QB ESE

32 Print the Pattern as per given etc

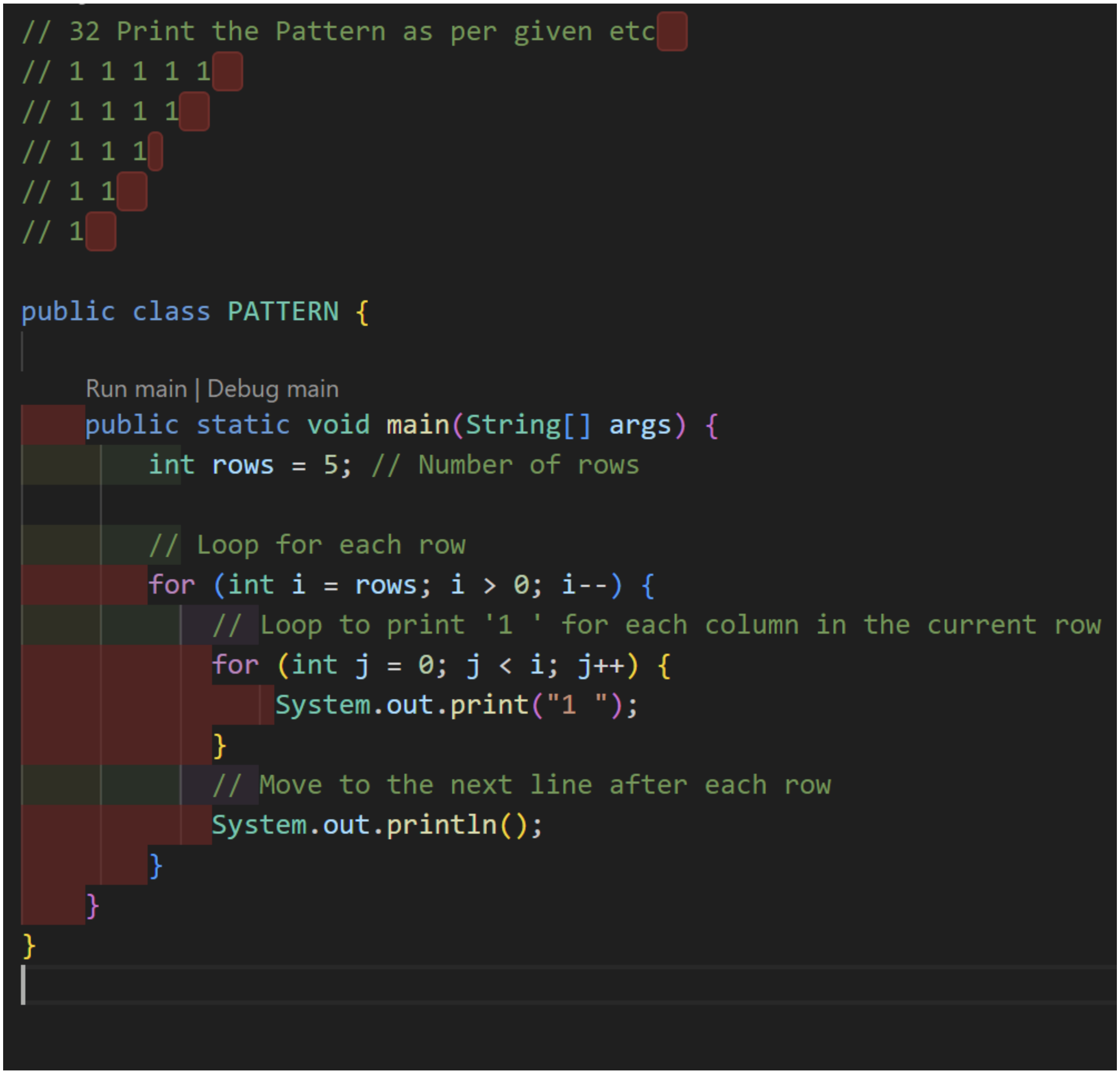
1 1 1 1 1

1 1 1 1

1 1 1

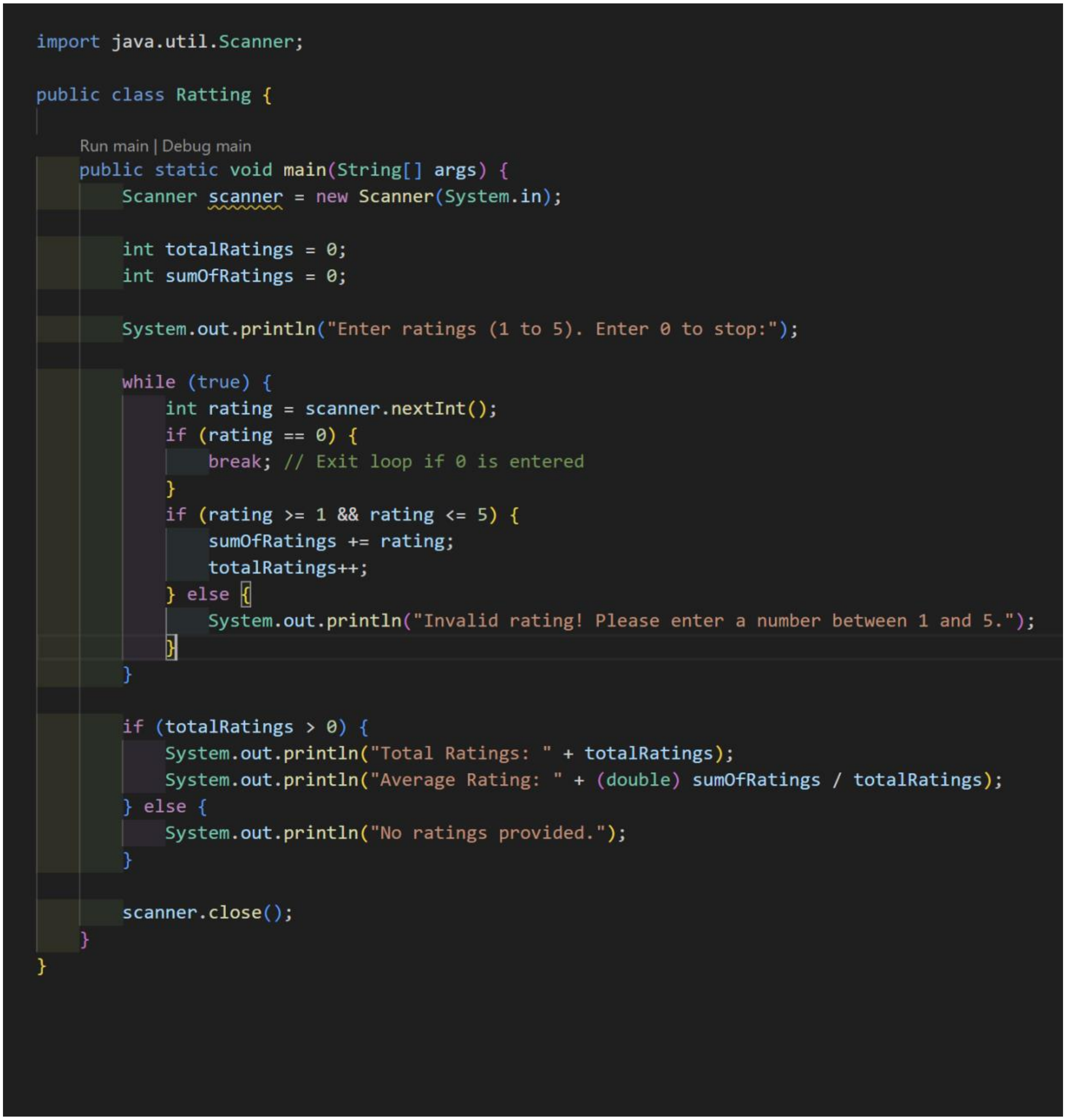
1 1

1



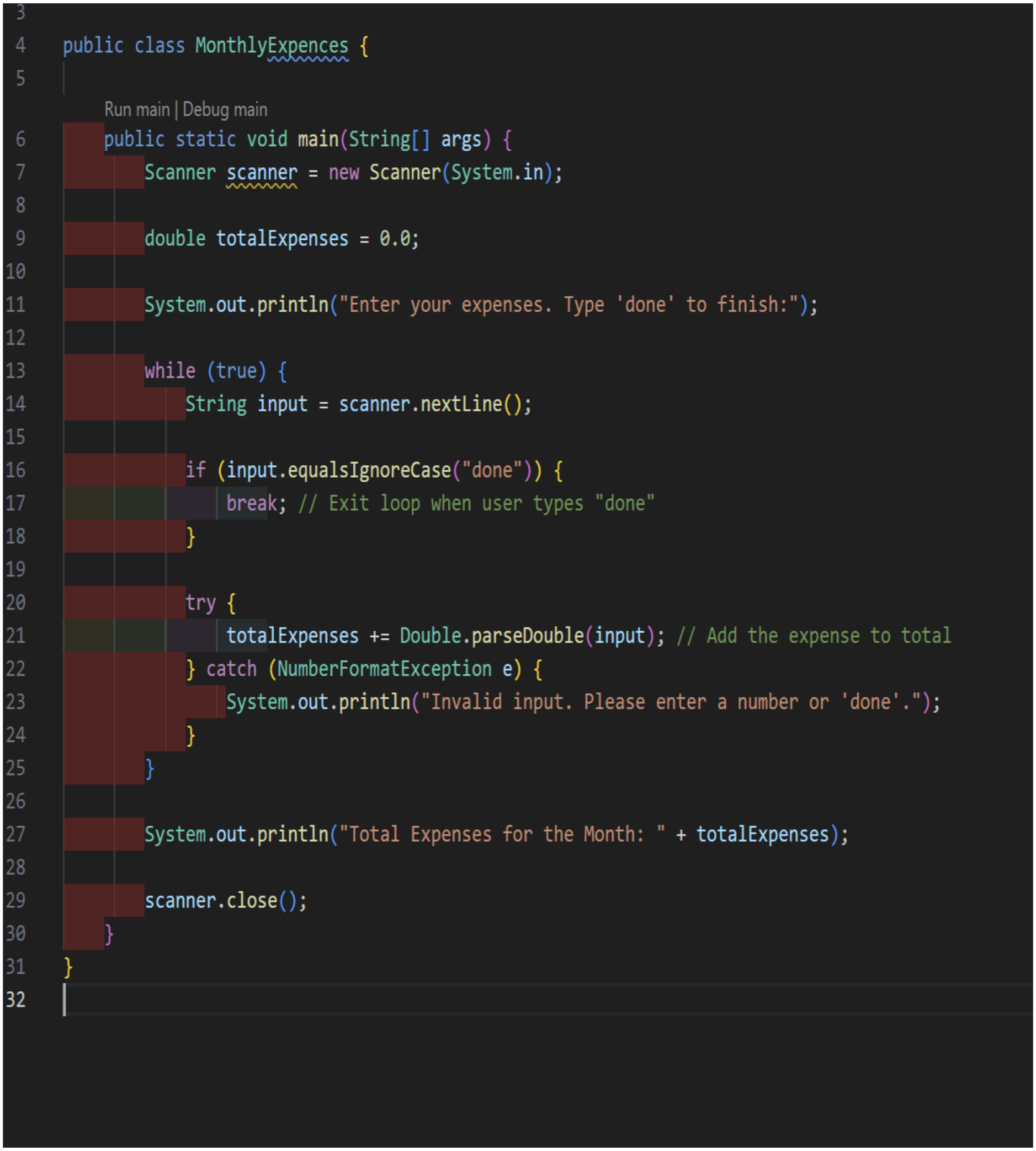
```
// 32 Print the Pattern as per given etc
// 1 1 1 1 1
// 1 1 1 1
// 1 1 1
// 1 1
// 1

public class PATTERN {

    public static void main(String[] args) {
        int rows = 5; // Number of rows

        // Loop for each row
        for (int i = rows; i > 0; i--) {
            // Loop to print '1 ' for each column in the current row
            for (int j = 0; j < i; j++) {
                System.out.print("1 ");
            }
            // Move to the next line after each row
            System.out.println();
        }
    }
}
```

33. As part of a feedback collection system, you want to gather customer ratings for a product. Design a program that prompts customers to rate the product from 1 to 5. Use a labeled while loop to continue collecting ratings until a customer enters 0. After collecting all ratings, compute and display the average rating and the number of ratings received.



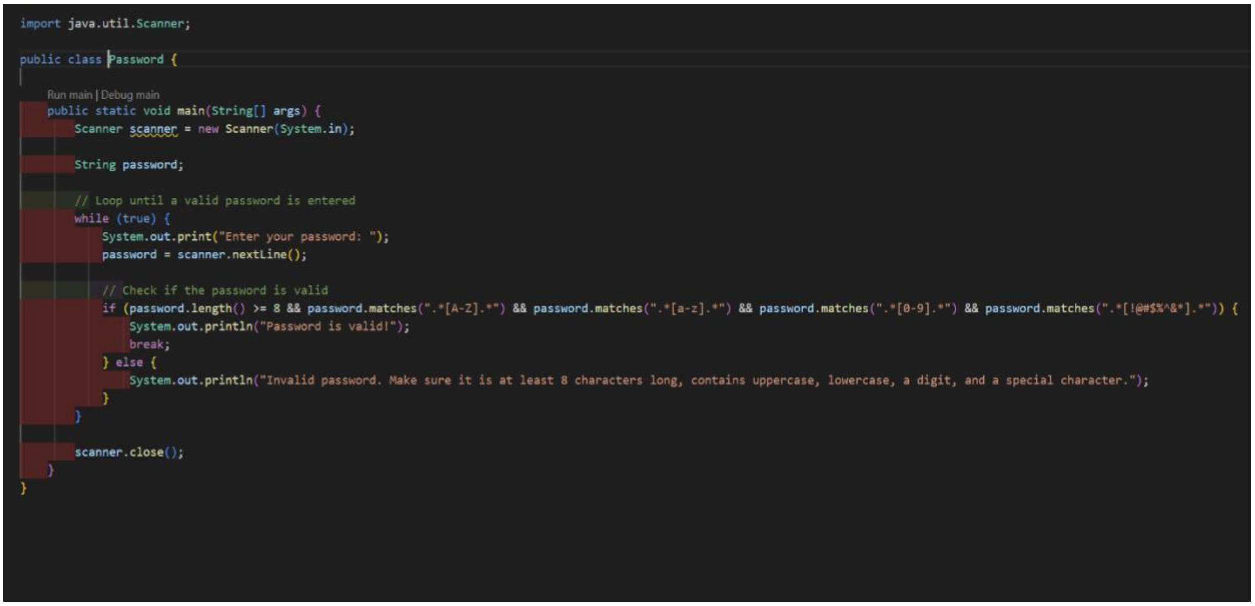
```java
import java.util.Scanner;

public class Ratting {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int totalRatings = 0;
        int sumOfRatings = 0;

        System.out.println("Enter ratings (1 to 5). Enter 0 to stop:");

        while (true) {
            int rating = scanner.nextInt();
            if (rating == 0) {
                break; // Exit loop if 0 is entered
            }
            if (rating >= 1 && rating <= 5) {
                sumOfRatings += rating;
                totalRatings++;
            } else {
                System.out.println("Invalid rating! Please enter a number between 1 and 5.");
            }
        }

        if (totalRatings > 0) {
            System.out.println("Total Ratings: " + totalRatings);
            System.out.println("Average Rating: " + (double) sumOfRatings / totalRatings);
        } else {
            System.out.println("No ratings provided.");
        }

        scanner.close();
    }
}
```

34. You are tasked with developing a program that tracks a user's monthly expenses. The program should repeatedly ask the user to input their expenses for different categories (like food, transportation, etc.) until they type "done". After the user is finished, display the total expenses for the month



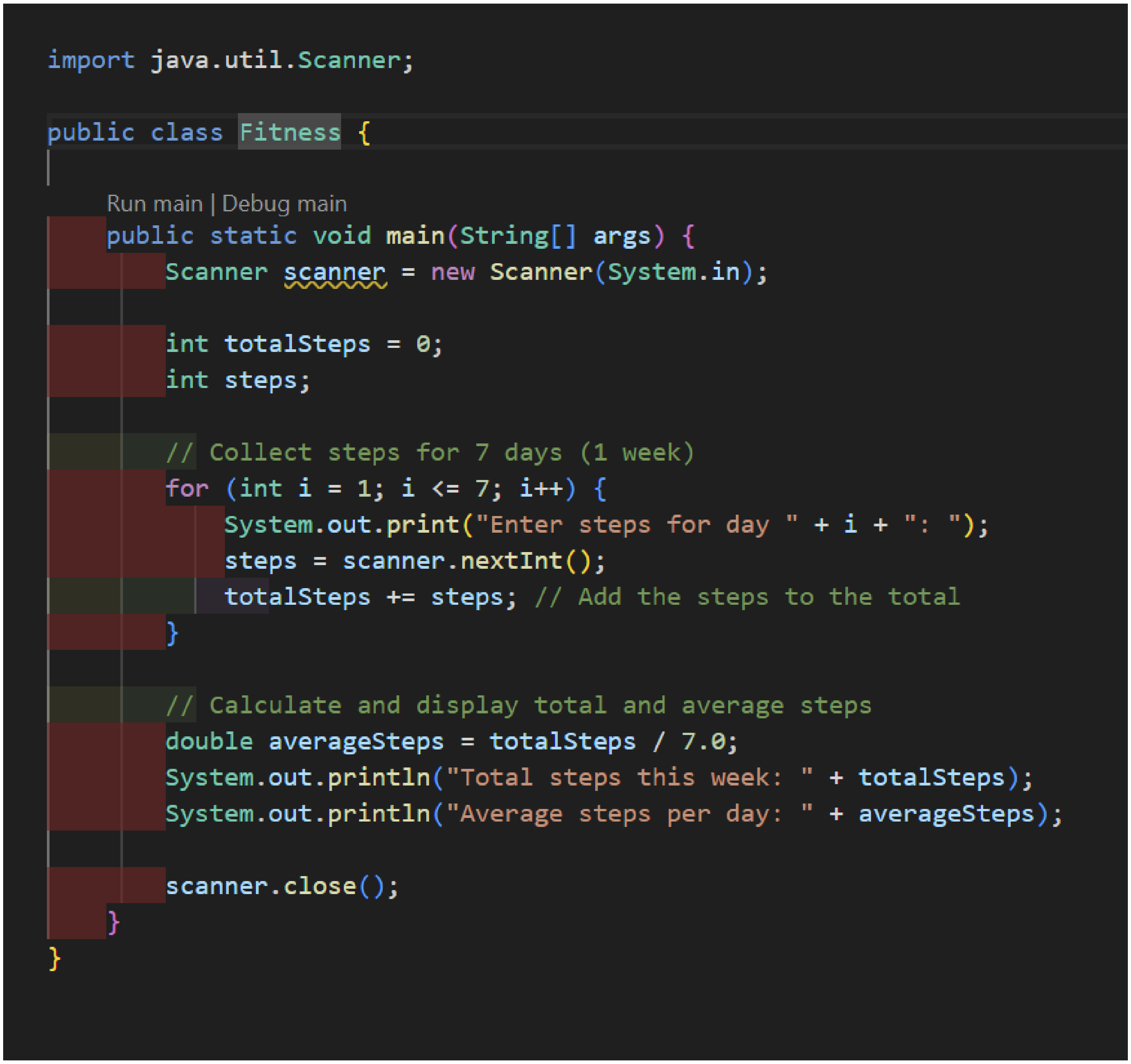
```java

public class MonthlyExpences {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        double totalExpenses = 0.0;

        System.out.println("Enter your expenses. Type 'done' to finish:");

        while (true) {
            String input = scanner.nextLine();

            if (input.equalsIgnoreCase("done")) {
                break; // Exit loop when user types "done"
            }

            try {
                totalExpenses += Double.parseDouble(input); // Add the expense to total
            } catch (NumberFormatException e) {
                System.out.println("Invalid input. Please enter a number or 'done'.");
            }
        }

        System.out.println("Total Expenses for the Month: " + totalExpenses);

        scanner.close();
    }
}

```

35. Develop a password validation system that prompts users to create a password. The program should check if the password meets certain criteria (length, special characters, etc.). If it doesn't meet the criteria, it should continue prompting the user until a valid password is entered.



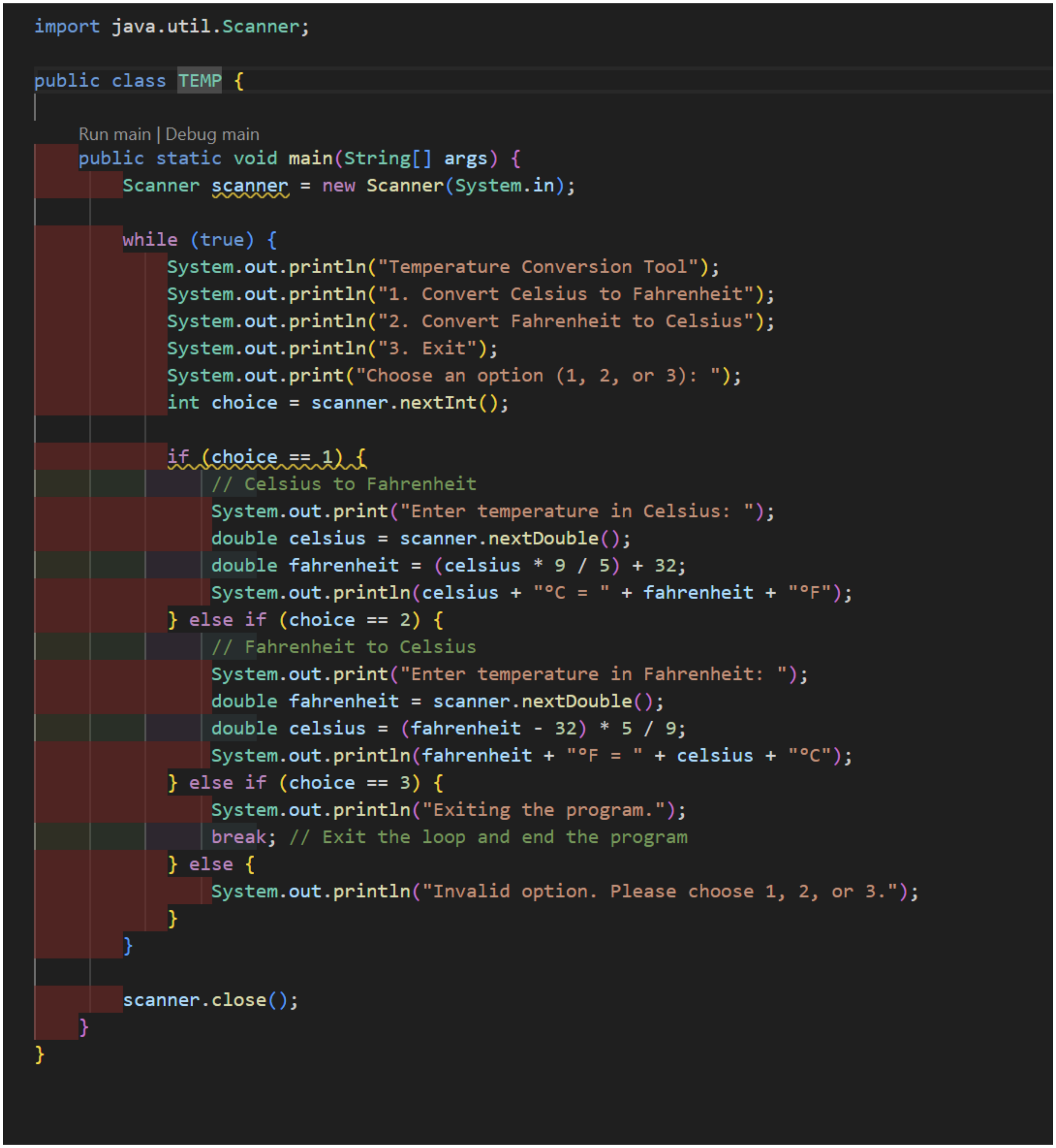
```java
import java.util.Scanner;

public class Password {

    Run main | Debug main
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        String password;

        // Loop until a valid password is entered
        while (true) {
            System.out.print("Enter your password: ");
            password = scanner.nextLine();

            // Check if the password is valid
            if (password.length() >= 8 && password.matches(".*[A-Z].*") && password.matches(".*[a-z].*") && password.matches(".*[0-9].*") && password.matches(".*[!@#$%^&*].*")) {
                System.out.println("Password is valid!");
                break;
            } else {
                System.out.println("Invalid password. Make sure it is at least 8 characters long, contains uppercase, lowercase, a digit, and a special character.");
            }
        }

        scanner.close();
    }
}
```

36. Create a fitness app that allows users to log their daily steps. The user should be prompted to enter their steps for each day of the week. Use a loop to collect this data, and at the end of the week, calculate and display the total steps taken and the average steps per day



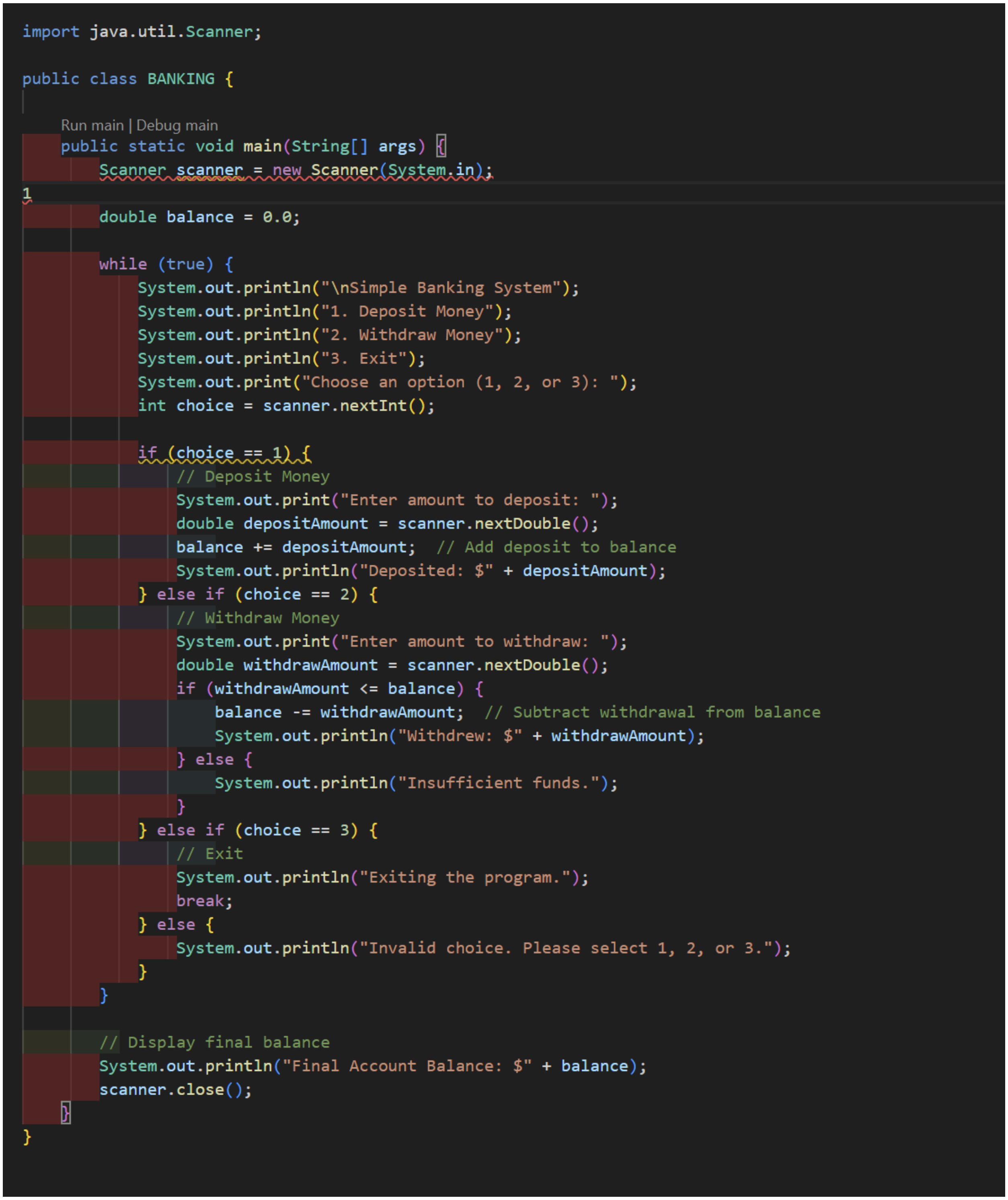
```java
import java.util.Scanner;

public class Fitness {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int totalSteps = 0;
        int steps;

        // Collect steps for 7 days (1 week)
        for (int i = 1; i <= 7; i++) {
            System.out.print("Enter steps for day " + i + ": ");
            steps = scanner.nextInt();
            totalSteps += steps; // Add the steps to the total
        }

        // Calculate and display total and average steps
        double averageSteps = totalSteps / 7.0;
        System.out.println("Total steps this week: " + totalSteps);
        System.out.println("Average steps per day: " + averageSteps);

        scanner.close();
    }
}
```

37. Develop a temperature conversion tool that allows users to convert temperatures between Celsius and Fahrenheit. Use a loop to continue asking for temperature values until the user chooses to exit. After each conversion, display the result and prompt the user again.



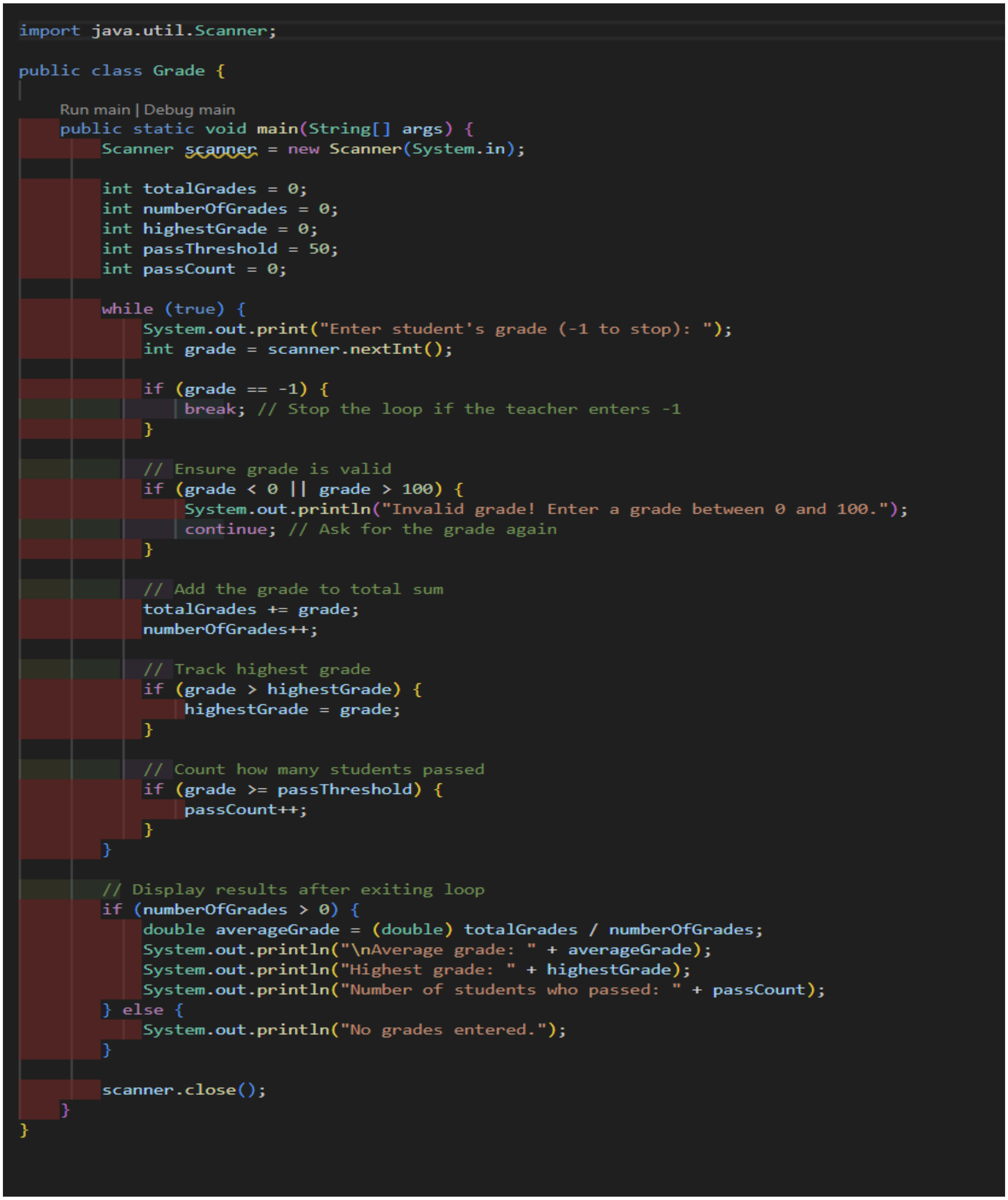
```java
import java.util.Scanner;

public class TEMP {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("Temperature Conversion Tool");
            System.out.println("1. Convert Celsius to Fahrenheit");
            System.out.println("2. Convert Fahrenheit to Celsius");
            System.out.println("3. Exit");
            System.out.print("Choose an option (1, 2, or 3): ");
            int choice = scanner.nextInt();

            if (choice == 1) {
                // Celsius to Fahrenheit
                System.out.print("Enter temperature in Celsius: ");
                double celsius = scanner.nextDouble();
                double fahrenheit = (celsius * 9 / 5) + 32;
                System.out.println(celsius + "°C = " + fahrenheit + "°F");
            } else if (choice == 2) {
                // Fahrenheit to Celsius
                System.out.print("Enter temperature in Fahrenheit: ");
                double fahrenheit = scanner.nextDouble();
                double celsius = (fahrenheit - 32) * 5 / 9;
                System.out.println(fahrenheit + "°F = " + celsius + "°C");
            } else if (choice == 3) {
                System.out.println("Exiting the program.");
                break; // Exit the loop and end the program
            } else {
                System.out.println("Invalid option. Please choose 1, 2, or 3.");
            }
        }

        scanner.close();
    }
}
```

38. Implement a simple banking system where users can deposit and withdraw money. Use a loop to allow the user to perform transactions until they choose to exit. After exiting, display the final account balance and transaction history.



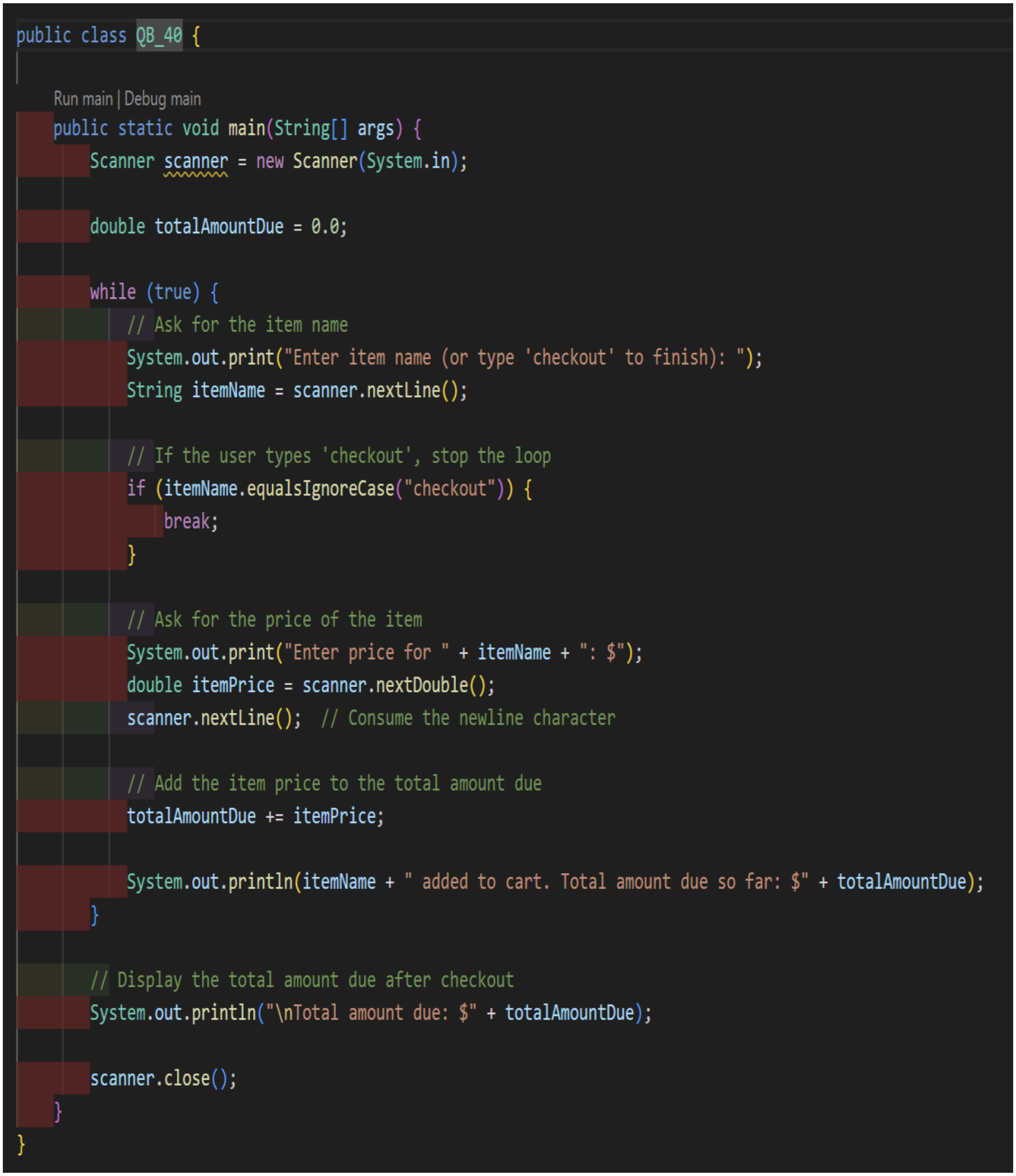
```java
import java.util.Scanner;

public class BANKING {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
1
        double balance = 0.0;

        while (true) {
            System.out.println("\nSimple Banking System");
            System.out.println("1. Deposit Money");
            System.out.println("2. Withdraw Money");
            System.out.println("3. Exit");
            System.out.print("Choose an option (1, 2, or 3): ");
            int choice = scanner.nextInt();

            if (choice == 1) {
                // Deposit Money
                System.out.print("Enter amount to deposit: ");
                double depositAmount = scanner.nextDouble();
                balance += depositAmount;  // Add deposit to balance
                System.out.println("Deposited: $" + depositAmount);
            } else if (choice == 2) {
                // Withdraw Money
                System.out.print("Enter amount to withdraw: ");
                double withdrawAmount = scanner.nextDouble();
                if (withdrawAmount <= balance) {
                    balance -= withdrawAmount;  // Subtract withdrawal from balance
                    System.out.println("Withdrew: $" + withdrawAmount);
                } else {
                    System.out.println("Insufficient funds.");
                }
            } else if (choice == 3) {
                // Exit
                System.out.println("Exiting the program.");
                break;
            } else {
                System.out.println("Invalid choice. Please select 1, 2, or 3.");
            }
        }

        // Display final balance
        System.out.println("Final Account Balance: $" + balance);
        scanner.close();
    }
}
```

39. Create a program that allows a teacher to input grades for students in a class. The program should continue to prompt for grades until the teacher enters -1 to stop. After all grades have been entered, calculate and display the average grade, the highest grade, and the number of students who passed (e.g., scored above a certain threshold). share code



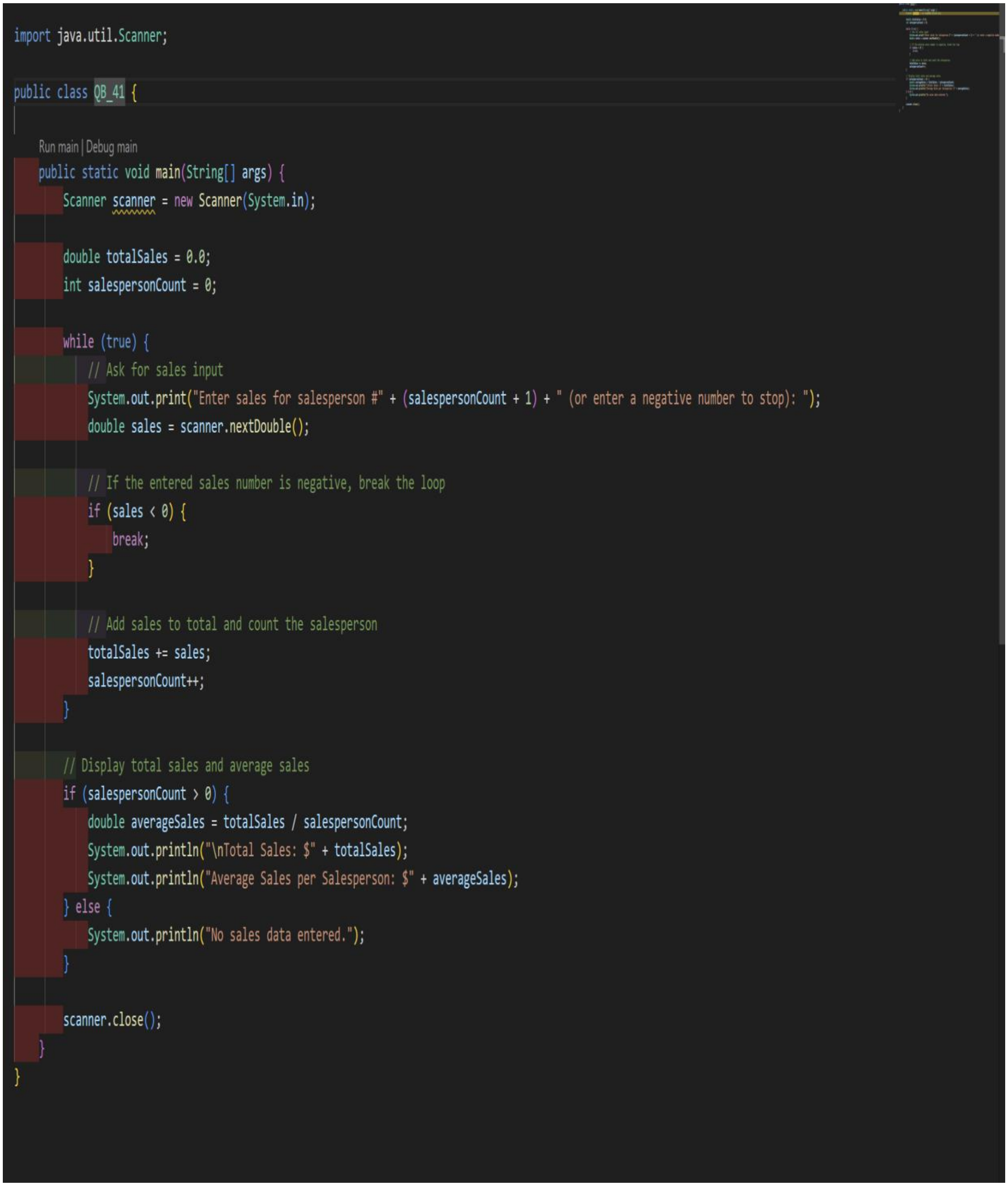
```java
import java.util.Scanner;

public class Grade {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int totalGrades = 0;
        int numberOfGrades = 0;
        int highestGrade = 0;
        int passThreshold = 50;
        int passCount = 0;

        while (true) {
            System.out.print("Enter student's grade (-1 to stop): ");
            int grade = scanner.nextInt();

            if (grade == -1) {
                break; // Stop the loop if the teacher enters -1
            }

            // Ensure grade is valid
            if (grade < 0 || grade > 100) {
                System.out.println("Invalid grade! Enter a grade between 0 and 100.");
                continue; // Ask for the grade again
            }

            // Add the grade to total sum
            totalGrades += grade;
            numberOfGrades++;

            // Track highest grade
            if (grade > highestGrade) {
                highestGrade = grade;
            }

            // Count how many students passed
            if (grade >= passThreshold) {
                passCount++;
            }
        }

        // Display results after exiting loop
        if (numberOfGrades > 0) {
            double averageGrade = (double) totalGrades / numberOfGrades;
            System.out.println("\nAverage grade: " + averageGrade);
            System.out.println("Highest grade: " + highestGrade);
            System.out.println("Number of students who passed: " + passCount);
        } else {
            System.out.println("No grades entered.");
        }

        scanner.close();
    }
}
```

40. Design a shopping cart application that allows users to add items to their cart. The program should ask the user for item names and prices in a loop until the user types



```java
public class QB_40 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        double totalAmountDue = 0.0;

        while (true) {
            // Ask for the item name
            System.out.print("Enter item name (or type 'checkout' to finish): ");
            String itemName = scanner.nextLine();

            // If the user types 'checkout', stop the loop
            if (itemName.equalsIgnoreCase("checkout")) {
                break;
            }

            // Ask for the price of the item
            System.out.print("Enter price for " + itemName + ": $");
            double itemPrice = scanner.nextDouble();
            scanner.nextLine();  // Consume the newline character

            // Add the item price to the total amount due
            totalAmountDue += itemPrice;

            System.out.println(itemName + " added to cart. Total amount due so far: $" + totalAmountDue);
        }

        // Display the total amount due after checkout
        System.out.println("\nTotal amount due: $" + totalAmountDue);

        scanner.close();
    }
}
```

41. Write a program that calculates the total sales and commission for a group of salespeople. Prompt the user to enter sales figures for each salesperson in a loop. The loop should continue until a negative number is entered, indicating the end of input. Calculate and display the total sales and the average sales per salesperson.



```java
import java.util.Scanner;

public class QB_41 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        double totalSales = 0.0;
        int salespersonCount = 0;

        while (true) {
            // Ask for sales input
            System.out.print("Enter sales for salesperson #" + (salespersonCount + 1) + " (or enter a negative number to stop): ");
            double sales = scanner.nextDouble();

            // If the entered sales number is negative, break the loop
            if (sales < 0) {
                break;
            }

            // Add sales to total and count the salesperson
            totalSales += sales;
            salespersonCount++;
        }

        // Display total sales and average sales
        if (salespersonCount > 0) {
            double averageSales = totalSales / salespersonCount;
            System.out.println("\nTotal Sales: $" + totalSales);
            System.out.println("Average Sales per Salesperson: $" + averageSales);
        } else {
            System.out.println("No sales data entered.");
        }

        scanner.close();
    }
}
```

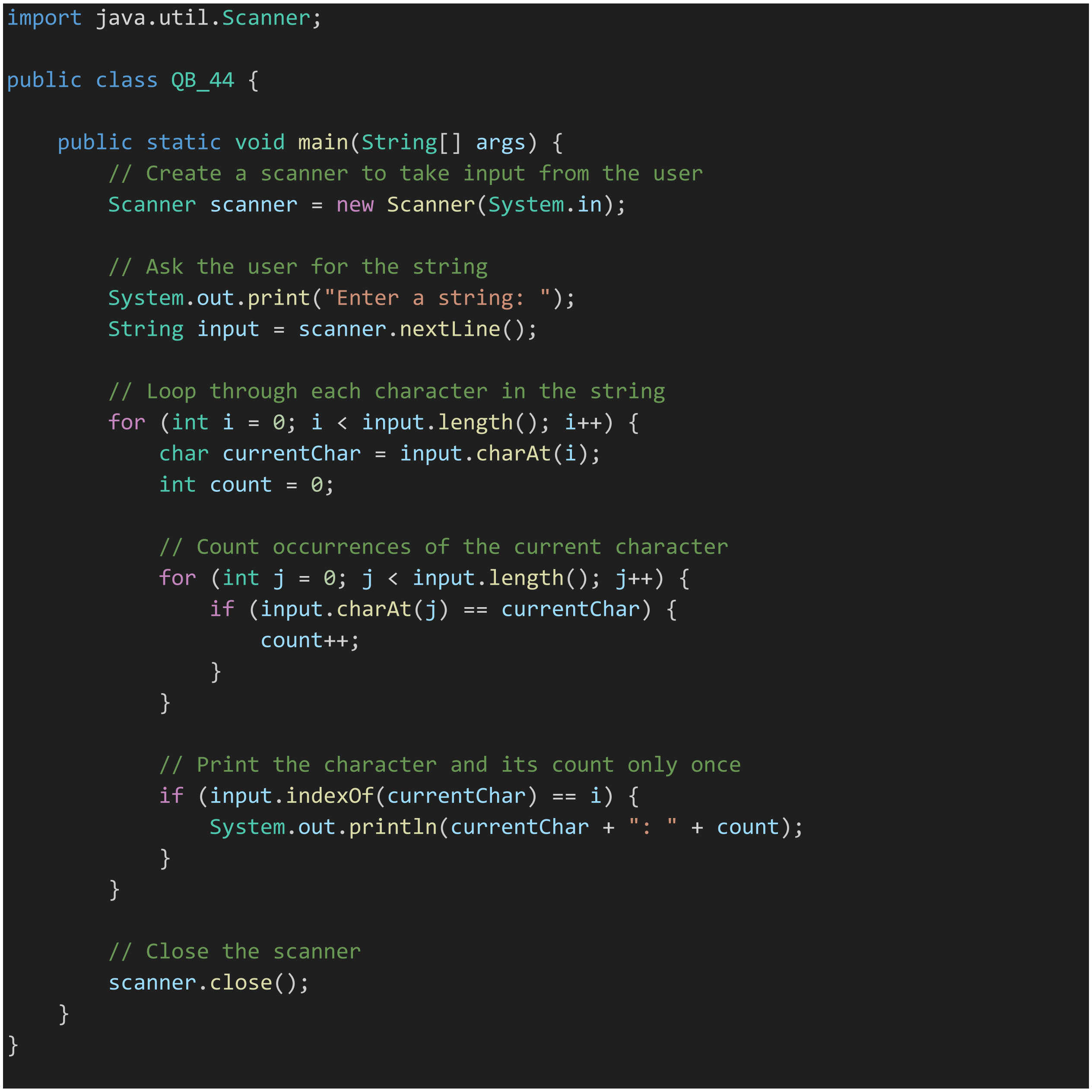
42. Write a Java program to reverse a String.

```java
import java.util.Scanner;

public class QB_42 {

    public static void main(String[] args) {
        // Create a scanner to read input
        Scanner scanner = new Scanner(System.in);

        // Ask for the user input string
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        // Reverse the string using StringBuilder
        String reversed = new StringBuilder(input).reverse().toString();

        // Display the reversed string
        System.out.println("Reversed String: " + reversed);

        // Close the scanner
        scanner.close();
    }
}
```

43. How would you check if a String is a palindrome in Java?

```
import java.util.Scanner;

public class QB_43 {

    public static void main(String[] args) {
        // Create a scanner object to read input
        Scanner scanner = new Scanner(System.in);

        // Ask the user for the string
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        // Check if the string is a palindrome
        if (isPalindrome(input)) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }

        // Close the scanner
        scanner.close();
    }

    // Method to check if the string is a palindrome
    public static boolean isPalindrome(String str) {
        int start = 0;
        int end = str.length() - 1;

        // Compare characters from start and end
        while (start < end) {
            if (str.charAt(start) != str.charAt(end)) {
                return false;  // Not a palindrome if characters don't match
            }
            start++;
            end--;
        }

        return true;  // The string is a palindrome
    }
}
```

44. How would you identify and count the occurrences of each character in a String?



```java
import java.util.Scanner;

public class QB_44 {

    public static void main(String[] args) {
        // Create a scanner to take input from the user
        Scanner scanner = new Scanner(System.in);

        // Ask the user for the string
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        // Loop through each character in the string
        for (int i = 0; i < input.length(); i++) {
            char currentChar = input.charAt(i);
            int count = 0;

            // Count occurrences of the current character
            for (int j = 0; j < input.length(); j++) {
                if (input.charAt(j) == currentChar) {
                    count++;
                }
            }

            // Print the character and its count only once
            if (input.indexOf(currentChar) == i) {
                System.out.println(currentChar + ": " + count);
            }
        }

        // Close the scanner
        scanner.close();
    }
}
```

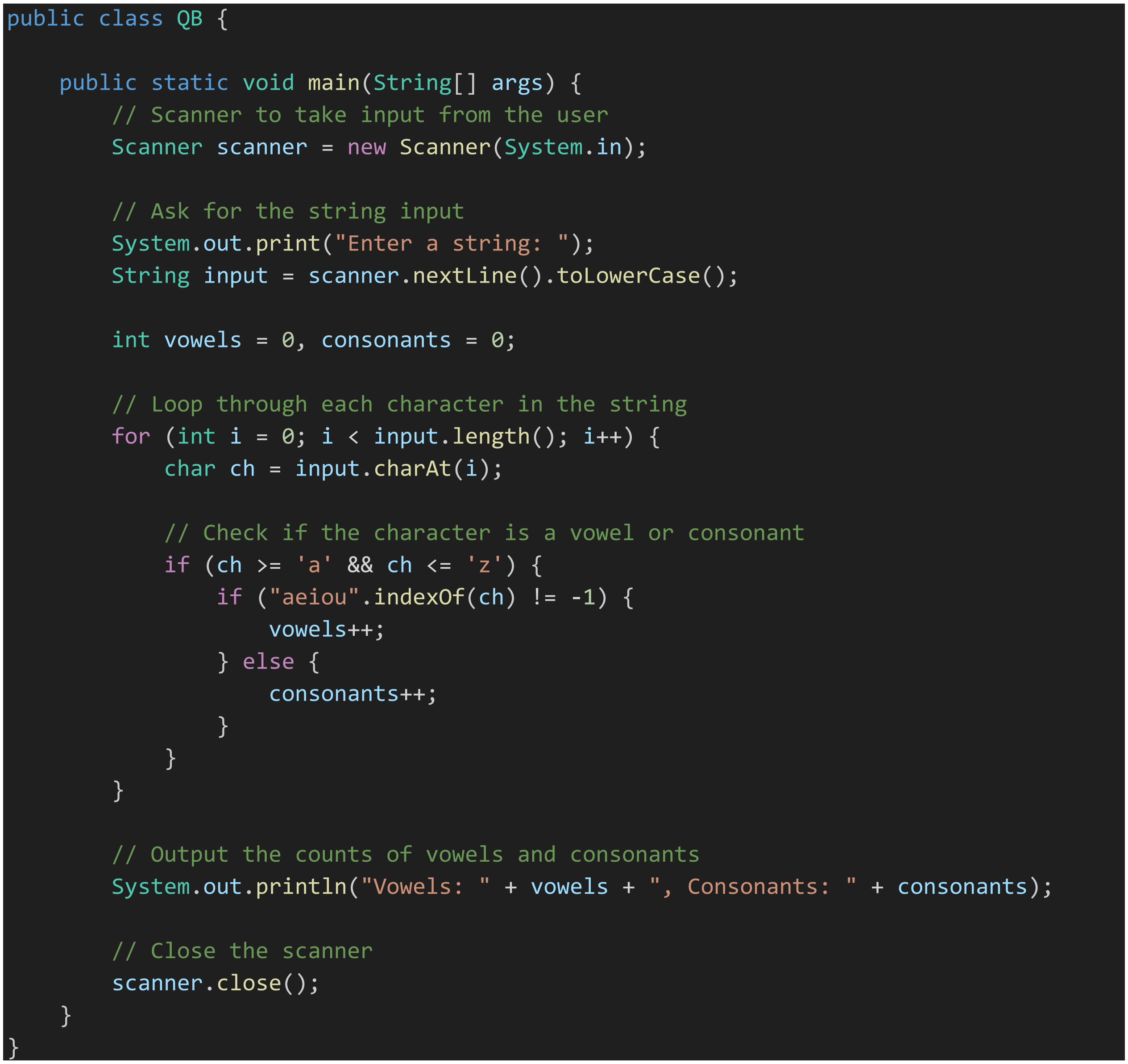
45. Write a Java program to reverse a given String without using the built-in reverse method. Example: Input: "Hello" Output: "olleH"

```
import java.util.Scanner;

public class QB_45 {

    public static void main(String[] args) {
        // Scanner to take input from the user
        Scanner scanner = new Scanner(System.in);

        // Ask for the string input
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        // Variable to store the reversed string
        String reversed = "";

        // Loop through the string from end to start
        for (int i = input.length() - 1; i >= 0; i--) {
            reversed += input.charAt(i);  // Append each character to the reversed
string
        }

        // Display the reversed string
        System.out.println("Reversed String: " + reversed);

        // Close the scanner
        scanner.close();
    }
}
```

46. Create a method that checks if a given String is a palindrome (reads the same forwards and backwards). Input: "racecar" Output: true

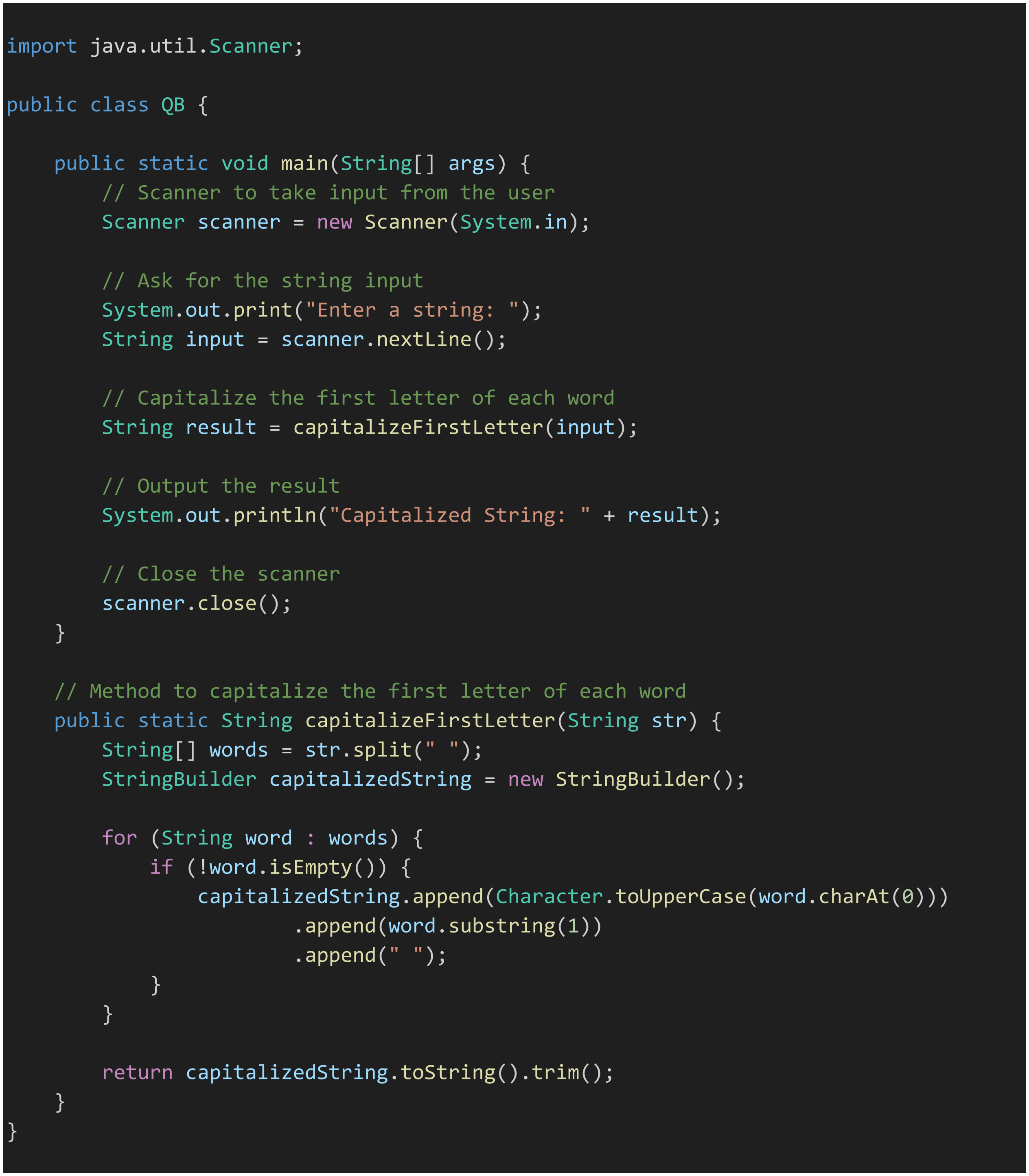
```
import java.util.Scanner;

public class QB_45 {

    public static void main(String[] args) {
        // Scanner to take input from the user
        Scanner scanner = new Scanner(System.in);

        // Ask for the string input
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        // Check if the string is a palindrome
        if (new StringBuilder(input).reverse().toString().equals(input)) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }

        // Close the scanner
        scanner.close();
    }
}
```

47. Write a program that counts the number of vowels and consonants in a given String. Input: "Hello World" Output: Vowels: 3, Consonants: 7



```java
public class QB {

    public static void main(String[] args) {
        // Scanner to take input from the user
        Scanner scanner = new Scanner(System.in);

        // Ask for the string input
        System.out.print("Enter a string: ");
        String input = scanner.nextLine().toLowerCase();

        int vowels = 0, consonants = 0;

        // Loop through each character in the string
        for (int i = 0; i < input.length(); i++) {
            char ch = input.charAt(i);

            // Check if the character is a vowel or consonant
            if (ch >= 'a' && ch <= 'z') {
                if ("aeiou".indexOf(ch) != -1) {
                    vowels++;
                } else {
                    consonants++;
                }
            }
        }

        // Output the counts of vowels and consonants
        System.out.println("Vowels: " + vowels + ", Consonants: " + consonants);

        // Close the scanner
        scanner.close();
    }
}
```

48 Implement a method that capitalizes the first letter of each word in a given String. Input: "hello world" Output: "Hello World"



```java
import java.util.Scanner;

public class QB {

    public static void main(String[] args) {
        // Scanner to take input from the user
        Scanner scanner = new Scanner(System.in);

        // Ask for the string input
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        // Capitalize the first letter of each word
        String result = capitalizeFirstLetter(input);

        // Output the result
        System.out.println("Capitalized String: " + result);

        // Close the scanner
        scanner.close();
    }

    // Method to capitalize the first letter of each word
    public static String capitalizeFirstLetter(String str) {
        String[] words = str.split(" ");
        StringBuilder capitalizedString = new StringBuilder();

        for (String word : words) {
            if (!word.isEmpty()) {
                capitalizedString.append(Character.toUpperCase(word.charAt(0)))
                        .append(word.substring(1))
                        .append(" ");
            }
        }

        return capitalizedString.toString().trim();
    }
}
```

49. Implement a method that checks if two Strings are anagrams of each other (contain the same characters in a different order). Input: "listen", "silent" Output: true

```java
import java.util.Arrays;
public class QB {
    public static void main(String[] args) {
        // Test strings
        String str1 = "listen";
        String str2 = "silent";

        // Check and print if the strings are anagrams
        System.out.println(areAnagrams(str1, str2));
    }

    // Method to check if two strings are anagrams
    public static boolean areAnagrams(String str1, String str2) {
        return str1.length() == str2.length()
                && Arrays.equals(
                        str1.chars().sorted().toArray(),
                        str2.chars().sorted().toArray()
                );
    }
}
```

50. Write a program to remove duplicate characters from a String while maintaining the original order of characters. Input: "programming" Output: "progamin"

```java
public class QB {
    public static void main(String[] args) {
        String input = "omsairam";
        System.out.println(removeDuplicates(input));
    }
    // Method to remove duplicate characters
    public static String removeDuplicates(String str) {
        StringBuilder result = new StringBuilder();
        for (char ch : str.toCharArray()) {
            if (result.indexOf(String.valueOf(ch)) == -1) {
                result.append(ch);
            }
        }
        return result.toString();
    }
}
```