TA:

* introduce yourself, projects
* Authentication, Authorization

Spring:

* IOC vs DI
* bean initialization: injection types
* @Qualifier

Java core:

* OOP concepts
* array vs ArrayList
* overriding and overloading
* Java 8:
  + What’s new in java 8
  + Functional interface
  + Stream: intermediate and terminate, stream and parallel stream

Database:

* Join types
* Query to select largest number->second largest number->second largest number in case there are some records with same value

TA:

* introduce yourself, projects
* 4 feature of oop

Java core:

* 4 feature of oop
* Compare 2 object
* Difference between list vs arraylist

Database:

* Join types
* Index
* Get max value of column
* limit

Phỏng vấn tháng 6:

1. Có biết về Exception k?

* Exception là sự kiện xảy ra trong quá trình thực thi của Java program và nó phá vỡ cái flow của 1 chương trình và có thể làm chết chương trình nếu không được xử lý

2. Em có biết immutable k?

* có thể hiểu như final là các đặc tính được tạo ra nhưng ko thể thay đổi được

3. Phân việt @PathVariable và @RequestParam

* @PathVaribale dùng ở Controller để lấy giá trị người dùng nhập trên URL nhưng ko theo dạng Key Value mà là
* @RequestParam để lấy giá trị của người dùng nhập trên URL nhưng theo dạng Key Value

Example : có 1 link URL [**http://localhost:8080/MyApp/user/1234/invoices?date=12-05-2013**](http://localhost:8080/MyApp/user/1234/invoices?date=12-05-2013)

ở trong controller ta sẽ set RequestMapping là /user/{user\_id}/invoices

như vậy

@PathVariable(“user\_id”) dùng để lấy user\_id trên link nghĩa là 1234 tương ứng với lihk URL ”) d  
 @RequestParam(“dateùng để lấy tham số date truyền vào bằng URL tương ứng ở trên link

4. Giải thích về Bean trong Spring

5. Những lưu ý khi dùng GROUP BY trong DB

6. Làm nào để select ra những record k trùng nhau. Ví dụ 1 bảng có id, name, age: select ra thằng tất cả thẳng record mà có name k trùng nhau

* dùng distinct trong sql SELECT DISTINCT name from table\_name

7. Làm nào để select ra những thằng có tuổi lớn hơn 21 mà ko dùng mệnh đề WHERE trong SQL

* Dùng GROUP BY và HAVING   
  Select \* from table\_name   
  GROUP BY AGE  
  HAVING AGE > 21

8. Có các số từ 1 đến 100 nhưng thiếu 1 số được ném vào mảng k sắp xếp, tìm số đó.

9. LinkedList làm thế nào để xóa node thứ 5

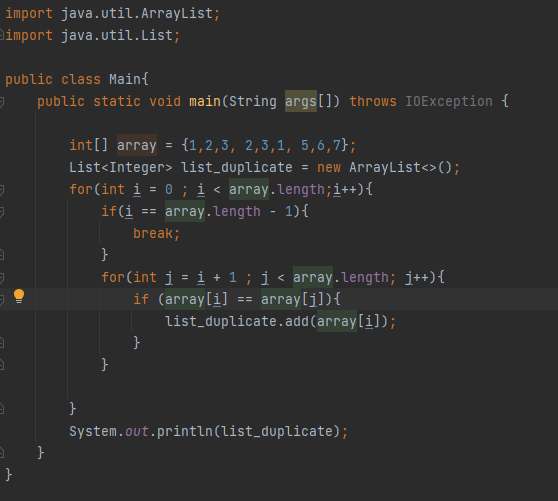
* 1 node trong linked list sẽ có data và next để chỉ đến node sau đó
* vậy thì mình sẽ lấy cái next của node thứ 4 ( hiện đang là 5) mình sẽ chuyển thành 6 ( còn nếu node thứ 5 nó đang trỏ đến null thì mình sẽ chuyền thành null) như vậy là mình đã xóa được node thứ 5

10. LinkedList làm thế nào để thay đổi node ban đầu của nó.

* Mình sẽ tạo ra 1 node có next trỏ đến node thứ 2 xong để Head trỏ để node ý

11. Tìm các phần từ trùng lặp trong mảng

Ví dụ: [1,2,3, 2,3,1, 5,6,7] in ra số đó



12. Garbage Collection trong java

* Là phần dọn rác trong JVM dọn rác trong Heap memory nơi để lưu trữ các object được khởi tạo bằng từ khóa new và các biến static
* Dùng để dọn các referrence các object ở trong Heap mà đã lâu chưa được sử dụng để tiết kiệm vùng nhớ

13. Finally

* được dùng có exception handling kết hợp với try catch và là bước cuối cùng dù có có bắt exception ở catch hay ko thì đều sẽ chạy finally

14. Có cách nào closed source ko dùng Finally ko?

15. Làm thế nào để tránh NullPointerException khi truyền param object vào method.

* mình có thể dùng exception handling (try-catch để check đầu vào trước)

16. Phân biệt Inner Join với Left Join.

* Inner join sẽ trả về tất cả các row mà các trường join ở 2 bảng đều không null
* Còn left join sẽ trả về các row mà các trường join ko null ở bảng thứ nhất

17. Làm thế nào để Map địa chỉ link với Class.

* Dùng @RequestMapping

18. Biết CASE trong SQL k?

* dùng để thể hiện 1 trường giá trị hiển thị như mong muốn của user

-------------------------------------------

1. 4 tính chất OOP.   
   - Abstraction  
   - Encapsulation  
   - Inheritance  
   - Polymorphism
2. hỏi kĩ hơn về tính đóng gói và che giấu dữ liệu   
   - Tính đóng gói : gói tất cả các thuộc tính, method của object vào trong 1 class có các phương thức Read-only và write-only (getter and setter)  
   - Tính che giấu dữ liệu : Như kiểu xây dựng 1 cái sườn để các lớp con sẽ kế thừa và xây dựng từ cái sườn đó đi lên
3. phân biệt String, Stringbuilder, String Buffer

|  | String | StringBuffer | StringBuilder |
| --- | --- | --- | --- |
| Định nghĩa | Là một chuỗi java để tạo đối tượng String, là một chuỗi các ký tự | Là một class để tạo đối tượng String có thể sửa đổi và đảm bảo an toàn cho các luồng | Là một class để tạo ra đối tượng String có thể sửa đổi nhưng ko đảm bảo an toàn cho các luồng |
| Tính đột biến | là bất biến | là một lớp có thể thay đổi | là một lớp có thể thay đổi |
| An toàn | an toàn | an toàn (Thread-safe) và còn đồng bộ hóa | ko an toàn và ko đồng bộ hóa |
| Hiệu quả | Nhanh | Chậm | Rất nhanh |

1. so sánh 2 String |  
   - Dùng compareTo() if str1 > str2 thì trả về 1 ngược lại thì -1 và bằng thì trả về 0  
   - Dùng equal() hoặc equalsIgnoreCase() trả về kiểu boolean  
   - Dùng hashCode() nếu 2 string bằng nhau thì hashcode sẽ bằng nhau
2. abstract class có constructor không? phân biệt abstract class với interface (câu này rất quan trọng)
3. phân biệt overload và override  
   -Overload là mình sẽ ghi đè phương thức sẽ ghi lại phương thức được khởi tạo ở lớp cha ( Thể hiện tính kế thừa và đa hình)  
   - Overrider là nạp chồng là có nhiều phương thức cùng tên, những phương thức sẽ khác tham số truyền vào hoặc kiểu trả về (Thể hiện tính đa hình)
4. Khái niệm DI ? có mấy loại Inject   
   - DI(Dependency Injection) là nguyên lý để giảm sự phụ thuộc giữa các module gốm có 2 ý chính  
    + Các module cấp cao ko nên phụ thuộc trực tiếp vào module cấp thấp mà nên thông qua abstraction  
    + Abstraction ko nên phụ thuộc vào các chi tiết mà ngược lại  
   - Có 2 loại inject  
    + Constructor based injection : inject bằng constructor   
    + Setter based injection : inject bằng setter
5. Spring bean là gì? có bao nhiêu loại scope, default là gì?  
   - Bean là một đối tượng được khởi tạo ,xây dựng và được quản lý bởi IoC Container hay có thể bảo là bean là module chính của chương trình tạo ra bởi Spring IoC container
6. các loại join trong SQL? lấy ví dụ cụ thể về Left join?   
   - inner join  
   - left join  
   - right join  
   - full outer join
7. khái niệm index trong SQL? có bao nhiêu loại index

* là bảng tra cứu đặc biệt mà công cụ truy xuất có thể sử dụng để tăng hiệu suất tìm kiếm , liên hệ với quyền sách, index giống như mục lục giúp mình tìm kiến nhanh hơn , nếu ko sử dụng index thì tìm từng trang từng chữ một.

1. phân biệt Session và Cookie trong Servlet

| **Cookie** | **Session session là gì session là gì** |
| --- | --- |
| **Cookie** được lưu trữ trên trình duyệt của người dùng. | **Session** không được lưu trữ trên trình duyệt. |
| Dữ liệu **cookie** được lưu trữ ở phía client. | Dữ liệu **session** được lưu trữ ở phía server. |
| Dữ liệu **cookie** dễ dàng sửa đổi hoặc đánh cắp khi chúng được lưu trữ ở phía client. | Dữ liệu **session** không dễ dàng sửa đổi vì chúng được lưu trữ ở phía máy chủ. |
| Dữ liệu **cookie** có sẵn trong trình duyệt đến khi expired. | Sau khi đóng trình duyệt sẽ hết phiên làm việc (session) |

------------------------------------------------

