Xin chào mọi người đã đến với buổi thuyết trình của mình về kiểu dữ liệu trừu tượng và kiểu dữ liệu ngăn xếp. 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. Thì ngay bây giờ mình xin đc bắt đầu (chuyển slide)

Những nội dung hôm nay sẽ bao gồm

* ADT là gì?
* Cần làm những gì khi thực thi 1 ADT?
* Kiểu dữ liệu ngăn xếp là gì?
* Và Các hoạt động trên kiểu dữ liệu ngăn xếp? (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 trừu tượng hóa phương thức là gì?**

Kiểu dữ liệu trừu tượng (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. 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)

**4. Bây giờ tôi sẽ lấy ví dụ với List**

Tôi tạo 1 mảng danh sách học sinh với kiểu dữ liệu là String

Để hoàn thành bài List này, tôi đã tạo ra 3 class gồm lớp Main, lớp Node và 1 lớp Student\_list (chuyển slide)

Cách hoạt động của List cũng khá dễ hiểu, đầu tiên tại lớp Node chúng ta khởi tạo 2 thuộc tính đó là Node next và String data;

Sau đó là 2 hàm khởi tạo trong đó 1 hàm khởi tạo có tham số đều bằng NULL. (chuyển slide)

Bây giờ ngay tại lớp Student\_list, tôi tạo 2 node đó là firstList và lastList, khởi tạo Constructor không tham số và Constructor mặc định bằng NULL.

Hàm check rỗng thì trả về firstList = Null (bấm)

Đây chỉ là 1 ví dụ đơn giản nên mình chỉ demo add 1 phần tử vào cuối và các phần tử trong 1 mảng danh sách mà thôi. Như mọi người thấy thì ở đây có public void add() và public void addMany()

Tại add thì ta truyền 1 string vào trong đó – tạoooo 1 Node mới. Nếu rỗng thì firstList = lastList = 1 Node mới còn không thì lastList.next sẽ = newNode.

Để thêm được nhiều phần tử trong 1 danh sách, ta phải dùng đến mảng. Ta cho 1 vòng lặp foreach là được (Chuyển slide)

Việc thêm các phần tử đã xong, h thì ta phải in chúng ra màn hình. Ở đây, tôi tạo 1 hàm show Student. Để show ra đc, ta phải có 1 cái nút hiện tại bằng firstList và 1 vòng lặp While, nếu như mà cái nút hiện tại không = null thì System out print Node\_hientai.data và Node\_hientai = Node\_hientai.next Ok, thì lớp node và lớp student list đã xong, không thể 0 có lớp Main để thực thi chương trình được (bấm)

Phần nhập vào n phần tử thì đã quá quen thuộc với chúng ta, và đây là kết quả in ra 1 mảng danh sách tên học sinh (chuyển slide)

**II. Chúng ta đi qua phần 2 lớn: 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)

**III. phần thứ 3, 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 *Các hoạt động trong Stack ntn ta qua phần 4* (chuyển slide)

**IV. Các hoạt động trong Stack**

Vì chỉ có thể lấy đĩa nếu có đĩa trên chồng và chỉ có thể thêm đĩa vào chồng khi có đủ chỗ; đó là nếu ngăn xếp không quá cao.

Do đó, một ngăn xếp được định nghĩa về các hoạt động thay đổi trạng thái của nó và các hoạt động kiểm tra trạng thái này.

Các hoạt động đó bao gồm như là:

* clear () - Xóa ngăn xếp.
* isEmpty () - Kiểm tra xem ngăn xếp có trống không.
* push (phần tử) - Đặt phần tử lên trên cùng của ngăn xếp.
* pop () - Lấy phần tử trên cùng từ ngăn xếp.
* topEl () - Trả lại 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 cùng tìm hiểu về 2 hoạt động này (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 thì lỗi và out ra

**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 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 mảng của hoạt động pop (), phần tử dữ liệu không thực sự bị xóa, thay vào đó phần **trên** được giảm xuống vị trí thấp hơn trong ngăn xếp để trỏ đến giá trị tiếp theo. Nhưng trong triển khai danh sách liên kết, pop () thực sự loại bỏ phần tử dữ liệu và phân bổ không gian bộ nhớ.

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 và out.

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

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

**Và** Trả về thành công (chuyển slide)

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**

**Tiếp là Backtracking: thuật toán quay lùi – nó giống thuật toán vét cạn ấy,** với những ai chưa biết về loại thuật toán này thì

**Vét cạn** là một phương pháp giải toán trong tin học: tìm nghiệm của một bài toán bằng cách xem xét tất cả các phương án có thể Ưu điểm của phương pháp này là luôn đảm bảo tìm ra nghiệm đúng và chính xác. ... Do đó vét cạn thường chỉ phù hợp với các bài toán có kích thước nhỏ

Và đó là toàn bộ phần thuyết trình của mình những gì về ADT và Stack (bấm qua reference) Nếu cô và các bạn có gì thắc mắc hay cần góp ý thì ngay bây giờ cứ hỏi mình xin lắng nghe (chuyển slide THANK YOU)