Java code refactoring with or without ChatGPT

Survey response 1

Response ID
14

Date submitted
2024-01-19 10:18:47

Last page
13

Start language
en

Seed
504564724

Date started
2024-01-19 09:52:20

Date last action
2024-01-19 10:18:47

Total time
1586.36

Survey questionnaires (Part 1)

How old are you?
29

How many years of experience do you have with Java programming?
8

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.
6

How many years of experience do you have with code refactoring?
6

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?
2

On a scale from 1 to 5, how would you rate your Java programming expertise (e.g 1-very inexperienced, 5-very experienced)?
3

How would you compare your Java expertise to those with over 20 years of practical experience (e.g 1-very inexperienced, 5-very experienced)?
4

How would you rate your Java expertise in comparison to your peers or colleagues (e.g 1-very inexperienced, 5-very experienced)?
2

How often have you used Chat GPT (e.g 1-low, 5-high)?
4

Have you used ChatGPT for code refactoring tasks (e.g 1-low, 5-high)?
4

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)
151.94

Task Explanation

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.).

Group time: Task Explanation
25.7

Question 1 for Pretest (Part 2)

Refactor the below code snippet without 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 Pretest (Part 2)
150.76

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) {
        if (amount > 0) {
            balance += amount;
        } else {
            System.out.println("Deposit amount must be positive.");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0) {
            if (amount
```

Group time: Question 2

179.16

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
167.29

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
113.97

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; }}

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
106.03

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)
38.47

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) {
  if (amount > 0) {
  balance += amount;
  } else {
  System.out.println("Deposit amount must be positive.");
  }
  }
 	
  public void withdraw(double amount) {
  if (amount > 0) {
  if (amount

Group time: Question 2
50.03

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

35.42

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

29.34

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();
  }
 	
  @Override
  public String toString() {
  return String.format("%s,%.2f,%d", customerName, price, orderId);
  }
}

Group time: Question 5
30.5

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 can quickly provide code refactoring suggestions, ensuring consistency, identifying errors, reducing complexity and offering alternative approaches.

In what ways did ChatGPT enhance your productivity and efficiency in completing code refactoring tasks? Please provide specific examples.
ChatGPT enhances code refactoring by quickly suggesting improvements. For example it can recommend clearer variable names, identify redundant code or propose more efficient algorithms. This accelerates the process and aids developers in producing cleaner and optimized code.

Did ChatGPT help you discover new refactoring techniques or approaches that you were previously unaware of? If yes, please elaborate on these insights.
ChatGPT can offer insights into new refactoring techniques, suggesting alternative design patterns, more efficient algorithms or lesser-known best practices. We have found these suggestions valuable for expanding their coding knowledge.

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 by suggesting improvements such as clearer variable names and better code organisation during refactoring. However, users should assess suggestions for alignment with project requirements.

Were there any specific challenges or limitations you encountered while using ChatGPT for code refactoring? How did you overcome them, if at all?
Users have reported potential challenges with ChatGPT such as generating code not adhering to project standards. To address this, they perform manual review, seek clarification and ensure alignment with project requirements.

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 for quick suggestions, exploring alternatives and learning opportunities in code refactoring. It may be less effective in extremely complex logic or safety-critical systems where human expertise is crucial. Human judgement is essential for careful review and ensuring alignment with project standards.

How does ChatGPT's performance vary depending on the complexity of the code?
ChatGPT's performance can vary based on the complexity of the code. It generally excels in providing assistance for common coding tasks, suggesting improvements and offering insights into simpler or moderately complex code. However, its effectiveness may decrease when dealing with highly intricate or project-specific logic where a deep understanding of context is required. Users should be mindful of the complexity of their code and may need to provide additional context or clarification for optimal results.

Would you recommend ChatGPT to other Java programmers?
Yes. I can recommend ChatGPT to Java programmers as a valuable tool for generating code snippets.

Group time: Interview Question (Part 4)
507.75