**Java**

Nội dung

[1. JVM hoạt động thế nào? 2](#_Toc151367190)

[2. Garbage Collection 5](#_Toc151367191)

[3. Quản lý bộ nhớ 5](#_Toc151367192)

[4. Gradle 5](#_Toc151367193)

[5. Ant 6](#_Toc151367194)

# JVM hoạt động thế nào?

A diagram of a diagram

Description automatically generated

JVM (Java Virtual Machine) là 1 máy ảo java – trình thông dịch của java. Nó cung cấp môi trường để code java có thể được thực thi, chương trình java khi biên dịch các file \*.class chứa byte code, các file này sẽ được JVM thực thi chuyển byte code thành mã máy tương ứng với từng hệ điều hành và phần cứng khác nhau để thực thi.

A diagram of a code

Description automatically generated

JVM là một phần của JRE – cung cấp kèm các thư viện lớp Java. Mở rộng: JDK là một bộ công cụ phát triển Java

A blue and black box with black text

Description automatically generatedA screenshot of a computer

Description automatically generated

Cấu trúc bên trong của JVM:

A diagram of a program

Description automatically generated

* **Classloader**: Hệ thống phụ tải các file class
* **Class area**: Vùng chứa các class và cung cấp các class nền tảng (Object, …) cho phép mở rộng hoặc ghi đè nó
* **Heap**: Sử dụng cấp bộ nhớ cho đối tượng
* **Stack**: Vùng nhớ dùng để lưu trữ các tham số và biến local của phương thức khi nó được gọi
* **Pc register**: Chứa địa chỉ của các máy ảo java hiện đang được thực hiên (đọc hệ điều hành nâng cao để biết thêm)
* **Native Method Stack**: Nơi chứa tất cả các method native (triển khai bởi ngôn ngữ khác) trong chương trình
* **Execution engine**: Gồm
  + Một bộ xử lý ảo
  + Phiên dịch
  + Just – In – Time (JIT): Sử dụng để cải thiện hiệu suất bằng cách dịch cùng lúc

# Garbage Collection

Garbage nghĩa là các đối tượng không còn được tham chiếu nữa. Bộ gom rác (Garbage Collection) trong java được sử dụng để thực hiện quá trình tự động khôi phục lại bộ nhớ không được sử dụng tại runtime một các tự động (xóa các đối tượng trong heap).

Phương thức **finalize()** được gọi mỗi lần trước khi đối tượng được thu gom rác (đối tượng không còn được sử dụng). Được định nghĩa trong lớp Object. Được sử dụng để thực hiện xử lý dọn dẹp (chỉ dọn nếu được tạo bằng từ khóa **new**)

Phương thức **gc()** được sử dụng để gọi bộ thu gom rác để thực hiện quá trình dọn dẹp. Được cài đặt trong System và Runtime.

* + - **Finalize()** gom lại và **gc()** tới dọn

**A computer screen shot of a program code

Description automatically generated**

# Quản lý bộ nhớ

Không giống với một số ngôn ngữ như C, C++,… Java được hỗ trợ bộ nhớ tự động mà không yêu cầu người dùng làm. Điều đó được thực hiện bởi JVM.

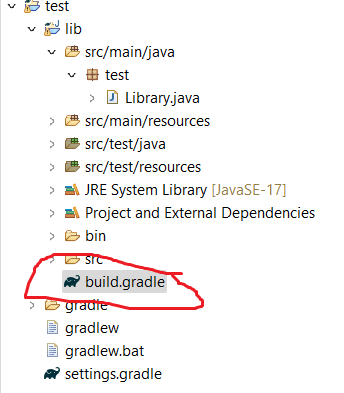
# Gradle

Là một công cụ quản lý phụ thuộc và xây dựng phần mềm được sử dụng trong các dự án Java. Nó được thiết kế để tự động hóa quá trình build và quản lý các thư viện, nhằm tối ưu hóa công đoạn biên dịch, đóng gói và thuận tiện hơn cho việc phát triển phần mềm, Gradle kết hợp các ưu điểm của Apache Ant và Apache Maven, đồng thời đem lại các cải thiện mới thông qua việc sử dụng Groovy (Một trong những ngôn ngữ JVM) trong việc giải quyết bài toán đóng gói.

Không chỉ hỗ trợ ở java mà còn nhiều ngôn ngữ khác như: Scala, Python, C/C++, …

Cách sử dụng:

* + Trước hết hãy đảm bảo JDK của bạn đã có phiên bản grandle tương thích.
  + Sau khi tạo grandle



# Ant

Còn gọi là Apache Ant. Đây là mộ trong những dự án nguồn mở, và là công cụ của phần mềm tự động hóa các quy trình build code. Sở dĩ gọi là Apache Ant vì phần mềm này được bắt nguồn từ dự án Apache Tomcat dùng để thay thế Make Build của Unix. **Ant sử dụng XML** để có thể mô tả đồng thời xây dựng mã phụ thuộc vào nó.

Ant – Another Neat Tool. Phải gọi cụ thể công việc cho Ant

Các bước khởi tạo:

* + Đảm bảo đã tồn tại bản ANT trong IDE:

A screenshot of a computer

Description automatically generated

* + Tiến hành tạo một file xml có nội dung như sau:

A screen shot of a computer code

Description automatically generated

* + Tạo Project ant link file xml tới file vừa tạo trên

Giải thích:



A screenshot of a computer code

Description automatically generated