**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA TOÁN-CƠ –TIN HỌC**

**\*\*\*\*\*\***



**BÁO CÁO ĐỒ ÁN**

**Đề tài: Tìm hiểu cấu trúc AJVM**

**Sinh viên: Nguyễn Văn Thạo - 18001079**

**Nguyễn Thị Hà – 18001036**

**Đặng Hà My - 18001063**

**Lớp: K63A5 MT&KHTT CLC**

**Giáo viên hướng dẫn: Bùi Sỹ Nguyên**

*Hà Nội, Ngày 18 tháng 01 năm 2021*

**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA TOÁN CƠ TIN**

**\*\*\*\*\*\***

**BÁO CÁO ĐỒ ÁN MÔN PYTHON**

**Đề tài: Xây dựng trang web bán đồ ăn**

Sinh viên thực hiện: **Trần Đức Thịnh – 18001080**

**Nguyễn Thị Hà – 18001036**

**Đặng Hà My -18001063**

**LỜI CẢM ƠN**

Để có thể hoàn thành tốt bài báo cáo đồ án này em xin gửi lời cảm ơn sâu sắc nhất đến thầy Bùi Sỹ Nguyên. Thầy đã giúp đỡ và tạo điều kiện, quan tâm chúng em trong suốt quá trình làm báo cáo và đồ án.

Sau bài báo cáo này đã giúp chúng em biết cách làm việc nhóm, hiểu được kiến trúc và các thành phần cấu tạo của một máy ảo JVM, bên cạnh đó còn biết cách sử dụng một JVM đơn giản. Trong quá trình làm báo cáo khó tránh khỏi những sai sót, rất mong thầy bỏ qua. Đồng thời do trình độ của bọn em còn hạn chế nên bài báo cáo khó tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp của Thầy để chúng em được củng cố thêm kiến thực và sẽ hoàn thành tốt bài báo cáo hơn.

***Em xin chân thành cảm ơn!***

MỤC LỤC

[ĐẶT VẤN ĐỀ 3](#_Toc61845935)

[**1.** **Mục đích-yêu cầu** 3](#_Toc61845936)

[**2.** **Đối tượng nghiên cứu** 3](#_Toc61845937)

[**3.** **Phương pháp nghiên cứu** 3](#_Toc61845938)

[**4.** **Phạm vi nghiên cứu** 3](#_Toc61845940)

[**5.** **Kết cấu đề tài** 3](#_Toc61845941)

[**6.** **Kết quả nghiên cứu** 4](#_Toc61845942)

[**NỘI DUNG BÁO CÁO** 4](#_Toc61845943)

[**I-** **Giới thiệu JVM** 5](#_Toc61845944)

[***1.1.*** ***Điều kiện cơ bản.*** 6](#_Toc61845945)

[***1.2.*** ***Flash Platform and ActionScript 3.0.*** 6](#_Toc61845946)

[**II-** **Related Work.** 6](#_Toc61845947)

[***2.1.*** ***Giả lập các nền tảng khác được triển khai trong Flash.*** 7](#_Toc61845948)

[**III-** **Thực hiện AJVM trong ActionScript 3.0.** 8](#_Toc61845949)

[***3.1.*** ***Máy ảo vs Thư viện tiêu chuẩn.*** 8](#_Toc61845950)

[***3.2.*** ***Tổng quan cấp cao về AJVM.*** 8](#_Toc61845951)

[***3.3.*** ***Thực thi bytecode trong AJVM.*** 10](#_Toc61845952)

[**IV-** **Nhận xét bổ sung về triển khai ActionScript 3.0.** 11](#_Toc61845953)

[***4.1.*** ***Thực hiện xử lý ngoại lệ.*** 11](#_Toc61845954)

[***4.2.*** ***Thu gom rác thải.*** 11](#_Toc61845955)

[***4.3.*** ***Đa luồng.*** 12](#_Toc61845956)

[***4.4.*** ***Giao diện gốc.*** 12](#_Toc61845957)

[**V-** **HƯỚNG DẪN CHẠY** 13](#_Toc61845958)

# ĐẶT VẤN ĐỀ

1. **Mục đích-yêu cầu**

* Mục đích: tìm hiểu và nghiên cứu cấu trúc, các thành phần của máy ảo JVM, biết cách sử dụng JVM đơn giản.
* Yêu cầu: nắm vững những kiến thức cơ bản về công nghệ phần mềm và tìm hiểu rõ kiến thức AJVM.

1. **Đối tượng nghiên cứu**

Cấu trúc AJVM

1. **Phương pháp nghiên cứu**

* Internet
* Adobe AIR SDK
* IntelliJ

1. **Phạm vi nghiên cứu**

Qua tờ báo tìm hiểu về cấu trúc AJVM.

1. **Kết cấu đề tài**

Gồm 5 phần:

I: Giới thiệu JVM

II: Related Work

III: Thực hiện AJVM trong ActionScript 3.0

IV: Nhận xét bổ sung về triển khai ActionScript 3.0

V: Hướng dẫn chạy

1. **Kết quả nghiên cứu**

* Biết cách làm việc nhóm, hiểu được cấu trúc, các thành phần của máy ảo JVM, biết cách sử dụng JVM đơn giản.

# NỘI DUNG BÁO CÁO

* Máy ảo hiện đại (JVM) được triển khai trong ActionScript 3.0

+ ActionScript: ngôn ngữ lập trình phát triển bởi Adobe sử dụng Flash để tạo các hình động.

ngôn ngữ lập trình khách quan hỗ trợ biên dịch.

+ Môi trường thực thi của máy ảo ActionScript dùng để xây dựng máy ảo Java.

+ JVM là máy ảo chạy bytecode Java.

* Thực hiện AJVM với ActionScript Virtual Machine(AVM) giúp nó có thể sử dụng mã java trong các ứng dụng nó được viết trong ActionScript 3.0.

1. **Giới thiệu JVM**

* Đặc điểm JVM: tăng hiệu suất của máy ảo thường được cài đặt trong các ngôn ngữ lập trình cấp thấp, tích hợp chặt chẽ với hệ điều hành và phần cứng.
* Mục đích: tăng khả năng tái sử dụng mã bằng cách sử dụng các thư viện java trong ứng dụng flash và cung cấp các khái niệm JVM.

APPLICATION

MacOS

JVM

Linux

Window

* 1. ***Điều kiện cơ bản.***
* Có 2 loại máy ảo chính :

+ Trình giả lập: là giải pháp cho phép thực thi (trong môi trường bị cô lập) toàn bộ hệ điều hành và phần mềm khác đươc thiết kế cho một nền tảng cụ thể do ảo hóa phần cứng.

+ Thông dịch viên: là phần mềm có khả năng thực thi mã nhị phân, mã hóa được biên dịch trước.

* 1. ***Flash Platform and ActionScript 3.0.***
* ActionScript 3.0 là ngôn ngữ lập trình bắt buộc, được biên dịch thành bytecode được thực thi trên AVM2 máy ảo, đơn luồng(điều khiển bởi các sự kiện do Flash Player kích hoạt) nên không hỗ trợ chặn phương pháp, với quản lý tự động, không hỗ trợ các lớp chung chung và các lớp ẩn danh.
* Adobe Flash: nền tảng đa phương tiện cho phép các tính năng mới trong các trang web: hình ảnh động, video streaming…
* Các ứng dụng Flash được xuất bản dưới dạng tệp SWF.

1. **Related Work.**

Có hai khía cạnh chính của sự đóng góp trong công việc:

* Triển khai máy ảo Java trong môi trường không chuẩn
* Mô phỏng nền tảng Java bên trong Flash Player.

Một số ấn phẩm về chủ đề này của các JVM:

+ Jamiga: thực thi Java ứng dụng trên máy tính Amiga.

+ JC: tất cả các lớp Java đều được dịch nhanh chóng đến mã nguồn bằng ngôn ngữ C và sau đó được biên dịch sang mã gốc.

+ Squawk: triển khai JVM cho Java ME(Micro Edition) cho các hệ thống nhúng và thiết bị di động nhỏ.

+ GNU Classpath: Thư viện chuẩn Java được phân phối theo giấy phép GNU cung cấp một cơ sở để xây dựng các JVM riêng.

+ JOP: Bộ xử lý được tối ưu hóa Java là một triển khai phần cứng của JVM. Do nhỏ kích thước của bộ xử lý được sử dụng, nó có thể được thực hiện trong FPGA chi phí thấp. Bộ xử lý thực thi bytecode trực tiếp mà không cần biên dịch trong thời gian cũng như để phân tích cú pháp / diễn giải các tệp lớp.

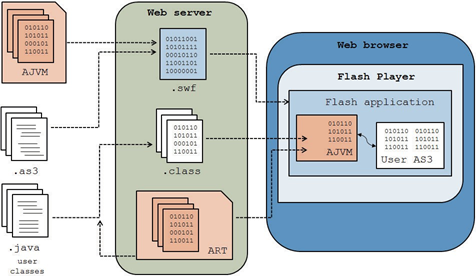
+ Sable VM: sử dụng tính năng diễn giải hiện đại kỹ thuật. Mục tiêu của nó là hợp lý nhỏ, nhanh chóng, và tuân thủ các thông số kỹ thuật khác nhau.

* 1. ***Giả lập các nền tảng khác được triển khai trong Flash.***
* **FC64** – (trình giả lập Flash Commodore 64 - FC64): một trình giả lập cấp thấp, đầy đủ chức năng của Commodore 64 cho phép người dùng thực thi các ứng dụng được thiết kế cho Commodore 64 cũng như để viết mã trong BASIC ngôn ngữ lập trình.
* **FlashZXSpectrum48k** - là một giải pháp tương tự như FC64. Sự khác biệt là nó mô phỏng Sinclair ZX Spectrum platform.
* **AminNes** - Trình giả lập Flash NES là trình mô phỏng của Hệ thống giải trí Nintendo - Bảng điều khiển chơi game 8 bit được trang bị RAM 2kB và bộ nhớ video 2KB. Trình giả lập hỗ trợ MOS 6502 và đảm bảo chất lượng video hiển thị cao nhất.
* **Flip8** - Flip8 là một trình giả lập CHIP-8. CHIP-8 là một máy ảo được thiết kế trong những năm 70 được sử dụng để diễn giải ngôn ngữ lập trình. Nó từng được cài đặt trên máy tính đồ họa. Flip8 là một trình giả lập có thể thực thi bytecode bao gồm 35 thao tác khác nhau.

1. **Thực hiện AJVM trong ActionScript 3.0.**
   1. ***Máy ảo vs Thư viện tiêu chuẩn.***

Phần quan trọng của việc thực hiện bất kỳ Máy ảo Java là một thư viện tiêu chuẩn được trình bày bởi các đặc điểm sau:

* *Thư viện AJVM* - Thư viện ActionScript 3.0 chứa các tệp thực thi của AJVM, sẵn sàng được sử dụng trong bất kỳ dự án Flash hoặc Flex
* *Thư viện tiêu chuẩn ART***-** thư viện cụ thể để triển khai VM như nó mong đợi triển khai bản địa của các phương thức đã chọn trong VM thực hiện một tiêu chuẩn thư viện có tên ART (ActionScript Runtime) dựa trên GNU Classpath.
  1. ***Tổng quan cấp cao về AJVM.***



AJVM có sẵn dưới dạng một đối tượng được tạo bởi mã của lập trình viên bên trong ứng dụng Flash của họ. Một số trường hợp riêng biệt của AJVM trong một ứng dụng lưu trữ flash có thể được tạo.

*Bắt đầu và dừng các thể hiện AJVM* - các thể hiện AJVM là các đối tượng độc lập được tạo, bắt đầu và dừng lại trong những khoảnh khắc được lựa chọn tùy ý trong suốt thời gian tồn tại của ứng dụng Flash máy chủ.

*Cấu hình classpath* - AJVM cần classpath để được định nghĩa biết nơi để tìm các lớp Java đã biên dịch. AJVM thêm một tiền tố với tên của máy chủ web riêng để soạn các URL dùng để định vị lớp mã.

*Chỉ định lớp chính* - Mỗi JVM bắt đầu thực thi mã của người dùng theo phương thức:

“ ***public static void main (String args []);***” được xác định trong lớp được chỉ định làm đối số của JVM

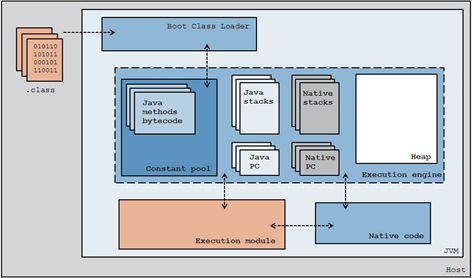
*Cấu hình vòng đời của VM* -, Flash là một môi trường đơn luồng. Do đó, AJVM phải làm việc trong rất các khoảng thời gian ngắn chia cho các lần tạm dừng đủ dài để xử lý các sự kiện không có Java khác và kết xuất các sự kiện tiếp theo khung của hoạt hình Flash. Đã có những triển khai logic tương tự cho AJVM . Trong tương lai sẽ có những thay đổi giúp AJVM để nó tự động thay đổi tham số tùy thuộc vào độ phức tạp của Java và / hoặc mã Flash và sức mạnh tính toán có sẵn.

*Triển khai các phương pháp gốc* -AJVM sử dụng mã gốc để triển khai ART, do đó cho phép người dùng khai báo và triển khai các phương thức gốc của riêng mình - được viết bằng ActionScript 3.0

*Định cấu hình trình ghi nhật ký* – cung cấp nền tảng có thể được người dùng sử dụng để thử nghiệm và tìm hiểu về kiến trúc của máy ảo

* 1. ***Thực thi bytecode trong AJVM.***

AJVM tiếp cận hoàn toàn khách quan đối với chức năng chính của VM đang thực thi mã bytecode.



Mô-đun thực thi của JVM truyền thống hoạt động trên hệ nhị phân các lệnh được lưu trong bộ nhớ vùng phương thức và được biểu diễn theo cách không thay đổi so với mã bytecode đã biên dịch.

Trong mã bytecode của Java một người cần giải thích hướng dẫn để hiểu bao nhiêu byte sau lệnh là đối số của lệnh, điều này cần thiết để lấy phần đầu của lệnh tiếp theo. Các đối số của lệnh có thể tham chiếu đến các lớp, phương thức hoặc trường được giải quyết dưới dạng các mục nhập trong nhóm hằng số .

Việc xây dựng biểu diễn khách quan của lớp Java - Xây dựng cây phân tích cú pháp, là một đối tượng biểu diễn các phần tử trong tệp lớp.



* Xây dựng biểu diễn hướng đối tượng của bytecode.

+ Tiếp cận mã máy được biểu diễn dưới dạng một chuỗi lệnh đơn chiều được thực thi từng cái một. Trong thực tế, con trỏ lệnh thay đổi phức tạp hơn do rẽ nhánh hướng dẫn, bước nhảy có điều kiện hoặc không điều kiện. Mã hoạt động được ánh xạ trên các lớp hướng dẫn tương ứng.

1. **Nhận xét bổ sung về triển khai ActionScript 3.0.**
   1. ***Thực hiện xử lý ngoại lệ.***

Một trong những thuộc tính của lớp đã biên dịch là trình xử lý ngoại lệ bàn. Trong AJVM không có hướng dẫn bù trừ ở tất cả, vì chúng được thay thế bằng đồ thị hướng dẫn. Mỗi phương thức có một tập hợp các đối tượng đại diện cho các trình xử lý ngoại lệ.

* 1. ***Thu gom rác thải.***

*Trình thu gom rác AVM2* - vì các đối tượng Java bên trong AJVM được đại diện bởi ActionScript’s đối tượng mà chúng được quản lý bởi AVM2’s Garbage Collector.

* 1. ***Đa luồng.***

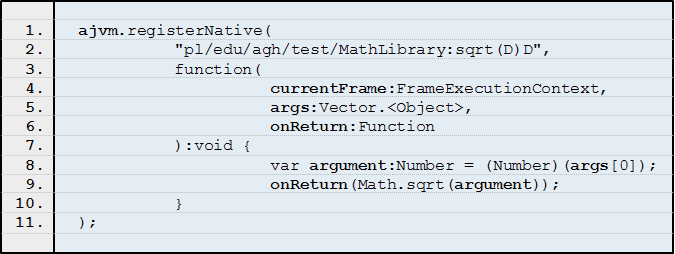
AJVM không có quyền truy cập vào các chuỗi hệ điều hành cấp thấp vì nó bị giới hạn bởi Flash Player, nó được chạy trong đó thực thi mã của ActionScript 3.0 ngay bên trong luồng của trình xử lý sự kiện. Các đối tượng mới được thêm vào bộ sưu tập mỗi khi một lệnh bytecode thực thi phương thức java.lang.

AJVM hoạt động theo chu kỳ và mỗi chu kỳ là việc thực hiện một lượng lệnh của mã bytecode của mỗi chủ đề hoạt động. Độ dài của chu kỳ AJVM là mục cấu hình AJVM.

Thực thi phương thức step một số lần dựa trên cơ sở tính toán về mức độ ưu tiên của chủ đề.

* 1. ***Giao diện gốc.***

Giao diện gốc của AJVM cho phép người dùng cung cấp việc triển khai các lớp Java của chúng dưới dạng các hàm ActionScript 3.0.



Các đối số của hàm được sử dụng để chuyển các phần tử sau từ giao diện gốc của AJVM đến phương pháp của người dùng

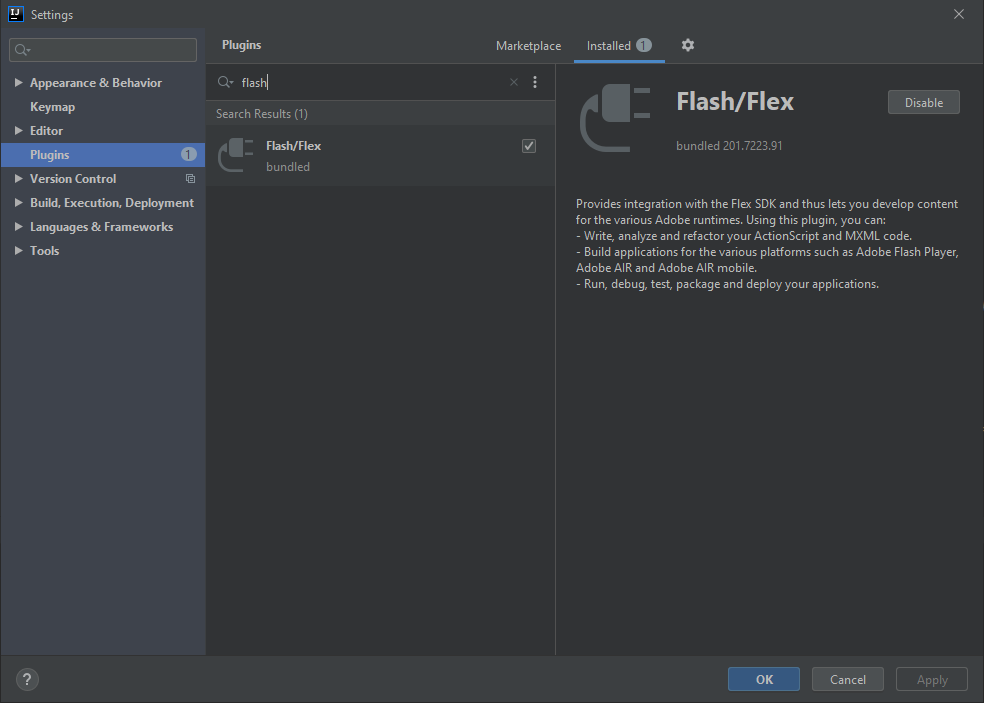
*• Bối cảnh của khung thực thi hiện tại trong khung ngăn xếp thực thi* - tham chiếu đến trường hợp của lớp FrameExecutionContext cung cấp quyền truy cập vào các thuộc tính của chuỗi Java hiện tại như cũng như ngăn xếp và vùng cục bộ của ngăn xếp khung.

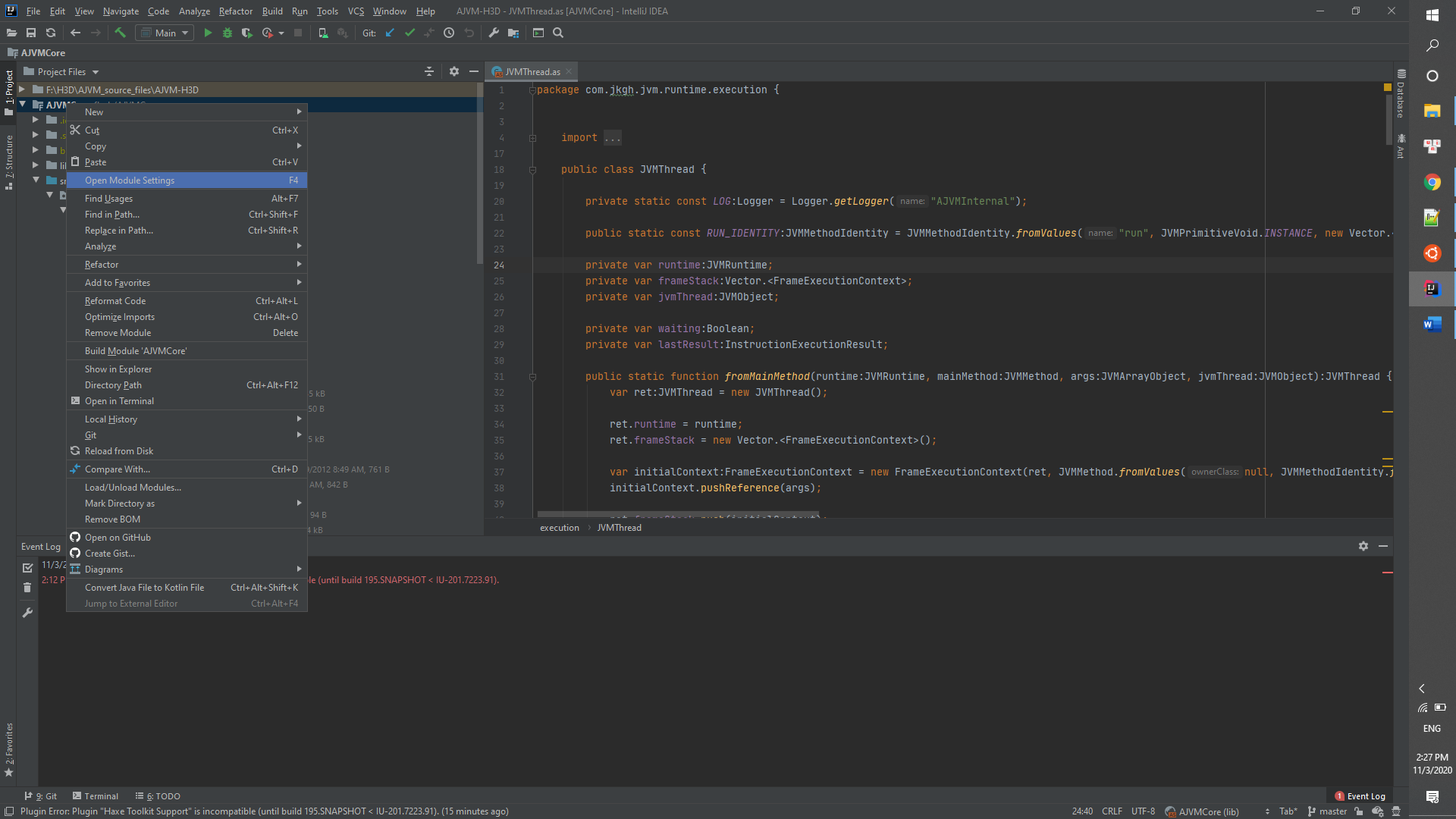
• Giá trị của các đối số được truyền cho phương thức Java. args chứa 0 hoặc nhiều đối số được chuyển đến phương pháp gốc bằng mã Java.

• Gọi lại trả về từ phương thức gốc- Mỗi triển khai gốc sẽ có trao lại quyền kiểm soát cho AJVM ngay sau khi có được vì Flash Player sử dụng một luồng duy nhất. Thực thi trả lại chức năng theo cách được mô tả cho phép một nhà phát triển để triển khai các phương pháp chặn

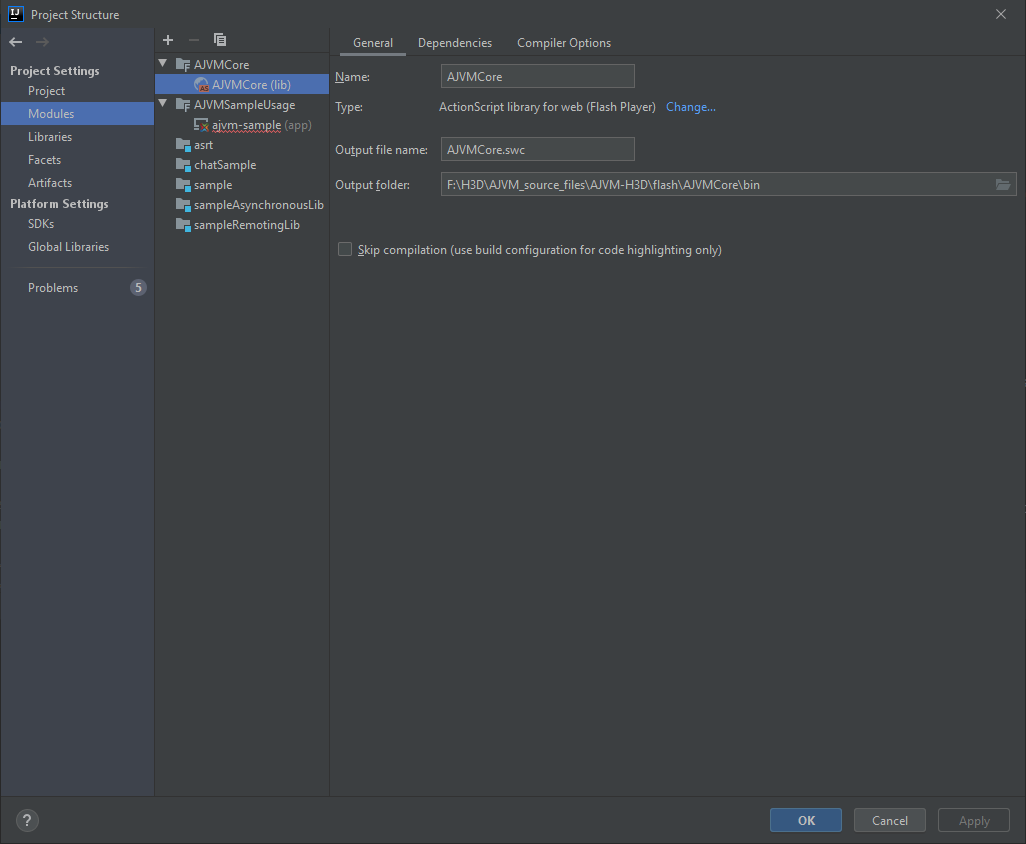
1. **HƯỚNG DẪN CHẠY**

Bước 1: Cài Adobe AIR SDK. Download bộ cài tại đây:  
<https://get.adobe.com/air/>

Bước 2: Vào Settings của IntelliJ, vào mục Plugins và cài Plugin có tên là Flash/Flex  


Bước 3: Vào Open Module Settings   


Sau đó chọn module cho AJVM Core có kiểu ActionScript Library for Web



Rồi chọn setting cho ajvm-sample là Flex application for desktop (Adobe AIR)

1. 