

Java code refactoring with or without ChatGPT

Survey response 1

Response ID
6
Date submitted
2024-01-13 12:26:57
Last page
13
Start language
en
Seed
811459154
Date started
2024-01-13 11:48:42
Date last action
2024-01-13 12:26:57
Total time
2301.04

Survey questionnaires (Part 1)

How old are you?
30
How many years of experience do you have with Java programming?
3
For how many years have you been programming for larger software projects e.g. in a company? Please enter a number between 0 and 30.
2
How many years of experience do you have with code refactoring?
1
Did you study programming or computer science at a university?
Yes
During your education, how many courses did you take where Java was the primary language?
4
On a scale from 1 to 5, how would you rate your Java programming expertise (e.g 1-very inexperienced, 5-very experienced)?
4
How would you compare your Java expertise to those with over 20 years of practical experience (e.g 1-very inexperienced, 5-very experienced)?
3
How would you rate your Java expertise in comparison to your peers or colleagues (e.g 1-very inexperienced, 5-very experienced)?
4

How often have you used Chat GPT (e.g 1-low, 5-high)?
3
Have you used ChatGPT for code refactoring tasks (e.g 1-low, 5-high)?
3
What is the average size of Java professional projects you typically work on, categorized as small-scale (up to 900 lines of code), medium-scale (900 to 40,000 lines of code), or large-scale (exceeding 40,000 lines of code)?
large-scale
Group time: Survey questionnaires (Part 1)
227.64

Task Explanation

<p>Tasks Overview: In each of the two sections, you will encounter five Java code snippets that require refactoring. The first five snippets must be refactored without assistance from ChatGPT, while the last five snippets can be refactored with the aid of ChatGPT. Primarily, you have two alternatives: Without Assistance: Refactor the code on your own, relying on your existing knowledge and skills. With ChatGPT Assistance: Utilize the assistance of ChatGPT to receive suggestions and guidance for refactoring the code. Timing: Each assignment must be completed within a strict time constraint of 3 minutes. You must complete the work within a 3-minute timeframe, otherwise, timeouts will occur. Efficiently allocate your time to ensure timely completion of all jobs. Instructions: Read the code: Begin by thoroughly understanding the provided Java code snippet. Refactor: Apply your refactoring skills to improve the code based on the given criteria (readability, efficiency, maintainability, etc.).</p>
Group time: Task Explanation
39.48

Question 1 for Pretest (Part 2)

<p>Refactor the below code snippet without ChatGPT within 3 minutes. <code>public double getPayAmount() { double result; if (isDead) { result = deadAmount(); } else { if (isSeparated) { result = separatedAmount(); } else { if (isRetired) { result = retiredAmount(); } else { result = normalPayAmount(); } } } return result;}</code></p> <p>// Simplify the nested conditions using the ternary operator to make the code more concise.</p> <pre>public double getPayAmount() { return isDead ? deadAmount() : isSeparated ? separatedAmount() : isRetired ? retiredAmount() : normalPayAmount(); }</pre>
Group time: Question 1 for Pretest (Part 2)
215.56

Question 2

Refactor the below code snippet without ChatGPT within 3 minutes. public class Customer { private String name; private String address; private double balance; public Customer(String name, String address) { this.name = name; this.address = address; this.balance = 0; } public void deposit(double amount) { this.balance += amount; } public void withdraw(double amount) { this.balance -= amount; } public double getBalance() { return balance; } }

//Refactor focusing the deposit and withdraw methods by combining the assignment and arithmetic operations into a single line for conciseness.

```
public class Customer {
    private String name;
    private String address;
    private double balance;

    public Customer(String name, String address) {
        this.name = name;
        this.address = address;
        this.balance = 0;
    }

    public void deposit(double amount) {
        // Combine assignment and addition in a single line
        balance += amount;
    }

    public void withdraw(double amount) {
        // Combine assignment and subtraction in a single line
        balance -= amount;
    }

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

Group time: Question 2

184.53

Question 3

Refactor the below code snippet without ChatGPT within 3 minutes. public class Customer { private String name; private String address; private double balance; public Customer(String name, String address, double initialBalance) { this.name = name; this.address = address; this.balance = initialBalance; } public void processPayment(double amount) { if (amount > balance) { throw new InsufficientFundsException(); } balance -= amount; } public void printStatement() { System.out.println("Customer name: " + name); System.out.println("Customer address: " + address); System.out.println("Customer balance: " + balance); }}

// See the inner comments of the code where the refactoring done

```
public class Customer {
    private String name;
    private String address;
    private double balance;

    public Customer(String name, String address, double initialBalance) {
        this.name = name;
        this.address = address;
        this.balance = initialBalance;
    }

    public void processPayment(double amount) {
        if (amount > balance) {
            // Throw an exception with a more descriptive message
            throw new InsufficientFundsException("Insufficient funds to process payment");
        }
        // Combine assignment and subtraction in a single line
        balance -= amount;
    }

    public void printStatement() {
        // Use System.out.printf for a formatted output
        System.out.printf("Customer name: %s\nCustomer address: %s\nCustomer balance: %.2f\n", name, address, balance);
    }
}
```

Group time: Question 3

150.3

Question 4

Refactor the below code snippet without ChatGPT within 3 minutes. public class ShippingService { public double calculateShippingCost(Order order) { double totalPrice = order.getTotalPrice(); double weight = order.getWeight(); if (totalPrice > 100) { if (weight > 10) { return totalPrice * 0.2; } else { return totalPrice * 0.05; } } else { return 0; } }}

// Used of a ternary operator to simplify the nested if-else structure, making the code more concise and readable.

```
public class ShippingService {
    public double calculateShippingCost(Order order) {
        double totalPrice = order.getTotalPrice();
        double weight = order.getWeight();

        if (totalPrice > 100) {
            // Use a ternary operator for concise conditional checks
            return (weight > 10) ? totalPrice * 0.2 : totalPrice * 0.05;
        } else {
            // Return 0 directly if totalPrice is not greater than 100
            return 0;
        }
    }
}
```

Group time: Question 4

125.45

Question 5

Refactor the below code snippet without ChatGPT within 3 minutes. import java.util.concurrent.atomic.AtomicInteger; public class Order { private String customerName; private String productName; private double price; private int orderId; private static final AtomicInteger orderIdGenerator = new AtomicInteger(1000); public Order(String customerName, String productName, double price) { this.customerName = customerName; this.productName = productName; this.price = price; this.orderId = orderIdGenerator.incrementAndGet(); } public String toString() { String nameAndPrice = customerName + ", " + String.valueOf(price); return nameAndPrice + ", " + orderId; }}

// Replaced string concatenation with String.format in the toString method for a more readable and formatted output. Also price format .3f

```
import java.util.concurrent.atomic.AtomicInteger;
```

```
public class Order {
    private String customerName;
    private String productName;
    private double price;
    private int orderId;
    private static final AtomicInteger orderIdGenerator = new AtomicInteger(1000);

    public Order(String customerName, String productName, double price) {
        this.customerName = customerName;
        this.productName = productName;
        this.price = price;
        this.orderId = orderIdGenerator.incrementAndGet();
    }

    @Override
    public String toString() {
        // Use String.format for a more readable and formatted output
        return String.format("%s,%.3f,%d", customerName, price, orderId);
    }
}
```

Group time: Question 5

219.83

Question 1 for Posttest (Part 3)

Refactor the below code snippet with ChatGPT within 3 minutes. public double getPayAmount() { double result; if (isDead) { result = deadAmount(); } else { if (isSeparated) { result = separatedAmount(); } else { if (isRetired) { result = retiredAmount(); } else { result = normalPayAmount(); } } } return result; }

```
public double getPayAmount() {
    if (isDead) {
        return deadAmount();
    }

    if (isSeparated) {
        return separatedAmount();
    }

    if (isRetired) {
        return retiredAmount();
    }

    return normalPayAmount();
}
```

Group time: Question 1 for Posttest (Part 3)
--

165.18

Question 2

Refactor the below code snippet with ChatGPT within 3 minutes. public class Customer { private String name; private String address; private double balance; public Customer(String name, String address) { this.name = name; this.address = address; this.balance = 0; } public void deposit(double amount) { this.balance += amount; } public void withdraw(double amount) { this.balance -= amount; } public double getBalance() { return balance; } }
--

<pre>public class Customer { private String name; private String address; private double balance; public Customer(String name, String address) { this.name = name; this.address = address; this.balance = 0; } public void deposit(double amount) { balance += amount; } public void withdraw(double amount) { balance -= amount; } public double getBalance() { return balance; } }</pre>
--

Group time: Question 2

36.29

Question 3

Refactor the below code snippet with ChatGPT within 3 minutes. public class Customer { private String name; private String address; private double balance; public Customer(String name, String address, double initialBalance) { this.name = name; this.address = address; this.balance = initialBalance; } public void processPayment(double amount) { if (amount > balance) { throw new InsufficientFundsException(); } balance -= amount; } public void printStatement() { System.out.println("Customer name: " + name); System.out.println("Customer address: " + address); System.out.println("Customer balance: " + balance); }}

```
public class Customer {
    private String name;
    private String address;
    private double balance;

    public Customer(String name, String address, double initialBalance) {
        this.name = name;
        this.address = address;
        this.balance = initialBalance;
    }

    public void processPayment(double amount) {
        if (amount > balance) {
            throw new InsufficientFundsException();
        }
        balance -= amount;
    }

    public void printStatement() {
        System.out.println("Customer name: " + name);
        System.out.println("Customer address: " + address);
        System.out.println("Customer balance: " + balance);
    }
}
```

Group time: Question 3

37.19

Question 4

Refactor the below code snippet with ChatGPT within 3 minutes. public class ShippingService { public double calculateShippingCost(Order order) { double totalPrice = order.getTotalPrice(); double weight = order.getWeight(); if (totalPrice > 100) { if (weight > 10) { return totalPrice * 0.2; } else { return totalPrice * 0.05; } } else { return 0; } }}

```
public class ShippingService {
    public double calculateShippingCost(Order order) {
        double totalPrice = order.getTotalPrice();
        double weight = order.getWeight();

        if (totalPrice > 100) {
            return (weight > 10) ? totalPrice * 0.2 : totalPrice * 0.05;
        } else {
            return 0;
        }
    }
}
```

Group time: Question 4

27.87

Question 5

Refactor the below code snippet with ChatGPT within 3 minutes. `import java.util.concurrent.atomic.AtomicInteger; public class Order { private String customerName; private String productName; private double price; private int orderId; private static final AtomicInteger orderIdGenerator = new AtomicInteger(1000); public Order(String customerName, String productName, double price) { this.customerName = customerName; this.productName = productName; this.price = price; this.orderId = orderIdGenerator.incrementAndGet(); } public String toString() { String nameAndPrice = customerName + "," + String.valueOf(price); return nameAndPrice + "," + orderId; }}`

`import java.util.concurrent.atomic.AtomicInteger;`

```
public class Order {
    private static final AtomicInteger orderIdGenerator = new AtomicInteger(1000);

    private String customerName;
    private String productName;
    private double price;
    private int orderId;

    public Order(String customerName, String productName, double price) {
        this.customerName = customerName;
        this.productName = productName;
        this.price = price;
        this.orderId = orderIdGenerator.incrementAndGet();
    }

    public String toString() {
        return customerName + "," + price + "," + orderId;
    }
}
```

Group time: Question 5

26.69

Interview Question (Part 4)

Can you share your experiences using ChatGPT for code refactoring? What were the specific benefits or advantages you observed during the process?

ChatGPT is faster for small code refactoring.
Give different suggestions for simple code.

In what ways did ChatGPT enhance your productivity and efficiency in completing code refactoring tasks? Please provide specific examples.

Sometimes in large projects it's necessary to refactor small methods/functions for readability. By using ChatGPT I can easily refactor those methods/functions which saves a lot of time.

Did ChatGPT help you discover new refactoring techniques or approaches that you were previously unaware of? If yes, please elaborate on these insights.

Nothing new.

How did ChatGPT contribute to the maintainability and readability of the code you produced during refactoring? Were there any notable improvements or challenges in this aspect?

Good for small code. Must be improved for large scale code.

Were there any specific challenges or limitations you encountered while using ChatGPT for code refactoring? How did you overcome them, if at all?

You can ask ChatGPT in different ways but sometimes it gives the same answer. I think it is a limitation.

In what scenarios do you believe AI assistance, like ChatGPT, is most beneficial for code refactoring? Conversely, are there situations where you think it might be less effective or not suitable at all?

For large scale code it is not suitable at all.

How does ChatGPT's performance vary depending on the complexity of the code?

It somehow give you suggestions of something:)

Would you recommend ChatGPT to other Java programmers?
For small scale code refactoring yes .
Group time: Interview Question (Part 4)
845.03