King Abdulaziz University
Faculty of Engineering
Electrical and Computer Engineering
EE-202-Object-Oriented Computer Programming

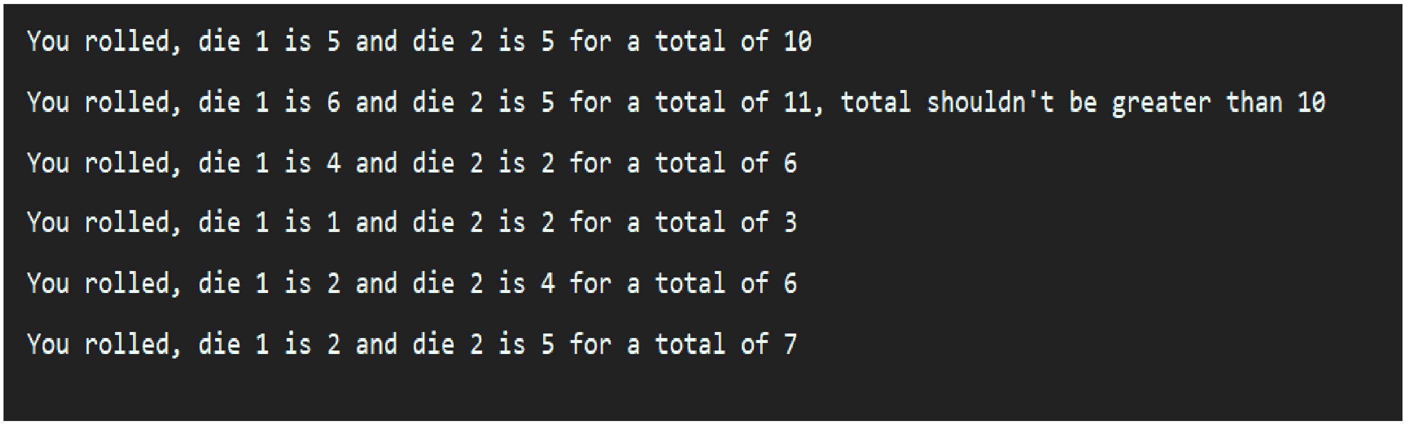
Lab-8

Q-1 Write a Java program that simulates the rolling of a pair of dice (**five-times**). For each die in the pair, the program should generate a random number between 1 and 6 (inclusive). It should print out the result of the roll for each die and the total roll (the sum of the two dice). If the sum is greater than 10, do not count that roll among the five rolls.

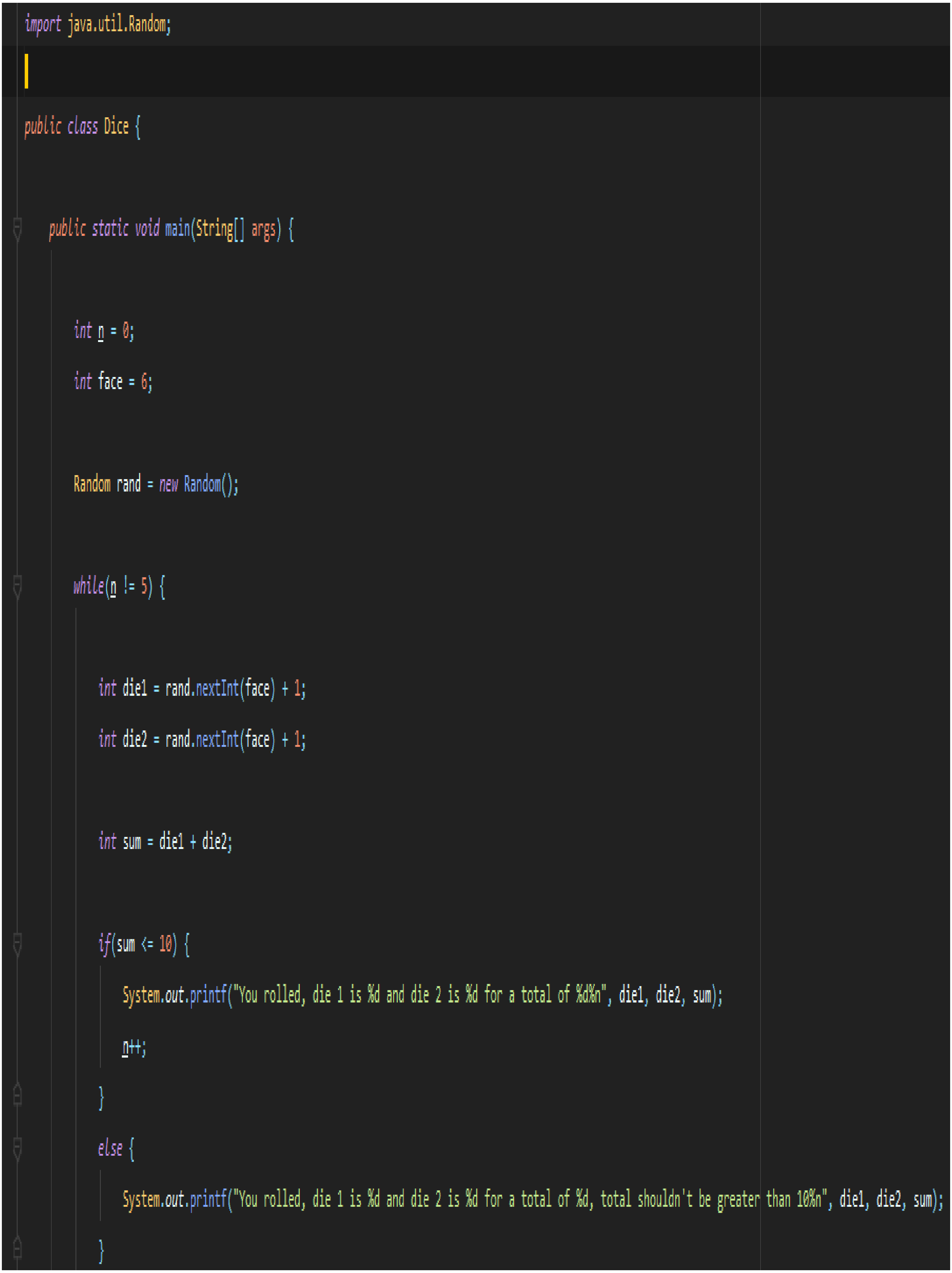
Sample Output

```
You rolled, die1-6 and die2-5 for a total of 11, Total is > 10
You rolled, die1-4 and die2-5 for a total of 9
You rolled, die1-6 and die2-5 for a total of 11, Total is > 10
You rolled, die1-3 and die2-5 for a total of 8
You rolled, die1-5 and die2-6 for a total of 11, Total is > 10
You rolled, die1-2 and die2-4 for a total of 6
You rolled, die1-3 and die2-4 for a total of 7
You rolled, die1-2 and die2-1 for a total of 3
```

Answer:



```
You rolled, die 1 is 5 and die 2 is 5 for a total of 10

You rolled, die 1 is 6 and die 2 is 5 for a total of 11, total shouldn't be greater than 10

You rolled, die 1 is 4 and die 2 is 2 for a total of 6

You rolled, die 1 is 1 and die 2 is 2 for a total of 3

You rolled, die 1 is 2 and die 2 is 4 for a total of 6

You rolled, die 1 is 2 and die 2 is 5 for a total of 7
```



```
import java.util.Random;


public class Dice {

    public static void main(String[] args) {

        int n = 0;
        int face = 6;

        Random rand = new Random();

        while(n != 5) {

            int die1 = rand.nextInt(face) + 1;
            int die2 = rand.nextInt(face) + 1;

            int sum = die1 + die2;

            if(sum <= 10) {
                System.out.printf("You rolled, die 1 is %d and die 2 is %d for a total of %d%n", die1, die2, sum);
                n++;
            }
            else {
                System.out.printf("You rolled, die 1 is %d and die 2 is %d for a total of %d, total shouldn't be greater than 10%n", die1, die2, sum);
            }
```

Q-2 You are given two classes (Student and Grades)

Modify class Student by adding:

- setGrade1() and setGrade2() methods
- static getAverage() method, to find the average of the two grades
- static compare() method, that returns true if the average is above 50, and false otherwise

Modify class Grades by:

- setting Mary’s grades to 55 and 60
- print the average of Mary and check its value using compare( ) method
- setting Mike’s grades to 5 and 6
- print the average of Mike and check its value using compare( ) method
- call the toString( ) method for both students

There are two method calls in class Grades,

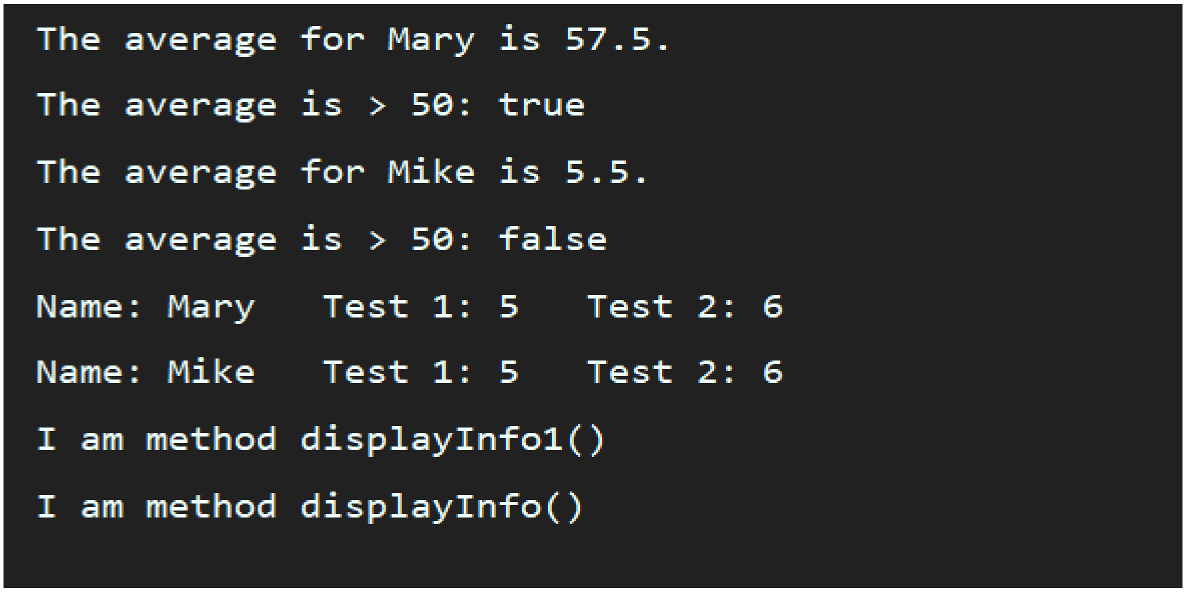
- displayInfo1()
- Student.displayInfo()

Add these static methods into their appropriate class. **Do not forget** to add their print statements

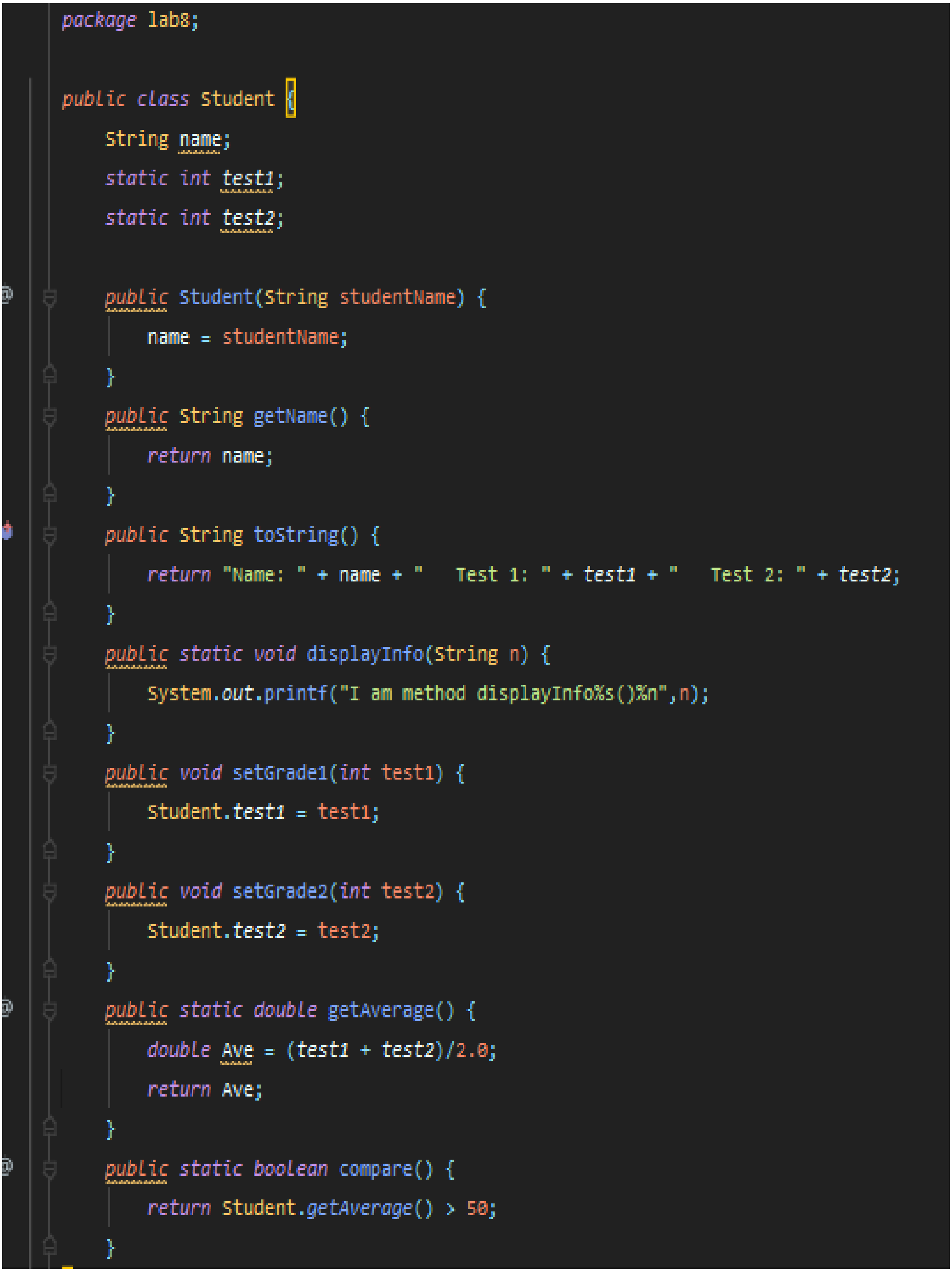
Sample Output

```
The average for Mary is 55.0.
The average is > 50: true

The average for Mike is 5.5.
The average is > 50: false

Student 1: Name: Mary   Test 1: 5   Test 2: 6
Student 2: Name: Mike   Test 1: 5   Test 2: 6

I am method displayInfo1()
I am method displayInfo()
```

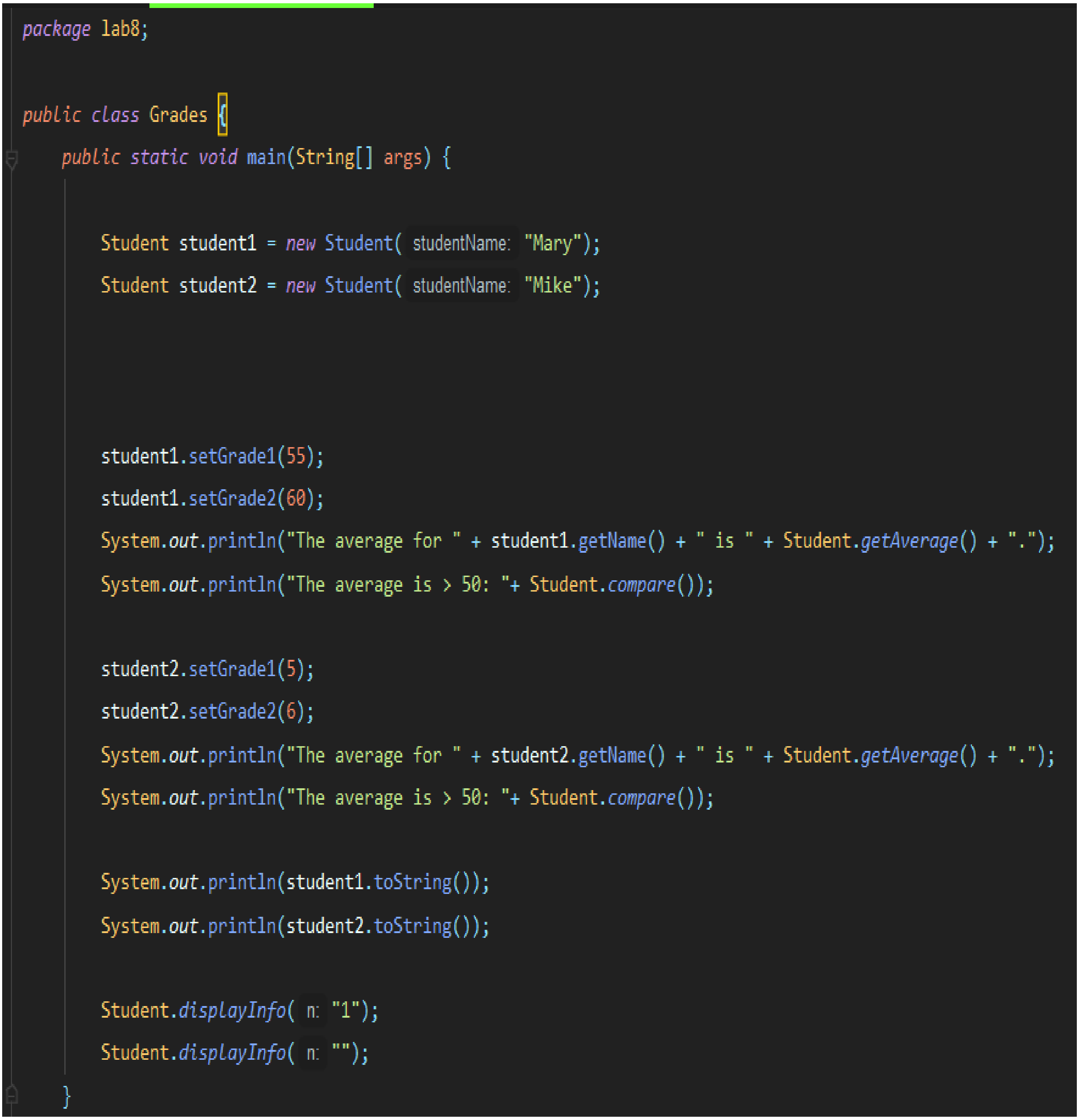
Answer:



```
The average for Mary is 57.5.
The average is > 50: true
The average for Mike is 5.5.
The average is > 50: false
Name: Mary   Test 1: 5   Test 2: 6
Name: Mike   Test 1: 5   Test 2: 6
I am method displayInfo1()
I am method displayInfo()
```



```java
package lab8;

public class Student {
    String name;
    static int test1;
    static int test2;

    public Student(String studentName) {
        name = studentName;
    }
    public String getName() {
        return name;
    }
    public String toString() {
        return "Name: " + name + "   Test 1: " + test1 + "   Test 2: " + test2;
    }
    public static void displayInfo(String n) {
        System.out.printf("I am method displayInfo%s()%n",n);
    }
    public void setGrade1(int test1) {
        Student.test1 = test1;
    }
    public void setGrade2(int test2) {
        Student.test2 = test2;
    }
    public static double getAverage() {
        double Ave = (test1 + test2)/2.0;
        return Ave;
    }
    public static boolean compare() {
        return Student.getAverage() > 50;
    }
```



```
package lab8;

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

        Student student1 = new Student( studentName: "Mary");
        Student student2 = new Student( studentName: "Mike");



        student1.setGrade1(55);
        student1.setGrade2(60);
        System.out.println("The average for " + student1.getName() + " is " + Student.getAverage() + ".");
        System.out.println("The average is > 50: "+ Student.compare());

        student2.setGrade1(5);
        student2.setGrade2(6);
        System.out.println("The average for " + student2.getName() + " is " + Student.getAverage() + ".");
        System.out.println("The average is > 50: "+ Student.compare());

        System.out.println(student1.toString());
        System.out.println(student2.toString());

        Student.displayInfo( n: "1");
        Student.displayInfo( n: "");
    }
```

Q-3 File Account.java contains a definition for a simple bank account class with methods to withdraw, deposit, get the balance, and return a String representation. Save this class to your directory and study it to see how it works. Then modify it as follows:

Overload the constructor as follows:

- public Account (double initBal, String owner, int number) - initializes the balance, owner, and account number
- public Account (double initBal, String owner) - initializes the balance and owner; and randomly generates the account number from [1,9] inclusive.
- public Account (String owner) - initializes the owner as specified; sets the initial balance to 0 and randomly generates the account number from [1,9] inclusive.

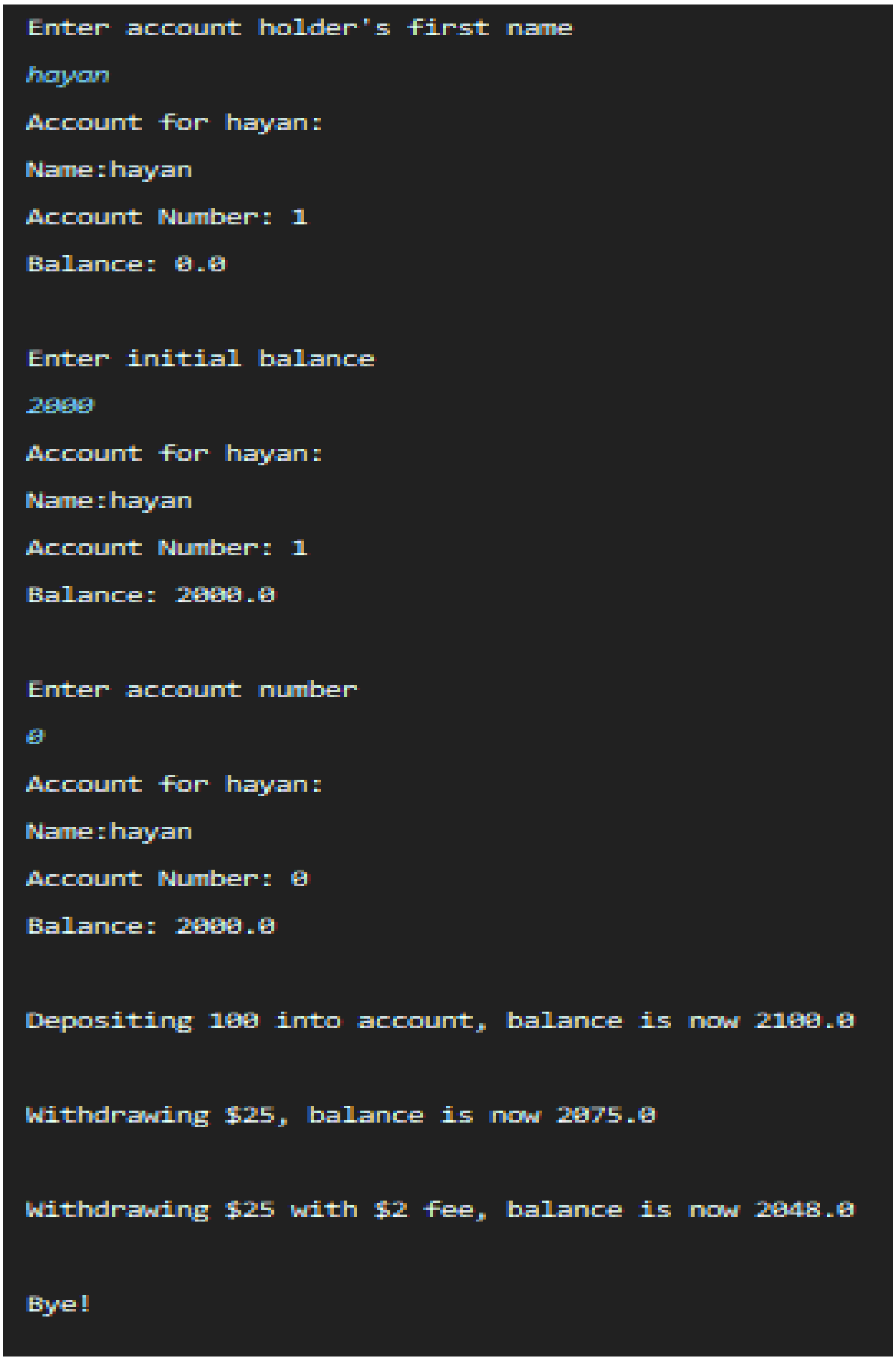
Overload the withdraw method with one that also takes a fee and deducts that fee from the account.

File TestAccount.java contains a simple program that exercises these methods. Save it to your directory, study it to see what it does, and use it to test your modified Account class.

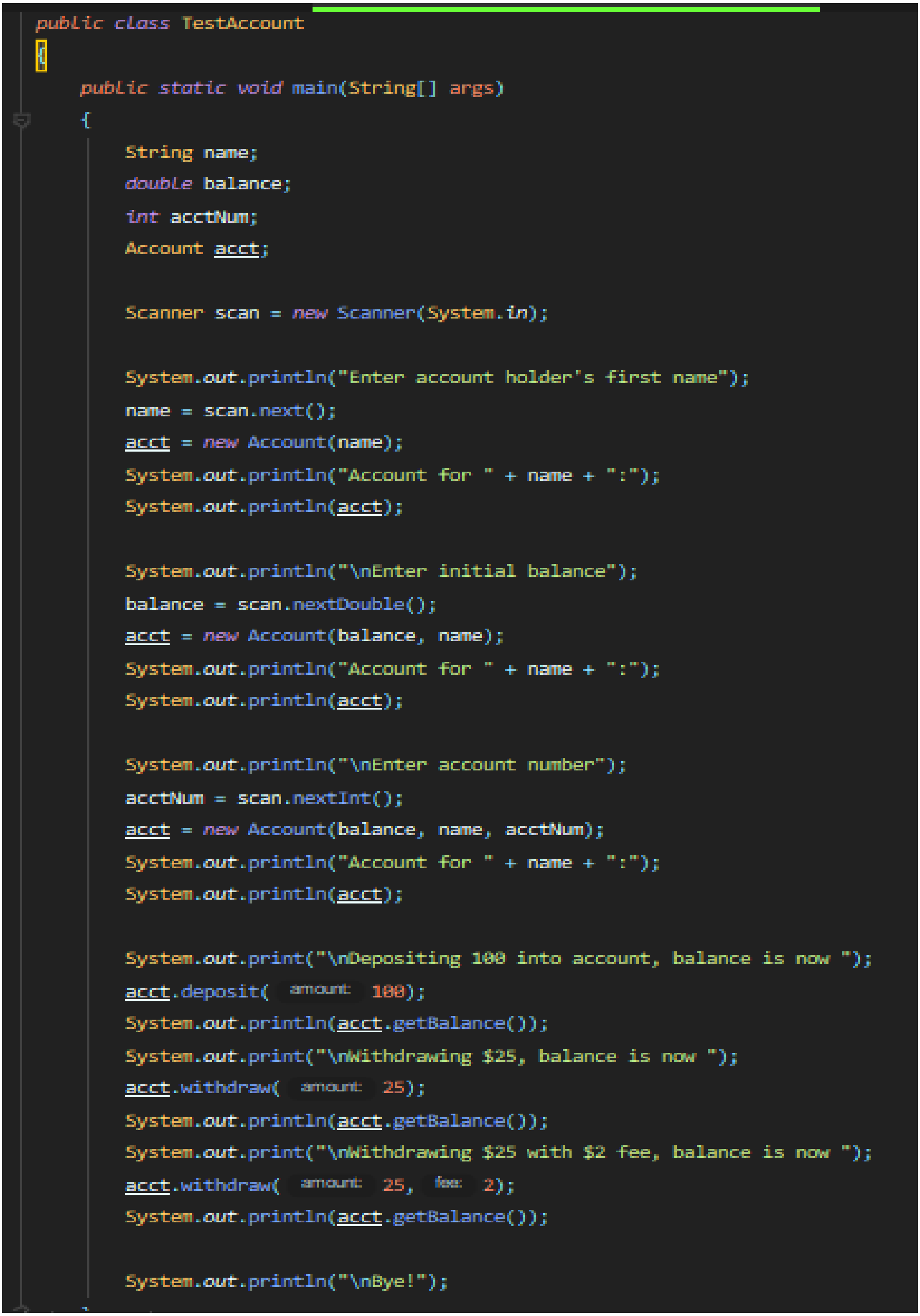
Sample Output

```
Enter account holder's first name
hanin
Account for hanin:
Name: hanin
Acct #: 7
Balance: 0.0

Enter initial balance
2000.5
Account for hanin:
Name: hanin
Acct #: 5
Balance: 2000.5

Enter account number
5
Account for hanin:
Name: hanin
Acct #: 5
Balance: 2000.5

Depositing 100 into account, balance is now 2100.5

Withdrawing $25, balance is now 2075.5

Withdrawing $25 with $2 fee, balance is now 2048.5

Bye!
```

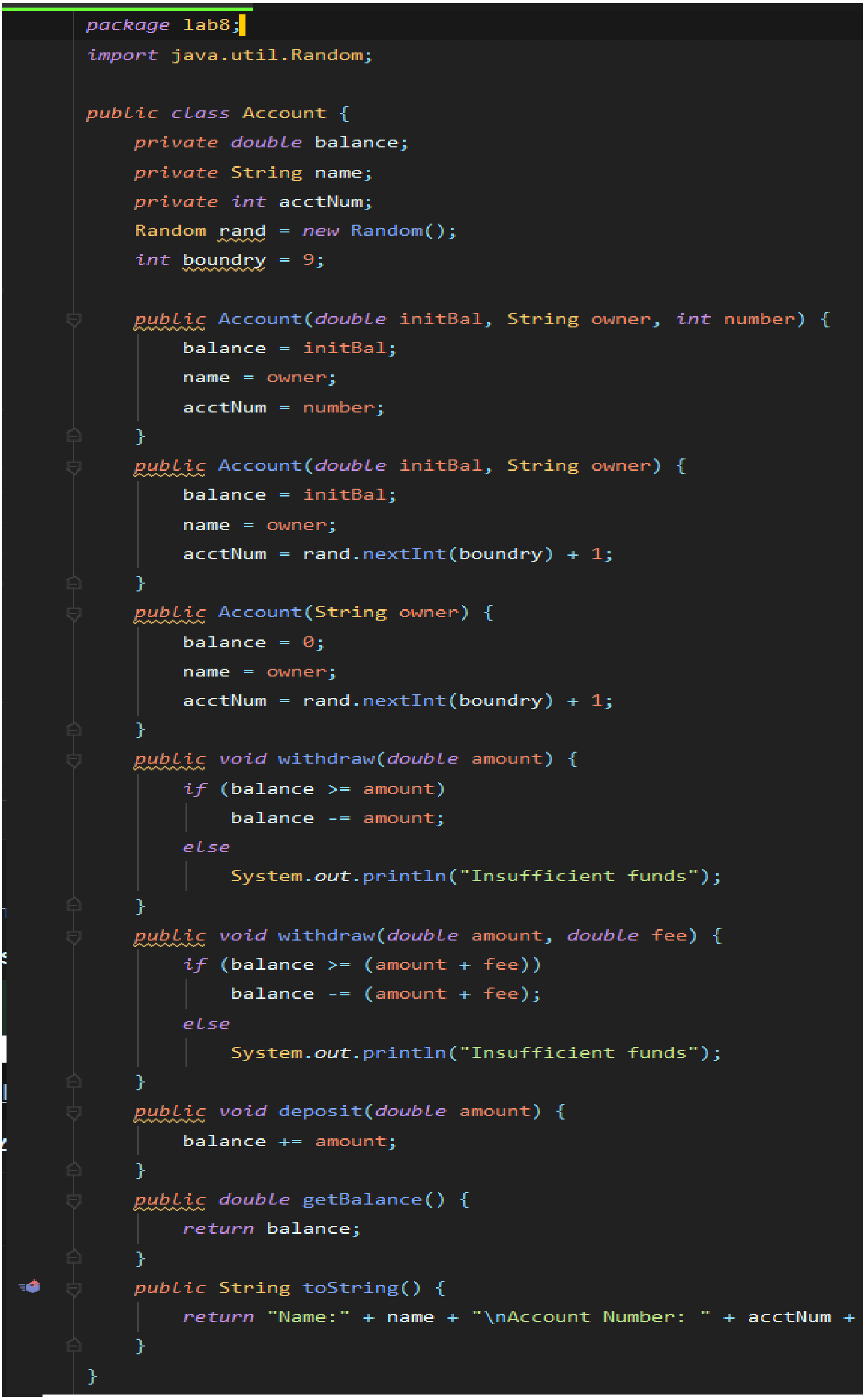
Answer:



```
Enter account holder's first name
hayan
Account for hayan:
Name:hayan
Account Number: 1
Balance: 0.0

Enter initial balance
2000
Account for hayan:
Name:hayan
Account Number: 1
Balance: 2000.0

Enter account number
0
Account for hayan:
Name:hayan
Account Number: 0
Balance: 2000.0

Depositing 100 into account, balance is now 2100.0

Withdrawing $25, balance is now 2075.0

Withdrawing $25 with $2 fee, balance is now 2048.0

Bye!
```



```
public class TestAccount
{
    public static void main(String[] args)
    {
        String name;
        double balance;
        int acctNum;
        Account acct;

        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();
        acct = new Account(balance, name);
        System.out.println("Account for " + name + ":");
        System.out.println(acct);

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

        System.out.print("\nDepositing 100 into account, balance is now ");
        acct.deposit( amount: 100);
        System.out.println(acct.getBalance());
        System.out.print("\nWithdrawing $25, balance is now ");
        acct.withdraw( amount: 25);
        System.out.println(acct.getBalance());
        System.out.print("\nWithdrawing $25 with $2 fee, balance is now ");
        acct.withdraw( amount: 25, fee: 2);
        System.out.println(acct.getBalance());

        System.out.println("\nBye!");
```



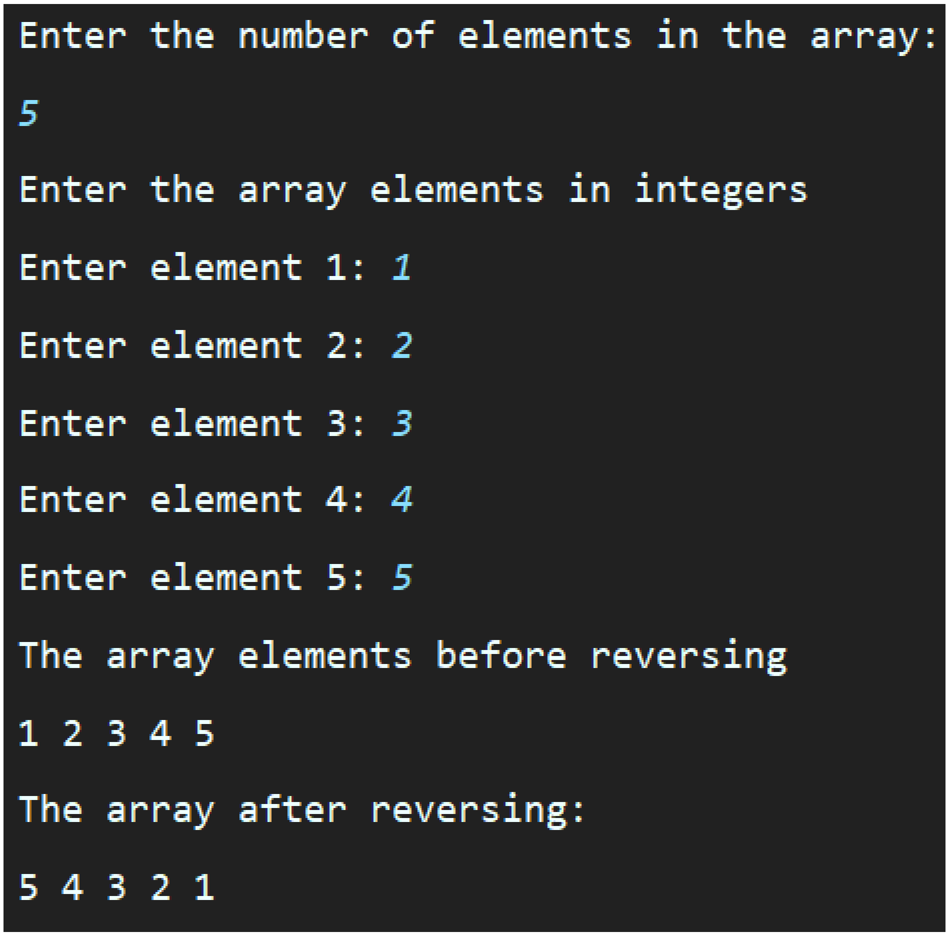
```java
package lab8;
import java.util.Random;

public class Account {
    private double balance;
    private String name;
    private int acctNum;
    Random rand = new Random();
    int boundry = 9;

    public Account(double initBal, String owner, int number) {
        balance = initBal;
        name = owner;
        acctNum = number;
    }
    public Account(double initBal, String owner) {
        balance = initBal;
        name = owner;
        acctNum = rand.nextInt(boundry) + 1;
    }
    public Account(String owner) {
        balance = 0;
        name = owner;
        acctNum = rand.nextInt(boundry) + 1;
    }
    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 + fee))
            balance -= (amount + fee);
        else
            System.out.println("Insufficient funds");
    }
    public void deposit(double amount) {
        balance += amount;
    }
    public double getBalance() {
        return balance;
    }
    public String toString() {
        return "Name:" + name + "\nAccount Number: " + acctNum +
    }
}
```

Q-4 Write a program that prompts the user for an integer, then asks the user to enter that many values. Store these values in an array and print the array. Then reverse the array elements so that the first element becomes the last element, the second element becomes the second to last element, and so on, with the old last element now first. Do not just reverse the order in which they are printed; actually change the way they are stored in the array. Do not create a second array; just rearrange the elements within the array you have. (Hint: Swap elements that need to change places.) When the elements have been reversed, print the array again.

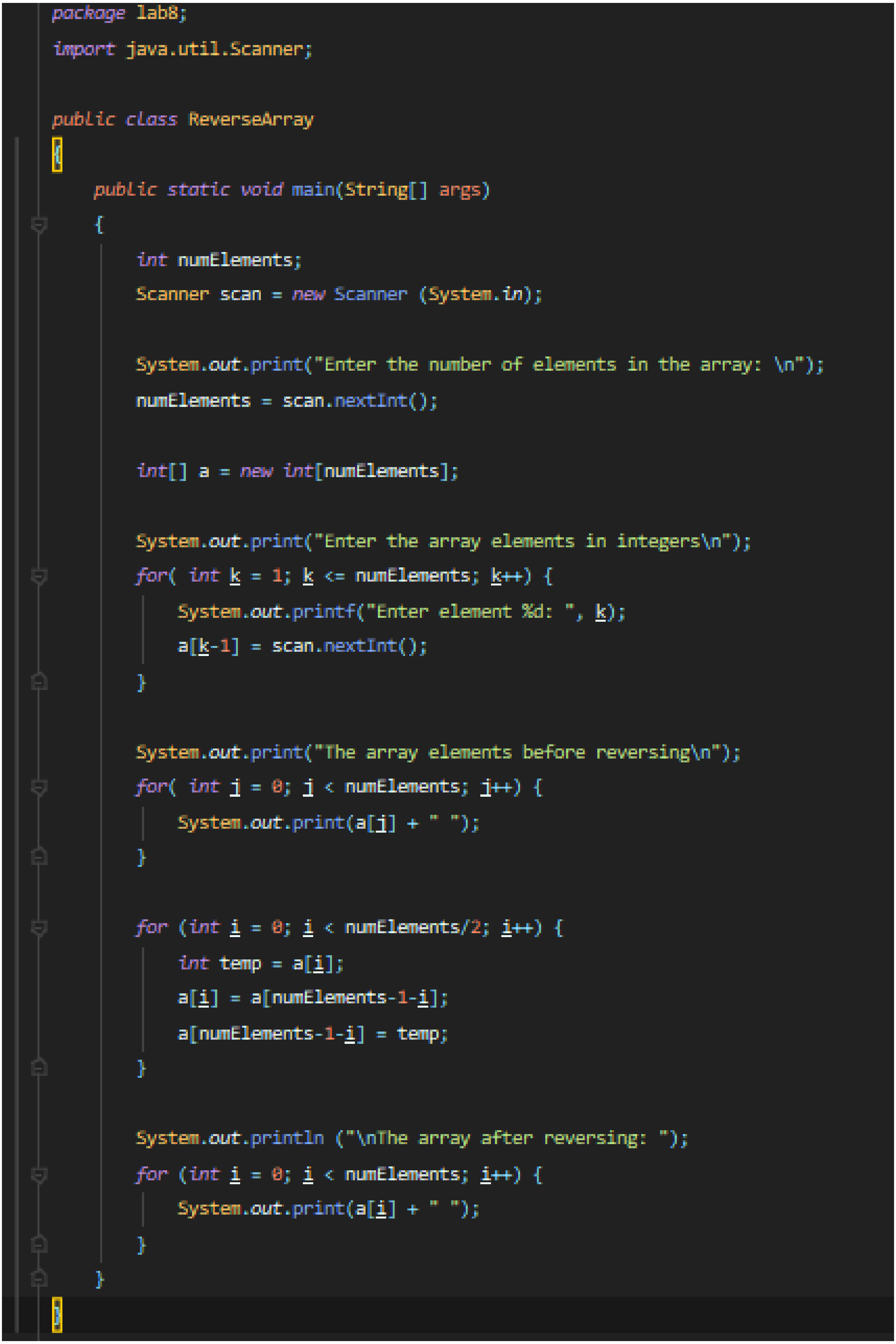
Sample Output

```
Enter the number of elements in the array: 5
Enter the array elements (integers)...
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Enter element 4: 4
Enter element 5: 5

The array elements before reversing:
1  2  3  4  5

The array after reversing:
5  4  3  2  1
```

Answer:



```
Enter the number of elements in the array:
5
Enter the array elements in integers
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Enter element 4: 4
Enter element 5: 5
The array elements before reversing
1 2 3 4 5
The array after reversing:
5 4 3 2 1
```



```java
package lab8;
import java.util.Scanner;

public class ReverseArray
{
    public static void main(String[] args)
    {
        int numElements;
        Scanner scan = new Scanner (System.in);

        System.out.print("Enter the number of elements in the array: \n");
        numElements = scan.nextInt();

        int[] a = new int[numElements];

        System.out.print("Enter the array elements in integers\n");
        for( int k = 1; k <= numElements; k++) {
            System.out.printf("Enter element %d: ", k);
            a[k-1] = scan.nextInt();
        }

        System.out.print("The array elements before reversing\n");
        for( int j = 0; j < numElements; j++) {
            System.out.print(a[j] + " ");
        }

        for (int i = 0; i < numElements/2; i++) {
            int temp = a[i];
            a[i] = a[numElements-1-i];
            a[numElements-1-i] = temp;
        }

        System.out.println ("\nThe array after reversing: ");
        for (int i = 0; i < numElements; i++) {
            System.out.print(a[i] + " ");
        }
    }
}
```