Write a Java program to create a class called "Customer" with attributes for name, email, and purchase history. Implement methods to add purchases to the history and calculate total expenditure. Create a subclass "LoyalCustomer" that adds a discount rate attribute and a method to apply the discount.

Sample Solution:

Java Code:

Customer.java

```
// Define the Customer class
// Import necessary classes
import java.util.ArrayList;
import java.util.List;
public class Customer {
    // Attributes for the customer's name, email, and purchase history
    private String name;
    private String email;
    private List<Double> purchaseHistory;

    // Constructor to initialize the customer's name and email
    public Customer(String name, String email) {
        this.name = name;
        this.email = email;
        this.purchaseHistory = new ArrayList<>();  // Initialize purchase h
    }

    // Method to add a purchase to the purchase history
    public void addPurchase(double amount) {
        purchaseHistory.add(amount);
    }

    // Method to calculate the total expenditure
    public double calculateTotalExpenditure() {
        double total = 0;
        // Loop through each purchase and sum up the amounts
        for (double purchase : purchaseHistory) {
            total += purchase;
        }
        return total;
```

```
    }

    // Getters for the name and email attributes
    public String getName() {
        return name;
    }

    public String getEmail() {
        return email;
    }

    // Getter for the purchase history
    public List<Double> getPurchaseHistory() {
        return purchaseHistory;
    }
}
```

Explanation:

Customer Class:

- The Customer class has attributes for customer name, email, and purchase history.
- The constructor initializes with the name, email, and an empty purchase history.
- "addPurchase()" method adds a purchase amount to the purchase history.
- "calculateTotalExpenditure()" method calculates the total amount spent by summing up the purchase history.
- Getters provide access to the customer's name, email, and purchase history.

LoyalCustomer.java

```
// Define the LoyalCustomer subclass that extends the Customer class
// Import necessary classes
import java.util.ArrayList;
import java.util.List;
class LoyalCustomer extends Customer {
    // Attribute for the discount rate
    private double discountRate;

    // Constructor to initialize the LoyalCustomer with name, email, and di
```

```
    public LoyalCustomer(String name, String email, double discountRate) {
        super(name, email);  // Call the superclass constructor
        this.discountRate = discountRate;
    }

    // Method to apply the discount to a given amount
    public double applyDiscount(double amount) {
        return amount - (amount * discountRate / 100);
    }

    // Override the addPurchase method to apply the discount before adding
    @Override
    public void addPurchase(double amount) {
        double discountedAmount = applyDiscount(amount);
        super.addPurchase(discountedAmount);  // Call the superclass metho
    }

    // Getter for the discount rate
    public double getDiscountRate() {
        return discountRate;
    }

    // Setter for the discount rate
    public void setDiscountRate(double discountRate) {
        this.discountRate = discountRate;
    }
}
```

Explanation:

LoyalCustomer Subclass:

- The LoyalCustomer class extends the Customer class and adds a discountRate attribute.
- The constructor initializes the LoyalCustomer with a discount rate, in addition to the attributes inherited from Customer.
- "applyDiscount()" method calculates the discounted amount.
- "addPurchase()" method overrides the superclass method to apply the discount before adding the purchase.
- Getter and setter methods provide access to the discount rate.

Main.java

```
// Main class to test the Customer and LoyalCustomer classes
// Import necessary classes
import java.util.ArrayList;
import java.util.List;
public class Main {
    public static void main(String[] args) {
        // Create a regular customer
        Customer customer1 = new Customer("Talisha Dion", "talisha@example.
        customer1.addPurchase(200);
        customer1.addPurchase(300);
        System.out.println("Total expenditure for " + customer1.getName() +

        // Create a loyal customer with a discount rate
        LoyalCustomer loyalCustomer = new LoyalCustomer("Fulchard Sofya", "
        loyalCustomer.addPurchase(200);
        loyalCustomer.addPurchase(300);
        System.out.println("Total expenditure for " + loyalCustomer.getName
    }
}
```

Explanation:

Main Class:

- The Main class contains the main method to test the functionality.
- It creates instances of Customer and LoyalCustomer, adds purchases, and prints their total expenditure.

Output:

```
Total expenditure for Talisha Dion: 500.0
Total expenditure for Fulchard Sofya after discount: 450.0
```

Java Code Editor: