**1.JAVA là gì ?**

Trả lời:

Java là ngôn ngữ lập trình hướng đối tượng

Java và javaScript khác nhau như thế nào?

- java được phát minh bởi jame A.gosling, javaScript được phát minh bởi Brendan của Nest cape.

**Điểm khác biệt**

* Java là ngôn ngữ lập trình tĩnh, hướng đối tượng, hoạt động trên nhiều nền tảng. [JavaScript là ngôn ngữ lập trình động](https://quantrimang.com/javascript-la-gi-155978) (hay ngôn ngữ kịch bản - scripted language) được sử dụng để làm cho các trang web và ứng dụng trở nên sinh động.
* Java dựa trên lớp (class), còn JavaScript thì động.
* Java là một ngôn ngữ độc lập. JavaScript phụ thuộc nhiều hơn, nghĩa là nó hoạt động với [HTML](https://quantrimang.com/html) và [CSS](https://quantrimang.com/hoc-css) trên các trang web để tạo nội dung động.
* Vào năm 2019, JavaScript là thứ bắt buộc phải học đối với các nhà phát triển web, vì nó được sử dụng khá nhiều ở mọi nơi, trong khi Java được coi là ngôn ngữ lập trình thế hệ trước (tuy nhiên, chắc chắn rất nhiều trang web vẫn sử dụng nó).

**Điểm tương đồng**

Như bài viết đã nói ở trên, không có nhiều điểm tương đồng giữa Java và JavaScript. Có thể điểm qua một số nét giống nhau ở cả 2 ngôn ngữ lập trình này như sau:

Cả Java và JavaScript thường được sử dụng nhiều nhất trong các ứng dụng phía máy khách

Cả Java và JavaScript đều sử dụng cú pháp C

JavaScript sao chép một số quy ước đặt tên của Java.

**2.JDK, JRE, JVM**

- JDK ( java development kit ) là bộ công cụ hỗ trợ lập trình viên phát triển java.

- JRE ( java runtime evironment) được sử dụng để cung cấp môi trường thực thi và thư viện để chạy java.

- JVM ( java vitual machine ) cung cấp máy ảo java để thông dịch java byte code ( . class) ra ngôn ngữ máy và hiển thị ra màn hình .

3 **Thông dịch, biên dịch khác nhau như thế nào ?**

* Biên dịch là dịch toàn bộ files 1 lần và sau đó sử dụng lại kết quả mà không cần phải dịch lại. Files .java biên dịch thành files.class
* Thông dịch là dịch từ dòng lệnh files.class ra ngôn ngữ máy và hiển thị ra màn hình. Mỗi lần chạy chương trình là phải dịch lại .

4 **JAVA là ngôn ngữ thông dịch hay biên dịch ?**

- Cả 2

5 **Java Platform là gì?**

 Java (Java platform) là một tập hợp các chương trình giúp lập trình viên phát triển và chạy các ứng dụng lập trình Java một cách hiệu quả. Nó bao gồm một execution engine, một trình biên dịch (compiler) và một bộ thư viện (library) trong đó.

6 **Các kiểu dữ liệu trong java, và giá trị mặc định khi khai báo.**

* Kiểu nguyên thủy :
  + Byte : 0
  + Short : 0
  + Int : 0
  + Long : 0L
  + Float : 0.0 f
  + Double : 0.0 d
  + Char : \u0000
  + Boolean : false
* Kiểu đối tượng :
  + String : null
  + Object : null

7 **Sự khác nhau giữa bộ nhớ heap và stack trong java?**

* Bộ nhớ Heap và bộ nhớ Stack bản chất đều cùng là vùng nhớ được tạo ra và lưu trữ trong RAM khi chương trình được thực thi.
* Bộ nhớ Stack được dùng để lưu trữ các biến cục bộ trong hàm, tham số truyền vào... Truy cập vào bộ nhớ này rất nhanh và được thực thi khi chương trình được biên dịch.
* Bộ nhớ Heap được dùng để lưu trữ vùng nhớ cho những biến con trỏ được [cấp phát động](https://www.banhoituidap.com/p/1247/co-nhung-kieu-cap-phat-dong-nao-trong-cc/) bởi các hàm malloc - calloc - realloc (trong C) hoặc từ khóa new (trong c++, c#, java,...).

**8 Trình bày các cách khởi tạo một mảng trong JAVA ?**

* DataType [] arrayName = new DataType[ array.length];
* DataType arrayName[] = new DataType[ array.length];
* DataType [] arrayName = {element,…};
* DataType arrayName[] = {element,…};

9 **Phần tử của mảng có thể dùng kiểu dữ liệu nào, và có giá trị mặc định là gì ?**

* Phần tử của mảng có thể dùng kiểu dữ liệu nguyên thủy hoặc object
* Giá trị mặc định theo giá trị mặc định mà kiểu dữ liệu của mảng khai báo.

10 **OOP là gì ?**

* OOP : Object oriented programing
* Là ngôn ngữ lập trình hướng đối tượng .

11 **Phân biệt class và object**

|  |  |
| --- | --- |
| Class | Object |
| * Là 1 nhóm các đối tượng có chung thuộc tính và phương thức * Là khuôn mẫu tạo ra các đối tượng * Được khởi tạo thông qua từ khóa class * Chỉ có thể khai báo 1 class trong 1 package | * Là thể hiện cụ thể của Class * Có các thuộc tính và phương thưc đặc trưng * Được khởi tạo thông qua từ khóa new * Có thể khởi tạo nhiều đối tượng trong cùng 1 class |

12 **Constructor là gì**

* Là phương thức đặc biệt để khởi tạo đối tượng .

13 **Cách khai báo constructor và đặc điểm constructor trong JAVA**

* Constructor phải trùng tên với Class
* Không có kiểu dữ liệu trả về.
* Tự động được gọi khi khởi tạo đối tượng bằng từ khóa new.
* Có thể gọi contructor từ contructor khác thông qua từ khóa this, và phải gọi trên cùng của contructor
* 1 lớp có thể có 1 hoặc nhiều constructor , nhưng phải khác tham số
* Nếu không khai báo constructor java sẽ cung cấp 1 constructor mặc định không có tham số .Khi khai báo 1 constructor có tham số thì constructor được java cung cấp sẽ mất đi.

**14 Phân biệt constructor và method**

|  |  |
| --- | --- |
| Constructor | Method |
| * Là phương thức đặc biệt để khởi tạo đối tượng * Phải trùng tên lớp * Không có kiểu dữ liệu trả về * Java cung cấp mặc định 1 constructor k tham số khi class không khai báo constructor * Không thể khai báo là static/final | * Thể hiện hành vi hành động của đối tượng * Khác tên tên lớp * Có kiểu dữ liệu trả về * Không cung cấp * Có thể gọi lại nhiều lần |

15 "**Tính bao đóng là gì ?**

* Tính bao đóng là kỹ thuật che giấu thông tin, chỉ hiển thị các thông tin cần thiết
* Mục đích: bảo vệ trạng thái thông tin của đối tượng, không cho truy cập trực tiếp vào thuộc tính của đối tượng.
* Thể hiện của tính bao đóng:
* Access modifier
* Getter và setter

**Làm sao để thu được tính bao đóng trong java ?"**

* Thu được tính bao đóng thông qua khai báo thuộc tính = private .
* Khai báo các phương thức getter/setter cần thiết để cấp quyền truy cập đến thuộc tính cần thiết

16 **Tham trị, tham chiếu**

* Tham trị : không thay đổi giá trị ban đầu của biến. Xảy ra khi truyền vào kiểu dữ liệu nguyên thủy
* Tham chiếu : thay đổi giá trị ban đầu của biến , xảy ra khi truyền vào kiểu dữ liệu đối tượng.

**17 Từ khóa static dùng để làm gì ?**

Từ khóa static dùng để khai báo thành phần thuộc lớp , quản lí bộ nhớ .

* Biến static ( biến tĩnh ) : sử dụng để tham chiếu đến thuộc tính chung của đối tượng , chỉ lấy bộ nhớ 1 lần tại thời điểm tải lớp , tiết kiệm bộ nhớ
* Phương thức static : áp dụng từ khóa static cho phương thức -> static method
* Khối static : thực thi lúc tải lớp .

**18 Ràng buộc khi sử dụng static**

* + Thuộc lớp chứ không thuộc đối tượng
  + Có thể gọi mà không cần tạo đối tượng
  + Có thể dùng để thay đổi giá trị static
  + Phương thức static : không thể gọi phương thức non-static , chỉ có thể truy xuất biến static , không thể dùng super, this.

19 **Các loại biến trong JAVA**

* Biến toàn cục ( instance): được khia báo ngoài phương thức, contructor, block
* Biến cục bộ ( local): Được khai báo trong phương thức, con tructor, block
* Biến static: biến thuộc về class

20 Trình bày các loại access modifier, và phạm vi truy cập.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Class | Package | Subclass | Public |
| Private | X |  |  |  |
| Default | X | X |  |  |
| Protected | X | X | X |  |
| Public | X | X | X | X |

21 **Kế thừa trong JAVA là gì ?**

Là cơ chế cho phép lớp con sử dụng những thuộc tính và hành vi của lớp cha.

Mục đích: rút gọn mã nguồn

Đặc điểm:

* Không có đa kế thừa
* Không kế thừa đc các thành phần được khai báo = private và constructor
* Lớp con có thể gọi được constructor của lớp cha = Super();
* Lớp con có thể mở rộng các thuộc tính và phương thức khác .
* Dùng từ khóa final để khai báo 1 class vô sinh
* Class A kế thừa class B => A is a B

22 Lớp con kế thừa được những tài sản nào(thuộc tính, phương thức) của lớp cha ?

* Lớp con kế thừa đc tất cả ngoại trừ các thuộc tính , phương thức được khai báo = private và constructor của lớp cha.

23 Lớp Object là gì

Là super class của tất cả các class trong java.

24 Khái niệm đa hình

- Cùng 1 phương thức nhưng mỗi trường hợp cho ra 1 kết quả khác nhau .

25 Phân biệt Overloading và Overriding

|  |  |
| --- | --- |
| Overloading | Overriding |
| * Xảy ra trên cùng 1 class * Cùng tên nhưng khác tham số * Có thể giống hoặc khác kiểu dữ liệu trả về * Thể hiện tính đa hình lúc biên dịch * Không quan trọng access modifier | * Xảy ra ở 2 class có quan hệ kế thừa * Cùng tên cùng tham số * Phải giống kiểu dữ liệu trả về * Thể hiện tính đa hình lúc runtime * Access modifier của lớp con phải có level >= lớp cha |

26 ép kiểu là gì ? các loại ép kiểu

* Ép kiểu là gán giá trị của 1 biến có kiểu dữ liệu này cho 1 biến có kiểu dữ liệu khác
* Có 2 loại ép kiểu :
* Ép kiểu ngầm định( ép kiểu nới rộng) là ép kiểu ngầm trong chương trình để nới rộng bộ nhớ. Không gây ra mất dữ liệu . ( con -> cha)
* Ép kiểu tường minh(ép kiểu thu hẹp) là ép kiểu thủ công để thu hẹp bộ nhớ dễ gây ra mất mát dữ liệu.

(cha ->con)

\*\* Một số ví dụ: byte-> short-> int -> long ->float-> double

Ép kiểu từ string -> int : Interger.parseint;

Ngược lại: String.valueOf()

27 Tính trừu tượng là gì ?

* Là chỉ quan tâm đầu vào và đâu ra . không quan tâm đến những công việc được thực hiện bên trong .

Hay nói cách khác là khả năng ẩn đi các chi tiết triển khai, chỉ quan tâm tới kết quả, không quan tâm tới quá trình

28 Phân biệt abstract class và interface

29 So sánh Array và ArrayList

30 So sánh ArrayList và LinkedList

31 So sánh List, Set và Queue ?

32 Set là gì, các lớp triển khai của Set

33 Generic là gì

34 Ưu điểm và hạn chế khi dùng generic ?

35 Stack là gì, các phương thức của stack ?

36 Queue là gì, các class triển khai của queue ?

37 Phương thức cơ bản của queue ?

38 So sánh Comparable và Comparator, khi nào dùng cái nào ?

39 Map là gì, các class triển khai

40 Phương thức equals() và hashcode() trong Java là gì?

41 Cây nhị phân là gì

42 Trình bày các cách duyệt cây nhị phân.

43 Ngoại lệ (Exception là gì) ?

44 Phân loại Exception

45 Phân biệt Error và Exception

46 Có bao nhiêu cách để xử lý ngoại lệ (handle)

47 Một số lưu ý khi dùng try-catch

48 Serialization là gì ? lưu ý khí sử dụng Serialization