1. : Học về Java core :

Mục tiêu trong 2 tuần có thể học basic java ở mức hiểu và có thể sử dụng được.

1. Tìm hiêu về thread trong java , multithreading trong java

Tìm hiểu basic về thread trong Java , thread là gì , tại sao điểm mạnh của Java là mutilthread.

1. Tìm hiểu về Java IO

Hiểu về IO trong Java (đọc file)

Trả lời:

* I/O đọc ghi dữ liệu vào file
* Stream là dãy dữ liệu bao gồm nhiều byte
* 1 số class hay dùng FileOutStream , FileInputStream, FileWriter , FileReader , … ( đều đc kế thừa từ class cha OutStream, InputStream)

1. Tìm hiểu về exception trong Java

Mục tiêu : nắm rõ được khái niệm exception , cách xử lý exception , quá trinh lan truyền exception , phân loại các loại exception , các từ khoá liên quan đến xử lý exception .

Trả lời:

* Exception: là quá trình từ 1 luồng đang chạy bình thường đột nhiên ở 1 phần nào đó của chương trình nó xảy ra lỗi và làm luồng đó ngừng chạy.
* Được xử lý bằng từ khóa catch() {} và dùng những class xử lý lỗi , với : tại 1 thời điểm chỉ có 1 lỗi và chỉ có 1 catch được thực thi và các catch nên để từ cụ thể đến chung chung.
* Quá trình lan truyền exception là : các phương thức con có lỗi mà lại đặt trong phương thức khác , qt này vẫn được xử lý.
* PHân loại: 3 loại ( check, uncheck và error)

+ check: các class kế thừa class exception trừ runtime exception: nó được kiểm tra lỗi ở quá trình compile time

+ unckeck: các class kế thừa class runtime exception (lỗi số học, vượt qúa chỉ số mảng) : nó được kiểm tra lỗi ở quá trình runtime

+ error: lỗi này thì không thể fix đc.

* Từ khóa try , catch , finally , throw ( ném ra lỗi) , throws ( khai báo lỗi). Phân biêt đc final , finally và finalize . Phân biệt được throw và throws.

1. Tìm hiểu về lớpNumber , kiểu native trong java

Mục tiêu : nắm rõ về box , unboxing trong Java , phân biệt được sự khác nhau giữa kiểu native và classwrapper trong Java.

Trả lời:

* Box: trình biên dịch sẽ bọc kiểu dữ liệu gốc thành kiểu đối tượng thông qua class wraper thành kiểu đối tượng Byte,Short,Int,Long,Float,Double.
* Unboxing : cũng thông qua trình biên dịch để chuyển từ kiểu đối tượng sang kiểu dữ liệu gốc. Ví dụ: Int x=5 // kiểu đối tượng ; x+=5 // kiểu dl gốc;
* Khác nhau giữa kld native và class wrapper là native là kiểu dữ liệu gốc còn class wrapper sẽ tạo ra các đối tượng từ kdl gốc và đối tượng thì có những phương thức mà native không có.

1. Tìm hiểu về Serialization

Mục tiêu : Tìm hiểu về khái niệm và mục đích.

Trả lời:

* Khái niệm và mục đích: dùng để chuyển đổi trạng thái của 1 đối tượng Java sang 1 định dạng mới ( ví dụ byte streams) rồi lưu vào đâu đó ( qua file ,trên sever hoặc truyền qua mạng… ) sau đó sẽ được tiến trình mới sử dụng.
* Để tạo ra Serialization class đó phải implements Serializable interface.
* Quá trình Deserialization là từ byte streams -> object Java.
* 1 số note: các class con kế thừa class được serialization thì class con cũng tự động đc serialization , qh HAS-A thì class dùng bên trong class cũng phải serialization , các trường dữ liệu static sẽ không được serialization ……

Kế hoạch xong các phần trên vào ngày 12/12

1. Tìm hiểu về Java DataTime , bao gồm cả datatime trong Java 8

Kế hoạch : buổi sáng ngày 13/12

Muc tiêu : Tìm hiểu về các thư viện datatime trong java bao gồm : Date , Calender , LocalDateTime , so sánh giữa các loại , tim hiểu về khái niệm timestamp . Làm bài tập về xử lý date time ,so sánh dateTime .

1. Tìm hiểu về Java Collection

Mục tiêu

Nắm rõ được collection là gì ?

Các cấu trúc dữ liệu được sử dụng trong từng loại collection.

Tìm hiểu các interface : List, Set , Map, Queue

Các class : ArrayList, LinkedList , HashSet , TreeSet , HashMap, LinkedHashMap

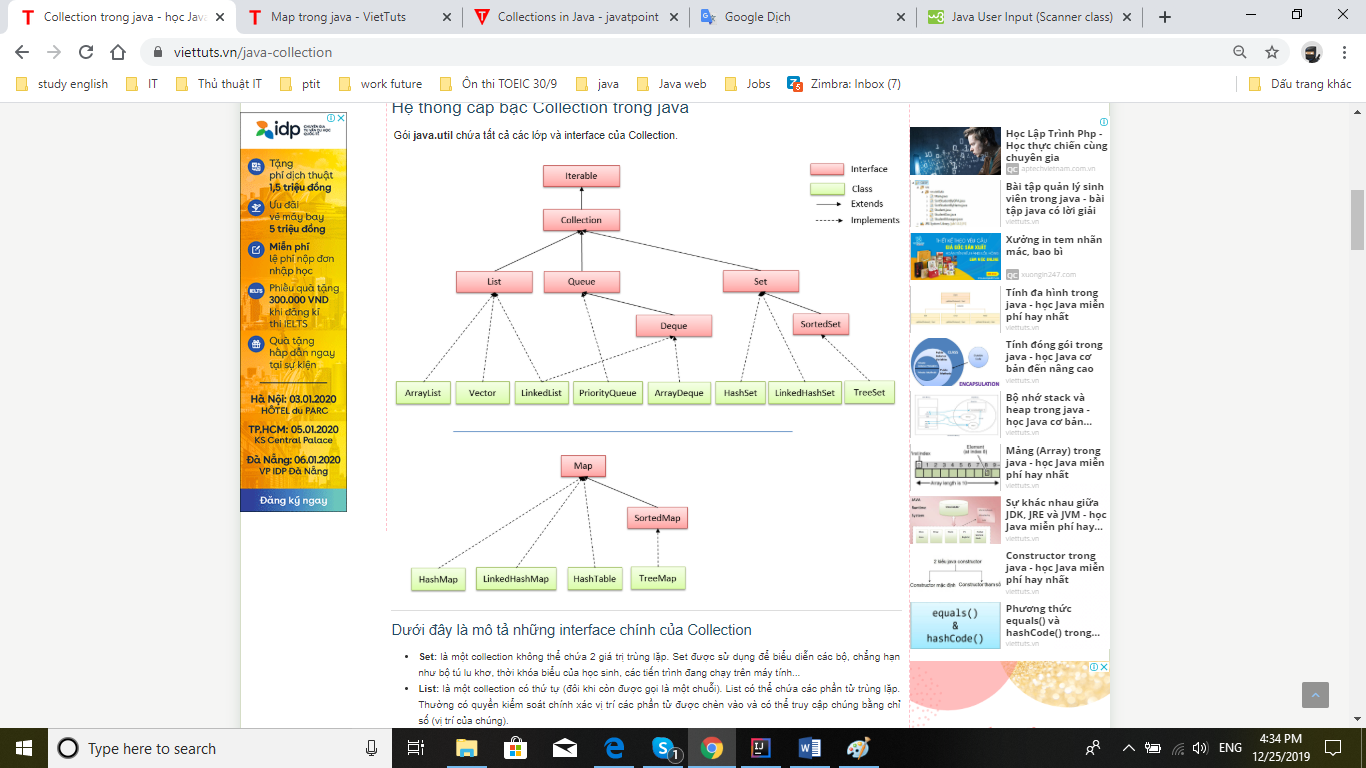
Phân biệt được các ưu , nhược điểm của từng loại , khi nào nên dùng loại nào

Tìm hiểu về khái niệm hashkey với mỗi object.

Chú ý : đọc vào code của thư viện để hiểu đc bản chất là tốt nhất.

Trả lời

* Interator : là 1 interface nó được collection interface kế thừa và nó thường được dùng để duyệt các phần tử trong collection . Nó có 3 phương thức cơ bản: hasNext() /\* kiểm tra xem phần tử tiếp theo có không và trả về Boolean \*/ , next() /\* lấy ra phần tử hiện tại và trỏ đến vị trí tiếp theo \*/ , remove().
* Note: Duyệt trong collection thường dùng Interator hoặc foreach . Khi khởi tạo các collection nên dùng theo generic và kiểu dữ liệu của collection là đối tượng không thể nào là loại nguyên thủy ( mảng tĩnh có thể là cả đối tượng hoặc dl nguyên thủy ).
* Collection là :root interface trong mô hình framework collection. Nó được nhiều interface khác kế thừa như List, Queue, Set ; và nó kế thừa Interator interface. Nó có 7 phương thức hay dùng như là : add( ) , addAll( ), remove , removeAll , contains , containsAll , inEmpty() , size , hashCode , interator() ,…



* Collections: là 1 class chứa các phương thức static.
* List : là 1 interface dùng để lưu trữ các phần tử và có thể các phần tử trùng lặp nhau. Lưu trữ các phần tử theo thứ tự input. Và có chỉ số index được lưu vào. Có các phương thêm sau: get(index) , set(index, object), remove (index),…

+ ArrayList: là class lưu trữ các phần tử liên tiếp nhau trong bộ nhớ nên vì vậy ứng dụng là dùng dể truy xuất, đọc, sửa các phần tử, không đồng bộ

+ LinkedList: là class lưu trữ các phần tử có thể liên tiếp hoặc rời rạc nhau trong bộ nhớ nên vì vậy ứng dụng là dùng dể thêm, xóa các phần tử. Và không đồng bộ

+ Vector: ứng dùng trong đa luồng và đồng bộ

* Queue : là 1 interface có thể lưu trữ nhiều phần tử trước khi xử lý.
* Set : là 1 interface có thể lưu trữ các phần và không được phép trùng lặp và không theo thứ tự đi vào.
* SortedSet: là 1 interface được kế thừa từ Set nó có ưu điểm là lưu trữ theo thứ tự tăng dần hoặc giảm dần.
* Map là 1 interface dùng để lưu trữ các phần tử và không cho phép các phần tử trùng lặp và phần tử có định dạng key- value , với mỗi 1 key chỉ ánh xạ duy nhất 1 value. Có các class được kế thừa nó là: HashMap, LinkedHashMap , HashTable , TreeMap.
* SortedMap: là 1 interface được kế thừa từ Map nó có ưu điểm là lưu trữ theo thứ tự tăng dần hoặc giảm dần theo Key ( ứng dujnng làm danh bạ điện thoại hay từ điển,…)

1. Tìm hiểu về Java JDBC

Kế hoach : Tìm hiểu trong 3 ngày : 16.17,18 . Chiều 18 sẽ review

Sử dụng postgres làm DB .

Dựng db local , connect tới pgadmin , tạo 1 db , tạo 1 table , các khái niệm về khoá chính , các kiểu dữ liệu, các câu lênh select , update , tìm kiếm phân trang.

Tìm hiểu về JDBC , connect tới db , tạo db : user info , thực hiện các câu lệnh CRUD (Create ,Read, Update , Delete )

* JDBC là một Java Api được dùng để kết nối và truy xuất đến cơ sở dữ liệu. Trình điều khiển là jdbc ( đc viết bằng ngôn ngữ java)
* Mô hình: JDBC api 🡪 java app 🡪 JDBC driver 🡪 database
* Muốn kết nối java app đến db phải dùng :
* tải jdbc driver tương ứng với database đó rồi add nó vào dự án
* Class.forName(org.postgresql.Driver)
* DriverManager(url, username, pass) // url phải chưa hostname + port + database name
* Connection interface: được dùng để kết nối java app vs db và để tạo ra các đối tượng Statement, PreparedStatement,….. thông qua các phương thức như creatStatement(),….
* Statement interface: được dùng để truy xuất đến db sử dùng các phương thức executeUpdate(String sql) , executeQuery(String sql) ,… ( riêng executeQuery() còn có tể tạo ra đối tượng ResultSet )
* PreparedStatement interface: được dùng để truy xuất đến db . Nó có thể sử dụng theo như cách của Statement hoặc nó có thể để sql trong hàng tạo PreparedStatement như conn.createPrepareStatement(sql) và nó có những phương thức : executeUpdate(String sql) , executeQuery(String sql) ,… ( riêng executeQuery() còn có thể tạo ra đối tượng ResultSet ), setInt() , setString() ,….

+ Ví dụ:

PreparedStatement ps= conn.prepareStatement("insert into employees(id,fullname,age) values(?,?,?)");

ps.setInt(1,id);

ps.setString(2,name);

ps.setInt(3,age);

ps.executeUpdate();

+ Note: execute(): dùng để tạo, xóa, sửa bảng, executeUpdate() : dùng cho câu INSERT, UPDATE, DELETE , **executeQuery**: dùng câu SELECT. ( đúng cho cả khi dùng Statemment và PreparedStatement ).

* ResultSet interface: được dùng để chỉ con trỏ đang trỏ tới hàng của bằng. Mặc định thì nó sẽ trỏ tới hàng đầu tiên của bảng db và nó sẽ không tự động dịch chuyển con trỏ. Nó có các phương thức next(),…,getInt(indexColumn/nameColumn) , getString(indexColumn/nameColumn).
* Còn về db postgresql giống la lá vs mysql
* Phân trang :

ví dụ: {

select \*

from (select \*, row\_number() over(order by id) as rowNumer

from employees) as db

where db.rowNumer between 2 and 3;

}

Dòng lệnh kia nó nghĩa là: đánh số tuần tự cho hàng theo id và gán tên tạm thời của giá trị số đó cho rowNumber. Và cả cái bảng vừa tạo gắn tạm thời với tên là db

1. : Tìm hiểu về spring boot và tạo API CRUD

( spring boot 2days, restful 3days, jpa 2 days )

DB sử dụng : postgres

FW sử dụng : springboot , Spring data JPA

Trong 2 tuần còn lại sẽ học sử dụng về spring boot , tìm hiểu về khái niệm restapi là gì , json là gì , server tomcat , spring core DI , các từ khoá của spring , ORM là gì , spring dataJPA , cách tạo 1 repository và query đến db .

1. Phần lý thuyết :

Spring boot là gì

Tạo 1 app spring đơn giản

RestApi là gì ,các phương thức : Post, get , Put , delete

Sử dụng spring boot tạo 1 api đơn giản : helloword

Học cách truyền param trên url : path , queryparam , json body

ORM là gì , hibernate là gì , spring data jpa là gì ?

Transaction trong DB là gì ?

Khái niệm về rollback trong transaction .

1. Phần thực hành

Sẽ tiến hành tạo các API sau

1. API đăng ký người dùng

Thông tin bao gồm Tên , Ngày sinh , Số điện thoại , Địa chỉ , Nghề nghiệp, Số CMT

Y/C :

* SDT làm key , nếu sdt đã đăng ký rồi thì hiển thị ra thông báo
* 1 số CMT không được phép quá 3 tài khoản (max = 3)

1. API update thông tin người dùng

* Có thể update thông tin người dùng , trừ số điện thoại
* CMT có thể update nhưng không được phép quá 3 tài khoản 1 số CMT

1. API xoá thông tin người dùng

Có 2 cách xoá

Yc impl cả 2.

* Sử dụng số điện thoại để xoá
* Sử dụng User ID để xoá

1. API get thông tin người dùng theo userID

* Sử dụng userId để tìm kiếm detail ,
* Có trả ra mã lỗi khi userId không tồn tại trong hệ thống.

1. API Tìm kiếm người dùng

* Có padding : truyền lên limit , offer . Khi không truyền set giá trị default là 0 và limit là 10.
* Có trả về total element (count xem có bao nhiêu kết quả)
* Có trả về page hiện tại
* Param tìm kiếm : Tên , Địa chỉ, Số điện thoai

Có áp dụng các khái niệm về exception , transaction trong spring datajpa với rollback khi xảy ra exception .

Plan

Phần I : Core

Các mục : 1 đến 5 : Hoàn thành vào ngày 12/12 , Chiều 12 sẽ review lại .

Mục 6 : Ngày 13 . Chiều 13 sẽ review

Mục 7 : Từ ngày 14 đến hết chủ nhật ngày 22 , sáng 23 sẽ review

Mục 8 : Ngày 23, 24, 25 . Sáng 26 sẽ review.

Phần II . Spring boot

1. Phần lý thuyết : ngày 26, 27 sẽ học lý thuyết , sáng thứ 7 ngày 28 sẽ review
2. Phần thực hành planing sau.