

Java code refactoring with or without ChatGPT

Survey response 1

Response ID
36
Date submitted
2024-01-31 19:36:09
Last page
13
Start language
en
Seed
402396557
Date started
2024-01-31 18:41:02
Date last action
2024-01-31 19:36:09
Total time
3313.25

Survey questionnaires (Part 1)

How old are you?
27
How many years of experience do you have with Java programming?
1
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.
0
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?
1
On a scale from 1 to 5, how would you rate your Java programming expertise (e.g 1-very inexperienced, 5-very experienced)?
2
How would you compare your Java expertise to those with over 20 years of practical experience (e.g 1-very inexperienced, 5-very experienced)?
1
How would you rate your Java expertise in comparison to your peers or colleagues (e.g 1-very inexperienced, 5-very experienced)?
3

How often have you used Chat GPT (e.g 1-low, 5-high)?
5
Have you used ChatGPT for code refactoring tasks (e.g 1-low, 5-high)?
5
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)?
small-scale
Group time: Survey questionnaires (Part 1)
192.29

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
1129.71

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> <pre>public double getPayAmount() { if (isDead) { return deadAmount(); } if (isSeparated) { return separatedAmount(); } if (isRetired) { return retiredAmount(); } return normalPayAmount(); }</pre>
Group time: Question 1 for Pretest (Part 2)
98.73

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; } }

```
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;
    }
}
```

Group time: Question 2

87.22

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); }}

```
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

140

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; } }}

```
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

129.89

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; }}

```
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() {
        return String.format("%s,%.2f,%d", customerName, price, orderId);
    }
}
```

Group time: Question 5

161.03

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();  
    } else if (isSeparated) {  
        return separatedAmount();  
    } else if (isRetired) {  
        return retiredAmount();  
    } else {  
        return normalPayAmount();  
    }  
}
```

Group time: Question 1 for Posttest (Part 3)

53.69

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; } }

```
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;  
    }  
}
```

Group time: Question 2

38.11

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

57.99

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

Group time: Question 4

145.52

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

    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() {
        StringBuilder stringBuilder = new StringBuilder();
        stringBuilder.append("Customer Name: ").append(customerName)
            .append(", Product Name: ").append(productName)
            .append(", Price: ").append(String.format("%.2f", price))
            .append(", Order ID: ").append(orderId);
        return stringBuilder.toString();
    }
}
```

Group time: Question 5

139.2

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?

Using ChatGPT for code refactoring can provide efficiency, ideation, consistency, learning opportunities, improved documentation, and some error detection, though it's crucial for developers to review and validate the suggested code.

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

It can efficiently suggest refactoring options for complex conditional statements or recommend more concise syntax for repetitive code patterns.

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

Yes, ChatGPT has been helpful in suggesting new refactoring techniques or approaches that I may not have been previously aware of. For example, it can propose alternative design patterns, or more efficient coding practices.

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?

ChatGPT contributes to code maintainability and readability during refactoring by suggesting concise and well-structured code snippets. Notable improvements include streamlined logic and formatting. However, challenges may arise in cases where the generated code requires additional context or adaptation to the specific project conventions.

Were there any specific challenges or limitations you encountered while using ChatGPT for code refactoring? How did you overcome them, if at all?
Challenges with ChatGPT in code refactoring include potential lack of context awareness and the need for careful validation. Furthermore, it can't process long codes. Overcoming involves thorough code review, testing, and adapting suggestions to project conventions.
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?
AI assistance, like ChatGPT, is beneficial in repetitive or boilerplate code scenarios for quick suggestions and ideation. It may be less effective in highly project-specific or domain-specific contexts where nuanced understanding is crucial. Not suitable for critical, safety-critical, or domain-specific tasks requiring domain expertise.
How does ChatGPT's performance vary depending on the complexity of the code?
ChatGPT's performance may vary with code complexity. It excels in providing suggestions for common or straightforward refactorings but might struggle with highly complex or domain-specific code where nuanced understanding is required.
Would you recommend ChatGPT to other Java programmers?
Recommendation for Java programmers using ChatGPT depends on specific needs. It can be beneficial for quick suggestions and ideation but requires careful validation, especially for complex or domain-specific tasks.
Group time: Interview Question (Part 4)
939.87