Hướng dẫn Assignment 1

Assignment 1 sẽ gồm 2 phần: slide thuyết trình + demo code và viết

Part 1: Làm slide

1. **Tạo đặc tả thiết kế cho cấu trúc dữ liệu và giải thích những hoạt động hợp lệ có thể được thực hiện trên cấu trúc (P1)**

* ADT là gì? Vai trò của ADT

Trước khi 1 chương trình đc viết, lập trình viên phải biết chương trình thực hiện những nhiệm vụ nào? Ngay từ đầu phải xác định rõ đầu vào và đầu ra của từng nhiệm vụ. Đồng thời quyết định sử dụng cấu trúc dữ liệu nào để hiệu quả nhất về thời gian và không gian. 1 items mà đc chỉ định các hoạt động gọi là kiểu dữ liệu trừu tượng. Trong Java, ADT có thể là 1 phần của chương trình dưới dạng interface

* Ví dụ về ADT:
* Khi cài đặt một ADT trên một ngôn ngữ cụ thể, các công việc chúng ta cần làm?

Ví dụ :

Bước 1: Biểu diễn ADT với cấu trúc dữ liệu đã được cài đặt hoặc một ADT khác

Bước 2: Viết chương trình con để thực hiện các thao tác trên ADT

1. **Kiểu dữ liệu ngăn xếp là gì? Các hoạt động trên kiểu dữ liệu ngăn xếp (P2)**

Mô tả từng hoạt động trên kiểu dữ liệu ngăn xếp

Push()

Pop()

Peek()

Size()

isEmpty()

isFull()

Ví dụ push():

Bước 1: Kiểm tra xem ngăn xếp có đầy hay không?

Bước 2:

* Nếu đầy thì hiển thị lỗi
* Nếu chưa đầy thì tăng con trỏ top trỏ đến vị trí trống tiếp theo

Bước 3: Thêm phần tử mới vào vị trí mà con trỏ top đang trỏ đến

Bước 4: Thông báo thành công

1. **Chỉ định kiểu dữ liệu trừu tượng cho ngăn xếp phần mềm (P3)**

Một stack có thể được triển khai sử dụng Mảng hoặc LinkedList

* Các bước triển khai kiểu dữ liệu ngăn xếp sử dụng mảng (chụp lại code demo)
* Các bước triển khai kiểu dữ liệu ngăn xếp sử dụng linkedlist (chụp lại code demo)

1. Kiểu dữ liệu hàng đợi là gì? Minh họa với 1 ví dụ cụ thể về kiểu dữ liệu hàng đợi (FIFO) (M1)

* Ghi định nghĩa
* Ghi rõ các bước cụ thể của enqueue và dequeue
* Minh họa (chụp ảnh code và output)

1. 2 thuật toán sắp xếp (sắp xếp nổi bọt và sắp xếp chọn) (M2)

* So sánh 2 thuật toán này
* Đánh giá độ phức tạp của 2 thuật toán

1. Giải thích ưu điểm khi sử dụng tính đóng gói và ẩn thông tin trong ADT (M3)

Nó cải thiện khả năng bảo trì của một ứng dụng.

Cung cấp sự linh hoạt cho người dùng để sử dụng hệ thống rất dễ dàng

Giúp các nhà phát triển tổ chức mã tốt hơn

Làm cho quá trình viết mã tổng thể dễ dàng hơn, vì bạn chỉ quan tâm đến

những gì lớp khác làm, không phải nó làm như thế nào

Phương pháp này giúp các nhà phát triển 'khách quan' hơn và định hướng kết quả.

Mã đóng gói khá linh hoạt và dễ thay đổi bằng mã mới

các yêu cầu.

Đóng gói giúp kiểm tra đơn vị dễ dàng

1. 2 thuật toán tìm đường đi ngắn nhất (Dijkstra và Bellman ford) (D1)
2. Thảo luận về quan điểm cho rằng ADT bắt buộc là cơ sở để hướng đối tượng(D2)

Buổi sau:

* Hạn nộp bài trên classroom để chữa là 23h ngày 28/6
* Thứ 5 (1/7) Thuyết trình assignment 1 và demo code.
* 23h50 thứ 5 (1/7) hạn cuối nộp assignment, nộp powerpoint chuyển sang định định dạng pdf (trang đầu chụp ảnh lại frontsheet ghi rõ thông tin họ tên, mã sv, giáo viên …)
* Lưu ý: Trang cuối phải có tài liệu tham khảo