Static Members in Different Contexts

Write a Java program to create a class called "BankAccount" with instance variables 'accountNumber' and balance, and static variables 'bankName' and 'interestRate'. Provide static methods to get and set the static variables. Create several 'BankAccount' objects and print their details along with the static variables.

**Sample Solution:**

**Java Code:**

**BankAccount.java**

```java
// Define the BankAccount class
public class BankAccount {
    // Instance variables
    private String accountNumber;
    private double balance;

    // Static variables
    private static String bankName;
    private static double interestRate;

    // Constructor to initialize instance variables
    public BankAccount(String accountNumber, double balance) {
        this.accountNumber = accountNumber;
        this.balance = balance;
    }

    // Static method to get the bank name
    public static String getBankName() {
        return bankName;
    }

    // Static method to set the bank name
    public static void setBankName(String name) {
        bankName = name;
    }

    // Static method to get the interest rate
    public static double getInterestRate() {
        return interestRate;
    }

    // Static method to set the interest rate
    public static void setInterestRate(double rate) {
        interestRate = rate;
    }

    // Method to print the details of the BankAccount object
    public void printDetails() {
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Balance: " + balance);
        System.out.println("Bank Name: " + bankName);
        System.out.println("Interest Rate: " + interestRate + "%");
    }

    // Main method to demonstrate the functionality
    public static void main(String[] args) {
        // Set static variables
        BankAccount.setBankName("National Bank");
        BankAccount.setInterestRate(3.5);

        // Create BankAccount objects
        BankAccount account1 = new BankAccount("123456789", 1000.00);
        BankAccount account2 = new BankAccount("987654321", 2000.00);

        // Print details of each BankAccount object
        account1.printDetails();
        System.out.println();
        account2.printDetails();
    }
}
```

Output:

```
Accoun
Balanc
Bank N
Intere

Accoun
Balanc
```

```
Bank Name: National Bank
Interest Rate: 3.5%
```

**Explanation:**

- **Define the BankAccount class:**
  - The BankAccount class is defined with instance variables accountNumber and balance.
- **Static variables:**
  - The static variables bankName and interestRate are declared within the class. These variables are shared across all instances of the BankAccount class.
- **Constructor to initialize instance variables:**
  - A constructor is provided to initialize the accountNumber and balance for each BankAccount object created.
- **Static methods to get and set static variables:**
  - getBankName and setBankName methods are used to get and set the static variable bankName.
  - getInterestRate and setInterestRate methods are used to get and set the static variable interestRate.
- **Method to print the details of the BankAccount object:**
  - The printDetails method prints the details of a BankAccount object, including the static variables.
- **Main method to demonstrate the functionality:**
  - The main method sets the static variables bankName and interestRate, creates two BankAccount objects, and prints their details using the printDetails method.

**Note on Java Static Members:**

In the above exercise, the static members work by:

- **Static Variables:** The static variables bankName and interestRate are shared among all instances of the BankAccount class. They are set once and can be accessed or modified through static methods.
- **Static Methods:** The static methods getBankName, setBankName, getInterestRate, and setInterestRate are used to interact with the static variables. These methods can be called without creating an instance of the class.

- **Shared State:** The static variables hold values that are shared across all instances of the BankAccount class, providing a common state that can be accessed and modified by all instances.

### Java Code Editor: