

**Phan Doản Trí**

eleType [ ] nameArray = { }

nameArray = new eleType [size]

JAVASCRIPT | JAVA

Độ dài không cố định | cố định

Cung cấp nhiều hàm hỗ trợ | chỉ có thuộc tính length

Giá trị mặc định: null | theo kiểu dữ liệu đã khai báo

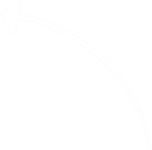
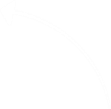
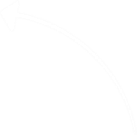
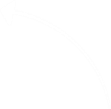
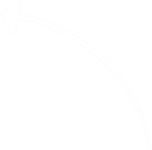
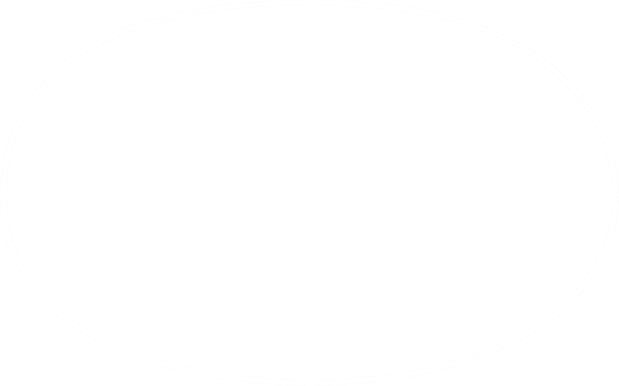
Không cần cung cấp độ dài khi khởi tạo | bắt buộc cung cấp

**Java** là NNLT hướng đối tượng, an toàn và chặt chẽ, để phát triển phần mềm, ứng dụng, wed trên di động, máy tính. **Write once, run anywhere:** Jv có tính độc lập, có thể chạy trên các nền tảng khác nhau mà không phải dịch lại, thông qua các bước thông dịch ra ngôn ngữ máy.

**Kiểu dữ liệu** Nguyên thủy: byte (0/1b), short (0/2b), int (0/4b), long (0L,8b), float (0.0f/4b), double (0.0d/8b), boolean (F/1b), char (/u0000/2b).

Đối tượng: String, Method, Array

Module Two

C:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.png

**NGÔN NGỮ JAVA**

**Method** là tập hợp các câu lệnh thực hiện 1 nhiệm vụ nhất định

Khai báo: AM + non/static + datatype (void:ko trả về gì + methodName {body}

**Construction**

- Không có kiểu dữ liệu trả về

- Tên trùng class

- Cung cấp mặc định 1 cons ko tham số

- Gọi 1 lần khi khởi tạo object

- Gọi ngầm định – “new”

- Không đi kèm abstract, final, static

**ACCESS MODIFIER**

**INHERITANCE**

**Class**: tập hợp 1 nhóm các đối tượng. là khuôn mẫu để tạo ra object, không thể đặt tên trùng class, dùng “class” để khai báo

**Object**: thể hiện cụ thể object, mỗi object chứa giá trị thuộc tính, hành vi khác nhau để phân biệt, có thể trùng tên object (khác file), dung “new”

**CLASS & OBJECT**

**ARRAY & METHOD**

**Inheritance**: lớp con thừa kế toàn bộ thuộc tính, phương thức của lớp cha. Có thể mở rộng, bổ sung thêm các thuộc tính, phương thức riêng của nó (giảm sự lặp lại, độ nặng, tái sử dụng

- Ko đa kế thừa, chỉ kế thừa không phải private

- Ko kế thừa Cons, phải gọi thông qua super( )

- Lớp con có thể mở rộng thêm các ttin, pthuc mới

**JDK** (Java Development Kit) cung cấp công cụ phát triển Java. **JRE** (Java Runtime Enviroment) tạo môi trường để chạy Java. **JVM** (Java Virtual Machine) là máy ảo Java, thông dịch ra ngôn ngữ máy để hiển thị cho người dùng

**Super(** ): gọi tới cons của lớp cha

**Super**: gọi tới các thành phần ko phải private của cha

**Final:** để tạo hằng số, ko cho phép ghi đè, tạo lớp vô hình

**toString:** hiển thị thông tin đối tượng

**Lớp Object:** là lướp cha của mọi lớp trong Java

**Ép kiểu:** việc gán giá trị kiểu dữ liệu này sang kiểu dữ liệu khác (toán tử casting)

**Ngầm định**: do ctrinh ngầm định kiểu, ko gây mất dữ liệu (ngthuy: nhỏ->lớn, dotuong: cha->con)

**Tường minh**: do LTV tự qdinh có thể sinh ra mất dữ liệu (ngthuy: lớn->nhỏ, đtuong: con->cha)

**Encapsulation**: kĩ thuật ẩn thông tin, chỉ hiển thị thông tin cần thiết. bảo vệ sự toàn vẹn của dữ liệu

**Private**: trong class

**Default**: class,package

**Protected**: class, package, lớp con có quan hệ kế thừa

**Public**: trong toàn bộ dự án

**Local:** trong phương thức, cons, block. Hủy khi kết thúc phương thức, cons. Không AM. Stack. Khai báo giá trị trước

**Instance**: ngoài. Đối tượng bị hủy (null). Có AM. Heap. Giá trị phụ thuộc kiểu dữ liệu

**Static**: ngoài. Hủy khi dừng chương trình. Có AM. Static area (heap). Giá trị phụ thuộc kiểu dữ liệu

**Ràng buộc Static**: static chỉ gọi được static, ko gọi được non-static. Non-static có thể gọi static. Dùng khối static để khởi tạo giá trị cho biến static. Không đi cùng this( ); super ( )

**Method**

- Có or không (void)

- Có thể (ko nên)

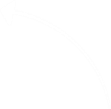
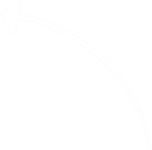
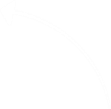
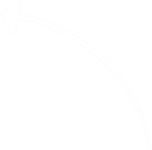
-Không cung cấp

- Gọi nhiều lần (diễn tả hành động, chức năng)

- Gọi tường minh – tên method

- Có thể đi kèm

**OOP** (Object Oriented Programming) là kĩ thuật lập trình, cho phép LTV ánh xạ các đối tượng thực tế thành các đối tượng trong chương trình

C:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.pngC:\Users\admininstall\AppData\Local\Microsoft\Windows\INetCache\Content.Word\arrow s9.png

**Interface**: là giao diện chỉ chứa các abs method, cons được dung để tạo các lớp dung chung

-Không có construction

-Các thuộc tính mặc định sẽ là public static final (static ko đi cùng với abs)

-Các method mặc định sẽ là abs method

-1 interface có thể có extends 1 or nhiều interface khác

-1 class có thể inplement 1 or nhiều interface

-Class khi implement interface bắt buộc phải implement toàn bộ abs method. Abs class thì không cần

=>thường dung trong mối quan hệ has-a, can-do



**Abstract class**: lớp có tính trừu tượng cao tới mức ko thể khởi tạo được đối tượng

-Ko thẻ tạo object nhưng vẫn có cons (để lớp con có thể sử dụng)

-1 lớp chứa abs method thì bắt buộc đó phải là abs class nhưng ngược lại thì không

-1 class khi extends 1 abs class thì bắt buộc phải override hết tất cả các abs method

-1 abs class khi kế thừa 1 abs class khác thì ko cần phỉa override các abs method

-Private/final ko thể đi cùng với abs

-1 abs class chỉ có thể extends 1 abs class

**Trừu tượng:** cho phép lượt bỏ đi các thuộc tính ko cần thiết. Chỉ quan tâm đầu vào, đầu ra, ko quan tâm đến các chi tiết phát triển khai bên trong (rút gọn mã nguồn, tang hiệu năng, giảm độ phức tạp)

**Override:**

- Trong 2 class có quan hệ kế thừa

- Pthuc phải cùng tên, cùng tham số(danh sách, kiểu dữ liệu trả về)

- AM lớp con lớn or bằng lớp cha

- Đa hình tại runtime

**Overload:**

- Trong cùng 1 class

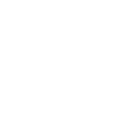
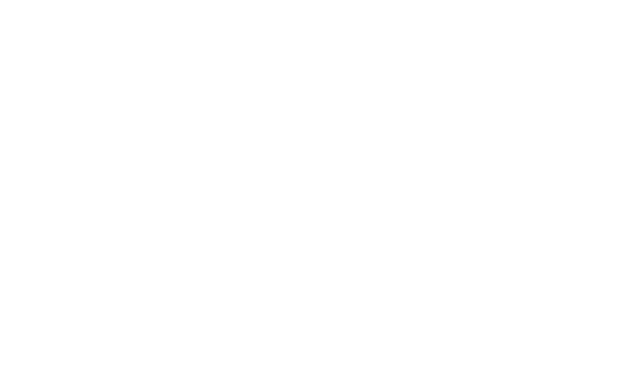
- Pthuc phải cùng tên, nhưng khác tham số(danh sách, số lượng,vị trí)

- Không quan trọng AM

- Đa hình tại compile

**Tính đa hình:** là tính chất cho phép cùng một hành vi có nhiều sự thể hiện khác nhau, tùy vào thời điểm, ngữ cảnh

**ABSTRACTION**



**POLYMORPHISM**



Module Two

**MIND MAP**

**Huy Nguyễn Hữu**