "The cat in the hat.") true
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab4\Exercise3>
```