Java code refactoring with or without ChatGPT

Survey response 1

Response ID
5

Date submitted
2024-01-12 22:46:17

Last page
13

Start language
en

Seed
18576555

Date started
2024-01-12 22:17:25

Date last action
2024-01-12 22:46:17

Total time
1737.8

Survey questionnaires (Part 1)

How old are you?
28

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.
4

How many years of experience do you have with code refactoring?
3

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)?
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)?
1

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

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)?
40000

Group time: Survey questionnaires (Part 1)
122.93

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
68.15

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() {
double result;
if (isDead) {
result = deadAmount();
} else {
if (isSeparated) {
result = separatedAmount();
} else {
if (isRetired) {
result = retiredAmount();
}
result = normalPayAmount();
}
}
return result;
}

Group time: Question 1 for Pretest (Part 2)
85.9

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 = 0.0;
  }
 	
  public void deposit(double amount) {
 	
  this.balance += amount;
  }
 	
  public void withdraw(double amount) {
  this.balance -= amount;
  }
 	
  public double getBalance() {
  return balance;
  }
 	
 }

Group time: Question 2
103.89

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 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

172.06

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) {
            if (weight > 10) {

                return totalPrice * 0.2;
            }
                return totalPrice * 0.05;
        }
            return 0;
    }
}
```

Group time: Question 4

77.49

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 final static 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;
    }
}
```

Group time: Question 5

145.85

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() {
  double result;
  if (isDead) {
    result = deadAmount();
  } else {
    if (isSeparated) {
      result = separatedAmount();
    } else {
      if (isRetired) {
        result = retiredAmount();
      }
        result = normalPayAmount();
    }
  }
  return result;
}
```

Group time: Question 1 for Posttest (Part 3)

98.99

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) {
  this.balance += amount;
  } else {
  System.out.println("Invalid deposit amount. Amount must be greater than 0.");
  }
  }
 	
  public void withdraw(double amount) {
  if (amount > 0) {
  if (amount

Group time: Question 2
51.06

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);
    }

    // Additional methods for getting and setting name, address, and balance
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public double getBalance() {
        return balance;
    }

    public void setBalance(double balance) {
        this.balance = balance;
    }
}
```

Group time: Question 3

32.75

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 {
  private static final double DISCOUNT_RATE_HIGH_TOTAL_PRICE = 0.2;
  private static final double DISCOUNT_RATE_LOW_TOTAL_PRICE = 0.05;
  private static final double MIN_TOTAL_PRICE_FOR_DISCOUNT = 100;
  private static final double MAX_WEIGHT_FOR_LOW_DISCOUNT = 10;
 	
  public double calculateShippingCost(Order order) {
  double totalPrice = order.getTotalPrice();
  double weight = order.getWeight();
 	
  if (isEligibleForHighDiscount(totalPrice, weight)) {
  return calculateDiscountedShippingCost(totalPrice, DISCOUNT_RATE_HIGH_TOTAL_PRICE);
  } else if (isEligibleForLowDiscount(totalPrice)) {
  return calculateDiscountedShippingCost(totalPrice, DISCOUNT_RATE_LOW_TOTAL_PRICE);
  } else {
  return 0;
  }
  }
 	
  private boolean isEligibleForHighDiscount(double totalPrice, double weight) {
  return totalPrice > MIN_TOTAL_PRICE_FOR_DISCOUNT && weight > MAX_WEIGHT_FOR_LOW_DISCOUNT;
  }
 	
  private boolean isEligibleForLowDiscount(double totalPrice) {
  return totalPrice > MIN_TOTAL_PRICE_FOR_DISCOUNT;
  }
 	
  private double calculateDiscountedShippingCost(double totalPrice, double discountRate) {
  return totalPrice * discountRate;
  }
}

Group time: Question 4
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 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();
  }
 	
  public String getCustomerName() {
  return customerName;
  }
 	
  public String getProductName() {
  return productName;
  }
 	
  public double getPrice() {
  return price;
  }
 	
  public int getOrderId() {
  return orderId;
  }
 	
  @Override
  public String toString() {
  return String.format("Customer: %s, Product: %s, Price: %.2f, Order ID: %d",
  customerName, productName, price, orderId);
  }
}

Group time: Question 5
37.13

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 might suggest possible improvements, refactorings, or cleaner code structures.

In what ways did ChatGPT enhance your productivity and efficiency in completing code refactoring tasks? Please provide specific examples.
ChatGPT can help generate ideas for refactoring or optimizing code. If we encounter unfamiliar code patterns or programming concepts during refactoring, ChatGPT can provide explanations and clarify doubts.

Did ChatGPT help you discover new refactoring techniques or approaches that you were previously unaware of? If yes, please elaborate on these insights.
no

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?
By using ChatGPT to explain complex code sections or refactorings, one can produce comments, documentation, or inline explanations that enhance the readability of the code. it also can provides suggestions for refactoring code to make it more readable and maintainable.

Were there any specific challenges or limitations you encountered while using ChatGPT for code refactoring? How did you overcome them, if at all?
no

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 is most beneficial for routine tasks, idea generation, and quick feedback in code refactoring. However, it's important to combine AI suggestions with human expertise, especially in scenarios that involve domain-specific knowledge, critical decisions, or real-time understanding of the codebase.

How does ChatGPT's performance vary depending on the complexity of the code?
For somewhat complicated code snippets, ChatGPT performs well, providing explanations, recommendations, and insights.

Would you recommend ChatGPT to other Java programmers?
Till some extend. Such as, when learning, exploring, and getting rapid insights for Java programming projects, developers can use ChatGPT as a helpful tool.

Group time: Interview Question (Part 4)
707.6