**Java String Concatenation**

* Kết nối 2 sâu lại với nhau:

+ hoặc có thể dùng phương thức concat();

Các kí tự đặc biệt trong String

* Ký tự thoát dấu gạch chéo ngược ( \) biến các ký tự đặc biệt thành các ký tự chuỗi:

\’…\’ : tạo thành ‘ ‘

\”…\” : tạo thành “ “

[\\...\\](file:///\\...\\) : tạo thành \

VD: 1. String txt = "We are the so-called \"Vikings\" from the north.";

* We are the so-called “Vikings” from the north.

2. String txt = "It\'s alright.";

* It’s alright.



**Java Math**

Math.max(x,y): Phương thức tìm ra giá trị lớn nhất giữa 2 giá trị x và y

Math.min(x,y): Phương thức tìm ra giá trị bé nhất giữa 2 giá trị x và y

Math.sqrt(x): Phương thức tìm căn bậc 2 của giá trị x

Math.abs(x): Phương thức trả về giá trị tuyệt đối của x

Math.random(): Phương thức trả về giá trị ngẫu nhiên từ 0 -> 1

**OOP**

**Java Class Methods:**

+ static: có thể truy cập mà không cần tạo đối tượng (obj).

+ public: không thể truy cập nếu không tạo đối tượng.

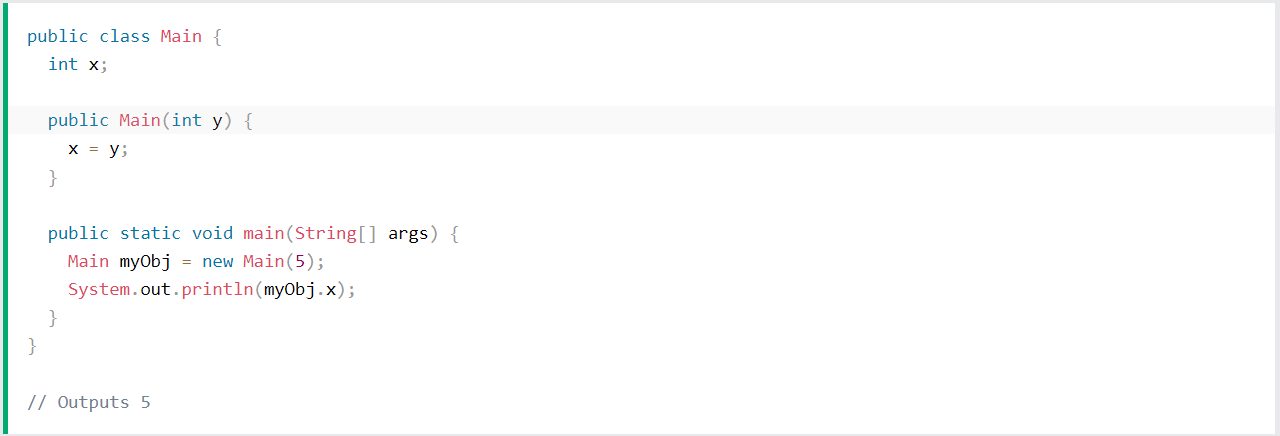
**Java Constructors:**

+ Tên của constructor phải trùng với tên của lớp(class), và không có giá trị trả về (no return)

+ Constructor được gọi chỉ khi đối tượng(obj) được khởi tạo

+ Tất cả các lớp đều có constructor mặc định: Nếu bạn không khởi tạo, Java sẽ tự khởi tạo, tuy nhiên bạn không được phép thiết lập giá trị cho các thuộc tính của đối tượng

Constructors can also take parameters, which is used to initialize attributes.



Java Modifiers (Đặt mức truy cập)

* **Access Modifiers: Kiểm soát mức độ truy cập**
* **Non-Access Modifiers**: Không kiểm soát, nhưng cung cấp các chức năng khác

ACCESS MODIFIERS

\*For Class: có thể dùng public or default

public: Lớp có thể truy cập được bởi bất kì lớp khác

default: Chỉ có thể truy cập bởi Class có cùng thư viện

\*For Attributes, Methods, Constructor:

public: Tất cả các lớp đều sử dụng được

private: Chỉ có thể sử dụng bên trong lớp khai báo

default: Chỉ có thể truy cập trong Class có cùng thư viện

protected: Có thể truy cập trong Class có cùng thư viện và các Lớp con

NON-ACCESS MODIFIERS

For Class: final or abstract

final: Lớp không thể được kế thừa bởi các lớp khác

abstract: Lớp không thể được sử dụng để tạo đối tượng(obj)

For Attributes and Methods

final: Không thể bị ghi đè hoặc sửa đổi

static: Thuộc về lớp, thay vì một đối tượng

|  |  |
| --- | --- |
| abstract | Can only be used in an abstract class, and can only be used on methods. The method does not have a body, for example **abstract void run();**. The body is provided by the subclass (inherited from). You will learn more about inheritance and abstraction in the [Inheritance](https://www.w3schools.com/java/java_inheritance.asp) and [Abstraction](https://www.w3schools.com/java/java_abstract.asp) chapters |
| transient | Attributes and methods are skipped when serializing the object containing them |
| synchronized | Methods can only be accessed by one thread at a time |
| volatile | The value of an attribute is not cached thread-locally, and is always read from the "main memory" |