Phỏng vấn Java

**Câu hỏi 1:** Nêu 4 tính chất OOP trong java

**Trả lời:**

**Tính đóng gói (encapsulation) và che giấu thông tin (information hiding)** :

* Tức là trạng thái của đối tượng được bảo vệ không cho các truy cập từ code bên ngoài như thay đổi trong thái hay nhìn trực tiếp. Việc cho phép môi trường bên ngoài tác động lên các dữ liệu nội tại của một đối tượng theo cách nào là hoàn toàn tùy thuộc vào người viết mã. Đây là tính chất đảm bảo sự toàn vẹn, bảo mật của đối tượng Trong Java, tính đóng gói được thể hiện thông qua phạm vi truy cập (access modifier). Ngoài ra, các lớp liên quan đến nhau có thể được gom chung lại thành package.

**Tính kế thừa** là khả năng cho phép ta xây dựng một lớp mới dựa trên các định nghĩa của một lớp đã có. Lớp đã có gọi là lớp Cha, lớp mới phát sinh gọi là lớp Con và đương nhiên kế thừa tất cả các thành phần của lớp Cha, có thể chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại.

**Tính đa hình (polymorphism)**:

* Khi một tác vụ được thực hiện theo nhiều cách khác nhau được gọi là tính đa hình.
* Đối với tính chất này, nó được thể hiện rõ nhất qua việc gọi phương thức của đối tượng. Các phương thức hoàn toàn có thể giống nhau, nhưng việc xử lý luồng có thể khác nhau. Nói cách khác: Tính đa hình cung cấp khả năng cho phép người lập trình gọi trước một phương thức của đối tượng, tuy chưa xác định đối tượng có phương thức muốn gọi hay không. Đến khi thực hiện (run-time), chương trình mới xác định được đối tượng và gọi phương thức tương ứng của đối tượng đó. Kết nối trễ giúp chương trình được uyển chuyển hơn, chỉ yêu cầu đối tượng cung cấp đúng phương thức cần thiết là đủ.
* Trong Java, chúng ta sử dụng nạp chồng phương thức (method overloading) và ghi đè phương thức (method overriding) để có tính đa hình.
* **Nạp chồng (Overloading)**: Đây là khả năng cho phép một lớp có nhiều thuộc tính, phương thức cùng tên nhưng với các tham số khác nhau về loại cũng như về số lượng. Khi được gọi, dựa vào tham số truyền vào, phương thức tương ứng sẽ được thực hiện.
* **Ghi đè (Overriding)**: là hai phương thức cùng tên, cùng tham số, cùng kiểu trả về nhưng thằng con viết lại và dùng theo cách của nó, và xuất hiện ở lớp cha và tiếp tục xuất hiện ở lớp con. Khi dùng override, lúc thực thi, nếu lớp Con không có phương thức riêng, phương thức của lớp Cha sẽ được gọi, ngược lại nếu có, phương thức của lớp Con được gọi.

**Tính trừu tượng (abstraction)**:

* Tính trừu tượng là một tiến trình ẩn các chi tiết trình triển khai và chỉ hiển thị tính năng tới người dùng. Tính trừu tượng cho phép bạn loại bỏ tính chất phức tạp của đối tượng bằng cách chỉ đưa ra các thuộc tính và phương thức cần thiết của đối tượng trong lập trình.
* Tính trừu tượng giúp bạn tập trung vào những cốt lõi cần thiết của đối tượng thay vì quan tâm đến cách nó thực hiện.
* Trong Java, chúng là sử dụng abstract class và abstract interface để có tính trừu tượng.

**Câu hỏi 2:** Tại sao java lại bảo mật hơn các ngôn ngữ khác?

**Trả lời:**

Vì java chạy trên máy ảo JVM nên sẽ không xác định được chính xác địa chỉ để tấn công

**Câu hỏi 3:** Phân biệt array và arraylist

**Trả lời:**

**Mảng (Array)** là một cấu trúc dữ liệu có kích thước cố định, trong khi **ArrayList** là một lớp Collection có thể thay đổi được kích thước. Nghĩa là chúng ta không thể thay đổi kích thước của mảng khi đã tạo, nhưng ArrayList có thể được thay đổi.

Chúng ta không thể lưu giữ dữ liệu nguyên thủy trong ArrayList, nó chỉ có thể chứa các đối tượng. Nhưng mảng có thể chứa cả hai kiểu dữ liệu nguyên thủy và các đối tượng trong Java. Kể từ Java 5, kiểu nguyên thủy được tự động chuyển đổi trong các đối tượng được gọi là auto-boxing.

**Sự khác nhau giữa Array với ArrayList** được tóm tắt lại như trong bảng sau:

|  |  |
| --- | --- |
| **Array** | **ArrayList** |
| 1) Kích thước **cố định**. | Kích thước có thể **thay đổi được**. |
| 2) Có thể lưu trữ dữ liệu kiểu **nguyên thủy** và **đối tượng**. | Chỉ có thể lưu trữ dữ liệu kiểu **đối tượng**. Kể từ Java 5, kiểu nguyên thủy được tự động chuyển đổi trong các đối tượng được gọi là **auto-boxing**. |
| 3) Tốc độ lưu trữ và thao tác **nhanh hơn**. | Tốc độ lưu trữ vào thao tác **chậm hơn**. |
| 4) Chỉ có thuộc tính **length**. | Có nhiều phương thức để thao tác với dữ liệu. |

**Câu hỏi 4:**  Thành phần nằm dưới ArrayList là gì? Cơ chế tăng kích thước của arraylist

**Trả lời:**

Thành phần nằm dưới ArrayList là Array.

Khi arraylist tăng kích thước có nghĩa là nó sẽ tạo 1 array mới và nhân size lên 1.5 sau đó copy toàn bộ phần tử từ array cũ sang array mới và loại bỏ array cũ đi.

**Câu hỏi 5:** Garbage collector trong java sẽ giải phóng các đối tượng như thế nào?

**Trả lời:**

Các object không còn được tham chiếu và bị tham chiếu trong bộ nhớ Heap sẽ được giải phóng.

**Câu hỏi 6:** Spring security là gì?

**Trả lời:** câu này trả lời đc phần lợi ích. Đọc theo link dưới này nhé!

<https://viblo.asia/p/co-che-hoat-dong-cua-spring-security-tat-tan-tat-cac-vi-du-ve-spring-security-63vKjmvx52R>

**Câu hỏi 7 :** JWT là gì? Các thành phần của JWT. (không trả lời được)

**Trả lời:** <https://topdev.vn/blog/jwt-la-gi/>

**Câu hỏi 8:** Dependency Injection là gì? IoC là gì?

**Trả lời:**

<https://toidicodedao.com/2015/11/03/dependency-injection-va-inversion-of-control-phan-1-dinh-nghia/>

đọc theo đây cho chuẩn nhé. Trả lời chỉ có ý đúng thôi

**Câu hỏi 9:** Biết viết thủ tục, hàm, package ko?

**Trả lời :** Biết xong nó hỏi: “Có tạo được object trong thủ tục không?” – Trả lời “Có.”

**Câu hỏi 10:** Biết các design pattern nào?

**Câu hỏi 11:** Có một object là Ngựa: (Đi bằng 2 chân). Có một object là Người: (Đi bằng 2 chân, cầm nắm bằng 2 tay). Có một object là Nhân mã: (Đi bằng 4 chân, cầm nắm bằng 2 tay). Thiết kế các class làm sao cho tối ưu về mặt coding. (Tự làm nhé. Hôm đó rep chắc hơi fail)

**Câu hỏi 12:** Xử lý dự liệu lớn. Có 1 file excel 10.000 dòng. Có 3 cột (A,B,C). Muốn đọc file excel và lưu vào DB vs 2 cột thông tin (A,B+C) thì làm như thế nào?

**Trả lời:**

Câu này hôm đó ko trả lời đc. Nhưng lúc về thì nghĩ ra 1 cách:

Dùng Hibernate Batch. <https://gpcoder.com/6632-hibernate-batch-processing/>

Tổng quan phỏng vấn java:

- Java core

- Sql: thường sẽ là oracle

- Spring: Tập trung vàoSpring boot, IOC, DI, Bean, BeanScope, Spring security

- Hibernate

- Design Pattern