**first and last name: Phan Tấn Doanh**

**MSSV:28201127971**

**Subject:**

**-Why is code refactoring important?**

+Refactoring code improves the readability and maintainability of the code, making it easier to understand and change. This makes it easier to detect and fix errors, as well as facilitate adding new features.

**-What are the challenges of code refactoring?**

+Some challenges include ensuring that refactored code remains functional, avoiding breaking other parts of the system, and requiring thorough testing to confirm that there is no illegal behavior. What wishes appear after restructuring.

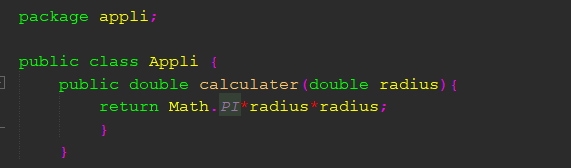
**-When should you avoid code refactoring?**

+Refactoring should be avoided when the code is stable and does not need to change, or when the costs and risks of refactoring are not worth the benefits it brings.

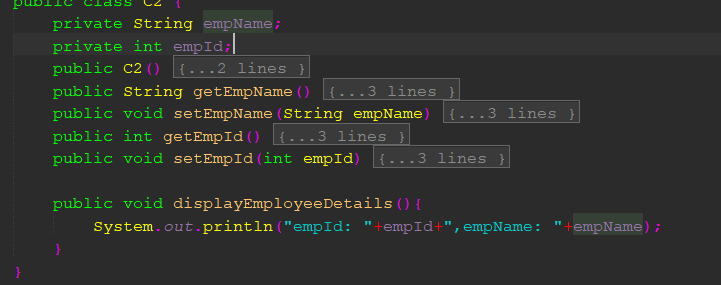
**-When is the ideal time to refactor your code?**

+The ideal time to refactor is when you realize the code could be improved to increase maintainability, or when you need to add new features and realize that refactoring will make the process easier. easier.

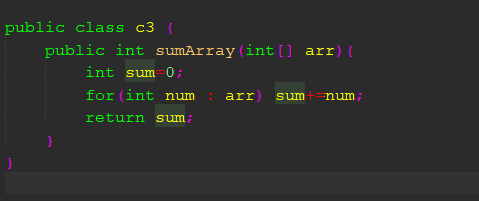
**C1:**

****

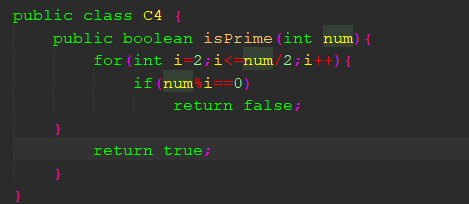
**C2:**

****

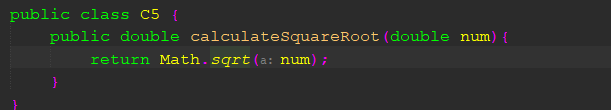
**C3:**



**C4:**

****

**C5:**

****

**Exercise 1:**

1.Note working days:

-This requirement includes storing information about the user's day and work schedule.

-There should be a user interface so users can set up and edit their work schedules.

+ Estimated time: 1-2 weeks.

+Resources needed: 1 developer, 1 designer.

2. Schedule appointments and reminders:

-This requirement requires the feature to schedule appointments and send reminders to users before the appointment time.

-There should be a user interface so users can add, modify, and delete appointments.

-A database is needed to store information about users' appointments.

+ Estimated time: 3 weeks to 5 weeks.

+Resources needed: 2 developer, 1 designer.

3.Activity tracking and time statistics:

-This requirement includes tracking user activity and providing statistics on time spent using the application.

-A database is needed to store information about user activities.

-It is necessary to develop functions to analyze data and generate statistical reports.

+ Estimated time: 4 weeks to 6 weeks.

+Resources needed: 2 developer, 1 data analyst.

**Exercise 2**

1.Product information management:

-This requirement includes storing and managing product information, including product name, description, price, inventory quantity, and other attributes.

-Need to develop a user interface for product management, including adding, modifying, deleting products and viewing product lists.

-Need a database to store product information.

+ Estimated time: 3 weeks

+Required resources: 1 Product Manager, 1 Backend Developer, 1 Frontend Developer, 1 Database Administrator

2.Processing orders from customers:

-This requirement includes storing and managing order information from customers, including information about the product ordered, quantity, price, and shipping information.

-Need to develop a user interface so customers can place orders and view the status of their orders.

-Need to integrate order confirmation and processing system, including updating order status and storing order history.

+ Estimated time: 3 weeks

+Required resources: 1 Product Manager, 1 Backend Developer, 1 Frontend Developer, 1 Database Administrator

3.Track payment status:

-This requirement includes tracking the payment status of orders and notifying users of payment status.

-Need to develop functions to update payment status and send notifications to customers.

-Need to integrate electronic payment gateways or payment systems to process payments from customers.

+ Total estimated time: 9 weeks

+Total resources needed: 1 Product Manager, 1 Business Analyst, 3 Backend Developers, 2 Frontend Developers, 1 Database Administrator, 2 Quality Testers