Xin chào Cô và mọi người đã đến với buổi thuyết trình của mình về assignment 1 môn Cấu trúc dữ liệu và giải thuật. Trên đây là những thông tin cá nhân của mình, nếu như mọi người có ý kiến hay có vấn đề gì cần trao đổi thì có thể liên hệ với mình sau buổi thuyết trình ạ (chuyển slide)

Nội dung hôm nay sẽ bao gồm 6 phần cả phần P và M nên sẽ hơi bị dài, mong mọi người thông cảm.

Mình sẽ nói sơ qua những nội dung sẽ trình bày trong ngày hôm nay

* P1. Tạo đặc tả thiết kế cho cấu trúc dữ liệu?
* P2 là nói về ngăn xếp?
* P3 là Chỉ định kiểu dữ liệu trừu tượng cho ngăn xếp phần mềm?
* M1 mình sẽ nói về hàng đợi?
* M2 là các thuật toán sắp xếp và so sánh
* M3 là mình sẽ giải thích các ưu điểm của tính động gói và ẩn thông tin trong ADT. Và sau đây, mình xin đc thuyết trình, uống cốc nước đã
* Về M1, M2 và M3 thì cô(chuyển slide)

**I. Trước tiên chúng ta làm rõ trừu tượng hóa dữ kiệu là gì?**

**1. Trừu tượng hóa dữ liệu** là việc rút gọn một phần **d**ữ liệu cụ thể thành một đại diện đơn giản của toàn bộ. Nói chung, trừu tượng là quá trình loại bỏ hoặc loại bỏ các đặc điểm từ một cái gì đó để giảm nó thành một tập hợp các đặc điểm thiết yếu.

**Ví dụ** trừu tượng hóa phương thức trong OOP như C ++ có thể sử dụng các phương thức (được định nghĩa trước) mà không cần quan tâm đến cách chúng hoạt động bên trong Ok (chuyển slide)

**2. Vậy kiểu dữ liệu trừu tượng là gì?**

Kiểu dữ liệu trừu tượng (viết tắt là: **ADT**) là đặc điểm kỹ thuật của kiểu dữ liệu trong một số ngôn ngữ lập trình, độc lập với việc triển khai.

Giao diện cho ADT được xác định theo kiểu và tập hợp các thao tác trên kiểu đó.

Hành vi của mỗi hoạt động được xác định bởi đầu vào và đầu ra của nó.

ADT không xác định y như cách thức triển khai kiểu dữ liệu. Các chi tiết triển khai này được ẩn với người dùng ADT và được bảo vệ khỏi sự truy cập từ bên ngoài, mà người ta gọi đó là sự Đóng gói. (chuyển slide)

**3. Về những lợi ích của ADT là**

Đối với Nhà sản xuất (người tạo ADT) được lợi ích như sau:

Dễ dàng sửa đổi, bảo trì

Có lợi nhuận

Có thể tái sử dụng

Đối với Khách hàng (người sử dụng ADT) được hưởng lợi:

sử dụng đơn giản, dễ hiểu

Quen thuộc với các giao diện đã từng dùng

Giá rẻ (chuyển slide)

**4. Đối với lập trình viên thì nên làm những gì khi thực thi 1 kiểu dữ liệu trừu tượng**

Dưới đây là quy trình của một ADT từ xác định cho tới triển khai

1. Định nghĩa vấn đề

2. Xác định các kiểu dữ liệu trừu tượng như là định nghĩa dữ liệu hay thuộc tính

3. Chỉ định hoạt động ADT

4. Chỉ định các tương tác ADT

**Đồng thời Xác định hệ thống phân cấp đối tượng (nếu sử dụng OOP)**

Và cuối cùng là triển khai chúng (chuyển slide)

**5. Thế kiểu dữ liệu trừu tượng trong Java thì ntn**

**Thư viện Java** có các kiểu dữ liệu trừu tượng như Danh sách, Ngăn xếp, Hàng đợi, Tập hợp, Map như các giao diện có sẵn đang được thực hiện bằng cách sử dụng các cấu trúc dữ liệu khác nhau.

JDK không cung cấp bất kỳ triển khai trực tiếp nào của giao diện này. Nó cung cấp triển khai các giao diện con cụ thể hơn như List, Set. Giao diện này thường được sử dụng để chuyển các bộ sưu tập xung quanh và thao tác chúng ở những nơi mong muốn có tính tổng quát tối đa.(chuyển slide)

Chúng ta đến với phần cuối cùng của P1 đó là quyết định sử dụng cấu trúc dữ liệu nào để hiệu quả nhất.

* Việc quyết định sử dụng cấu trúc dữ liệu nào để hiệu quả nhất về thời gian và không gian.

Thì ở đây tôi khuyên mọi người sử dụng Cấu trúc dữ liệu hàng đợi

Hàng đợi thường được ứng dụng làm bộ đệm máy tính để lưu lại những lênh mà máy chưa kịp sử lý. Và sau đó nó sẽ được xử lý thao trình tự FIFO. ví dụ như trường hợp của bộ đệm bàn phím, phím nào nhấn trước sẽ được xử lý trước

Nó còn đc ứng dụng trong Sản xuất và tiêu thụ (ứng dụng trong các hệ điều hành song song).

Xử lý các lệnh trong máy tính (ứng dụng trong hệ điều hành, trình biên dịch), hàng đợi các tiến trình chờ được xử lý. (chuyển slide)

**Bây giờ phần P2, đó là tìm hiểu về kiểu dữ liệu ngăn xếp**

Định nghĩa về ngăn xếp như sau: Ngăn xếp là một cấu trúc dữ liệu tuyến tính chỉ có thể được truy cập ở một trong các đầu của nó để lưu trữ và truy xuất dữ liệu. Ví dụ về một chồng đĩa: Đĩa cuối cùng được đặt sau cùng sẽ bị loại bỏ đầu tiên của ngăn xếp đó.

Vì lý do này, một ngăn xếp được gọi là Last in/First out tức là Vào sau cùng / Ra đầu tiên (chuyển slide)

**Các hoạt động trong Stack**

Các hoạt động cơ bản trên ngăn xếp bao gồm như là:

* isEmpty () - Kiểm tra xem ngăn xếp có trống không.
* isFull **()** — Check to see if the stack is full
* Push pớt (phần tử) - Đặt phần tử lên trên cùng của ngăn xếp.
* pop () - Lấy phần tử trên cùng từ ngăn xếp.
* peek() - Trả về phần tử trên cùng trong ngăn xếp mà không cần xóa nó.

Nhưng trong Ngăn xếp thường thì chỉ có 2 hoạt động chính đó là Push và Pop. Để làm rõ hơn về những cách hoạt động này, chúng ta đến với ví dụ về nó.

(Chuyển slide)

**ĐẦU TIÊN LÀ PUSH**

Quá trình đưa một phần tử dữ liệu mới vào ngăn xếp được gọi là Hoạt động Đẩy. Hoạt động này bao gồm một loạt các bước như sau

**Bước 1** là check xem thử ngăn xếp đã đầy hay chưa

**Bước 2** Nếu ngăn xếp đầy rồi thì báo lỗi

**Bước 3** Nếu ngăn xếp chưa đầy, thì tăng dần **TOP** để trỏ đến không gian trống tiếp theo.

**Bước 4** Thêm phần tử dữ liệu vào vị trí ngăn xếp, nơi Top đang được trỏ

Và bước 5 là trả về thành công (chuyển slide)

**TIẾP THEO LÀ HOẠT ĐỘNG POP**

Truy cập nội dung trong khi xóa nó khỏi ngăn xếp, được gọi là Thao tác POP. Trong quá trình triển khai hoạt động POP() Đối với mảng thì phần tử đó không thực sựu bị xóa còn hoạt động POP() đối với danh sách liên kết thì phần tử đó bay màu luôn.

Hoạt động Pop có thể bao gồm các bước sau:

**Bước 1** - Kiểm tra xem ngăn xếp có trống không.

**Bước 2** - Nếu ngăn xếp trống, tạo ra lỗi.

**Bước 3** - Nếu ngăn xếp không trống, thì truy cập phần tử dữ liệu mà ở **TOP** đang trỏ đến.

**Bước 4** - Giảm giá trị của **Top** đi 1.

**Và cuối cùng** Trả về thành công

Nói chung, ngăn xếp rất hữu ích trong các tình huống khi dữ liệu phải được lưu trữ và sau đó được truy xuất theo thứ tự ngược lại.

Những ứng dụng thực tế của ngăn xếp như là

**Đánh giá biểu thức và phân tích cú pháp**

**Biên dịch quản lý bộ nhớ thời gian**(chuyển slide)

**Chúng ta qua phần P3**

Đối với stack chúng ta có thể triển khai sử dụng Mảng hoặc LinkedList

* Đối với triển khai với 1 mảng thì chúng ta có thể làm đơn giản như sau (click)
* Mình sẽ tạo 1 project có 3 lớp main, Node và Mystack
* Lớp MyStack có những hàm nào (Click) Có các thuộc tính này, hàm khởi tạo, hàm check đầy và rỗng, hàm push hàm pop và 1 hàm hiển thị (Chuyển slide)
* Lớp Node có những hàm nào (Click) Có các thuộc tính này, hàm khởi tạovà 1 hàm hiển thị
* Ở lớp main mình đã cho 3 cái data nguyen huu hoang push chúng vào ngăn xếp – sau đó hiển thị chúng ra (chuyển slide)

Kết quả khi chạy chương trình

Khi ta push 3 cái node này thì kết quả sẽ trả về đúng 3 cái node này

Còn khi ta pop đi 1 thì phần tử cuối cùng của mảng sẽ bị loại bỏ

Đó là chúng ta sử dụng mảng, Bây giờ chúng ta triển khai Ngăn xếp với 1 LinkedList thì sẽ ntn? Chúng ta cùng qua phần sau (chuyển slide)

Với ví dụ này thì mình có tất cả 4 lớp đó là: Main, Node, MyStack và Student

* Đầu tiên là lớp Student, mình cho 3 thuộc tính id tên tuổi, 1 hàm constructor và 1 hàm trả về khi hiển thị thông tin
* Lớp tiếp theo đó là lớp Node – làm việc với danh sách liên kết thì ta cần 1 cái Node next để trỏ tới cái móc tiếp theo (chuyển slide)

Tiếp tục ở lớp MyStack.java mình tạo 1 Node head và cho nó = null, với danh sách liên kết thì chúng ta chỉ càn check rỗng chứ Không cần check nó đầy hay Không

Với hàm Push làm việc với danh sách liên kết thì chúng ta lấy móc của node mới móc vào head XONG cho head trỏ vào cái node mới

Với hàm POP thì check xem có rỗng hay Không? Rồi lấy 1 node tạm giữ lấy head SAU đó cho head trỏ vào node tiếp theo

Hàm hiện thị thì nó cũng tương tự như trước kia mà chúng ta đã từng làm. Chúng ta sẽ có 1 nút hiện tại = head. Nếu Nút hiện tại Không Null thì sẽ in ra lần lượt các phần tử có trong mảng.

Sau cùng là lớp MAIN, trong lớp main chúng ta sẽ triển khai chương trình bằng Switch case. Có 4 trường hợp cho option khác nhau như:

Sau cùng là lớp MAIN, trong lớp main chúng ta sẽ triển khai chương trình bằng Switch case. Có 4 trường hợp cho option khác nhau như:

Thêm, hiển thị, xóa thông tin và thoát chương trình. Đây là kết quả khi triển khai chương trình (chuyển slide)

Mình cho 2 đối tượng đó là nguyễn hữu hoàng và nguyễn anh huy, sau đó bấm chức năng 2 để hiển thị. Còn khi remove thì phần tử cuối cùng sẽ bị xóa (chuyển slide)

THẾ LÀ MÌNH VỪA KẾT THÚC PHẦN P BÂY GIỜ XIN MỜI MỌI NGƯỜI ĐẾN VỚI THUYẾT TRÌNH PHẦN M CỦA MÌNH

M1 mình sẽ tổng quan về Hàng đợi

KN về Hđ: Hàng đợi là một cấu trúc dữ liệu dùng để chứa các đối tượng làm việc theo cơ chế FIFO (viết tắt của: First In First Out), nghĩa là "vào trước ra trước"

Trong hàng đợi, các đối tượng có thể được thêm vào hàng đợi bất kỳ lúc nào, nhưng chỉ có đối tượng thêm vào đầu tiên mới được phép lấy ra khỏi hàng đợi. Thao tác thêm vào và lấy một đối tượng ra khỏi hàng đợi được gọi lần lượt là "enqueue" và "dequeue". Việc thêm một đối tượng luôn diễn ra ở cuối hàng đợi và một phần tử luôn được lấy ra từ đầu hàng đợi (chuyển slide)

Tương tự như ngăn xếp, hàng đợi hỗ trợ các thao tác:

EnQueue(o): thêm đối tượng vào cuối hàng đợi.

DeQueue(): lấy đối tượng ở đầu ra khỏi hàng đợi và trả về giá trị của nó. Nếu hàng đợi rỗng thì lỗi sẽ xảy ra.

IsEmpty(): kiểm tra xem hàng đợi có rỗng không.

isFull(): kiểm tra hàng đợi đã đầy chưa

Front(): trả về giá trị của phần tử nằm ở đầu hàng đợi mà không hủy nó. Nếu hàng đợi rỗng thì lỗi sẽ xảy ra.

Các thao tác thêm và xóa một phần tử phải được thực hiện ở hai phía khác nhau của hàng đợi, do đó hoạt động của hàng đợi được thực hiện theo nguyên tắc FIFO.

Cũng như ngăn xếp, cấu trúc mảng một chiều hoặc cấu trúc danh sách liên kết có thể dùng để biểu diễn cấu trúc hàng đợi. Hôm nay thì mình sẽ dùng mảng để biểu diễn cấu trúc hàng đợi, mọi người xem mình demo trẻn máy nhá (out, demo tren IDE

hãy nhớ tab qua lại giữ word và intelliJ)

Lớp student bao gồm các thuộc tính id tên tuổi, hàm khởi tạo và hàm trả về

Lớp Node có 1 thuộc tinh do là Student data và 1 hàm khởi tạo đó

Trong lớp MyQueue có các thuộc tính như first, last, size v.v

Tại hàm constructor ta truyền tham số int maxSize, và cho first, last đều = 0, size cũng = 0, maxSize = maxSize và queue = 1 mảng Node có chứa maxSize

Đối với hàm check rỗng và check đầy. Rỗng khi kích thước của hàng đợi đó = 0, và đầy khi nó là maxSize

Bây giờ là 2 nhân vật chỉnh enqueue và dequeue

- Tại hàm enqueue tham số truyền vào là 1 node. Chúng ta sẽ check xem hàng đợi đó đầy chưa, nếu rồi thì báo đã đầy và Không thể nhập tiếp

Chưa đầy thì phần tử cuối của hàng đợi sẽ bằng 1 node mới đẩy vào, hàng đợi sẽ tăng lên 1

* Tại hàm dequeue thì ta check rỗng, nếu rỗng thì báo rỗng CÒN trong hàng đợi mà Không rỗng thì chương trình sẽ thực hiện xóa phần tử đầu tiên.
* First ++ là vì phần tử sau sẽ thay thế phần tử trước, đồng nghĩa kích thước của hàng đợi đó bị giảm đi 1
* Cuối cùng là hàm Display(). Ta có vòng lặp for cho I = first ; I < maxSize và I ++. Sau đó cho hiển thị các dữ liệu có trong hàng đợi ra màn hình

Khởi chạy chương trình thì chắc chắn cần có lớp main rồi :D

Trong lớp Main này mình dùng switch case, cái này thì quen thuộc quá với lớp mình rồi, và bây giờ chạy thử chương trình xem nó chạy ổn Không nha (chạy thử chuong trình)

Vậy là xong M1, M2 của mình là so sánh 2 loại giải thuật sắp xếp, nổi bọt và sắp xếp chọn

Định ngĩa về Giải thuật sắp xếp

Sắp xếp là sắp xếp dữ liệu theo một định dạng cụ thể như theo thứ tự các chữ cái tăng/giảm dần, theo thứ tự số tăng/giảm dần. Trong khoa học máy tính, giải thuật sắp xếp xác định cách để sắp xếp dữ liệu theo một thứ tự nào đó. Sắp xếp theo thứ tự ở đây là sắp xếp theo thứ tự dạng số hoặc thứ tự dạng chữ cái như trong từ điển chuyển slide

Sắp xếp nổi bọt trong Java?

Giải thuật này dựa trên tính chất đó là cái gì nhẹ thì sẽ nổi lên và cái gì nặng thì nó chìm xuống, nghe có vẻ đơn giản đúng Không (chuyển slide)

Tiếp theo là sắp xếp chọn selection sort

Sắp xếp chọn trong Java

Sắp xếp lựa chọn là [một thuật toán sắp xếp](https://www.programiz.com/dsa/sorting-algorithm) chọn phần tử nhỏ nhất từ ​​danh sách chưa được sắp xếp trong mỗi lần lặp và đặt phần tử đó ở đầu danh sách chưa được sắp xếp. (chuyển slide)

Ý tưởng như sau về sắp xếp nổi bọt như sau:

Đi từ đầu đến cuối mảng, so sánh phần tử thứ i và j, nếu phần tử thứ j nhỏ hơn phần tử thứ i thì ta hoán vị hai phần tử này.

Cứ như vậy ta so sánh từng cặp cho đến hết mảng. Ý tưởng là như vậy (chuyển slide)

Ưu điểm chính của Bubble Sort là tính đơn giản của thuật toán.

Độ phức tạp không gian cho Sắp xếp nổi bọt là O (1) , vì chỉ cần một không gian bộ nhớ bổ sung duy nhất, tức là cho biến "swap".

Ngoài ra, độ phức tạp thời gian của trường hợp tốt nhất sẽ là O (n) , đó là khi danh sách đã được sắp xếp. (chuyển slide)

Về Ý tưởng của Selection sort là ta tìm từng phần tử cho mỗi vị trí của mảng hoán vị cần tìm.

**Thuật toán:**

* Tìm phần tử nhỏ nhất đưa vào vị trí 1
* Tìm phần tử nhỏ kế tiếp đưa vào vị trí 2
* Tìm phần tử nhỏ tiếp theo đưa vào vị trí 3 cứ thế cho đến hết (chuyển slide)

Chúng ta sẽ triển khai bằng vòng lặp for. Cho vòng lặp for duyệt qua từng phần tử của mảng

Sau đó tìm phần tử nhỏ nhất trong mảng chưa được sắp xếp

Tiếp theo chúng ta sẽ Hoán đổi phần tử nhỏ nhất và phần tử đầu tiên (chuyển slide)

Hoán đổi xong thì dùng hàm print để in mảng đó ra màn hình

Trong hàm main sẽ có 2 chức năng đó là in ra mảng ban đầu và in ra kết quả sau khi các ptu đc sắp xếp (chuẩn bị mà bấm)

Các phần đã cho trước đó là 99, 1, 15, 30, 21 sau khi in ra đó là 1, 15, 21, 30, 99(chuyể slide)

Về Sự phức tạp của thuật toán này

Sắp xếp lựa chọn mất O (n ^ 2)O ( n2) thời gian và O (1)O ( 1 ) không gian.

Chi phí thời gian chính đến từ việc quét qua mảng để tìm mục nhỏ nhất tiếp theo. Chúng ta làm điều đón n lần. Lần đầu tiên, chúng ta sẽ xem xét n các yếu tố, lần sau nó sẽ n - 1 các phần tử, v.v., cho đến khi chúng ta chỉ còn lại một phần tử.

Ngay cả khi đầu vào đã được sắp xếp, sắp xếp lựa chọn vẫn liên quan đến việc quét liên tục tất cả các phần tử chưa được sắp xếp để tìm phần tử nhỏ nhất tiếp theo. Vì vậy, nó quá chậm để được sử dụng trên các tập dữ liệu siêu lớn. (chuyể slide)

Phần cuối cùng trong buổi ngày hôm nay là M3 – các ưu điểm của tính đóng gói và ẩn thông tin trong ADT. Ẩn thông tin là gì, dòng gói là gì thì ngay bh chúng ta cùng tìm hiểu

Thứ nhất Phương châm của tính ẩn thông tin là gì?

Đó là "Giao diện hoặc định nghĩa của nó đã được chọn để tiết lộ một cách ít nhất có thể về hoạt động bên trong của nó”.

"Sự nhầm lẫn có thể xảy ra khi mọi người không phân biệt được việc che giấu thông tin và một kỹ thuật (ví dụ: trừu tượng hóa) được sử dụng để giúp xác định thông tin nào sẽ bị ẩn." (chuyển slide)

Tiếp theo là tính đóng gói

**Tính đóng gói trong java** là kỹ thuật ẩn giấu những thông tin và hiển thị không liên quan. Mục đích chính của đóng gói trong java là giảm thiểu mức độ phức tạp phát triển của phần mềm. (chuyển slide)

Mình sẽ lấy một ví dụ về tính đóng gói như này:

Mình có 1 class Student có kiểu dữ liệu String name, phạm vi là private

Để lấy đc các thông tin từ lớp Student ra thì chúng ta cần các phương thức setter and getter

Trong hàm main muốn gán giá trị cho nó thì setName mà muốn lấy ra giá trị đó thì getName là xong (chuyển slide)

Trừu tượng hóa và đóng gói là các khái niệm bổ sung: trừu tượng tập trung vào hành vi có thể quan sát được của một đối tượng ... đóng gói tập trung vào việc thực hiện dẫn đến hành vi này ... đóng gói thường đạt được thông qua việc che giấu thông tin, đó là quá trình che giấu tất cả những bí mật của đối tượng không đóng góp vào đặc điểm thiết yếu của nó.

Vậy ưu điểm của động gói trong ADT là gì?

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

Thứ 2 nó Cung cấp sự linh hoạt cho người dùng để sử dụng hệ thống rất dễ dàng

Thứ 3 nó Giúp các nhà phát triển tổ chức mã tốt hơn

Thứ 4 là 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

Thứ 5: Phương pháp này giúp các nhà phát triển khách quan hơn và định hướng kết quả.

Thứ 6: Mã đóng gói khá linh hoạt và dễ thay đổi bằng mã mới các yêu cầu.

Cuối cùng Đóng gói giúp kiểm tra đơn vị dễ dàng V.v và v.v

Đó là những ưu điểm tuyệt vời của tính động gói và che giấu thông tin trong ADT

(chuyển slide)

**Phần kết luận:**

"Trừu tượng hóa, che giấu thông tin và đóng gói nó rất khác nhau, nhưng các khái niệm có liên quan rất cao. Người ta có thể lập luận rằng trừu tượng hóa là một kỹ thuật giúp chúng ta xác định thông tin cụ thể nào sẽ được hiển thị và thông tin nào sẽ được ẩn đi. Để đóng gói thông tin theo cách che giấu những gì cần ẩn và hiển thị và những gì được dự định hiển thị. “Về phần này, ai muốn tìm hiểu kĩ hơn thì có thể tham khảo đưuòng link mình để bên dưới. Và phần thuyết trình của mình đến đây đã kết thúc, cảm ơn mọi người đã lắng nghe! Ai có nhận xét hay góp ý cho mình thì cứ tự nhiên ạ. (hết)