**BÁO CÁO THỰC HÀNH**

1. **Thông tin chung**
2. *Tên nhóm- họ và tên thành viên : Nhóm 11 -Nguyễn Thanh Liêm*
3. *Lớp:2022DHHTTT01 Khóa: 17*
4. *Học phần:Lập trình Java Bài:*
5. *Ngày thực hành: 19/9/2024 Phòng TH*
6. *Ca TH: Ca1 Phòng máy:*
7. **Tóm tắt lý thuyết**

#### Java Tutorial

Java là gì?

Java là một **ngôn ngữ lập trình** và một **nền tảng**. Java là một ngôn ngữ lập trình cấp cao, mạnh mẽ, hướng đối tượng và an toàn.

Ứng dụng

* Các ứng dụng dành cho máy tính để bàn như trình đọc acrobat, trình phát đa phương tiện, phần mềm chống vi-rút, v.v.
* Các ứng dụng web như irctc.co.in, javatpoint.com, v.v.
* Các ứng dụng doanh nghiệp như ứng dụng ngân hàng.
* Di động
* Hệ thống nhúng
* Thẻ thông minh
* Robotics
* Trò chơi, v.v.

Các tính năng:

1. [Giản dị](https://www.javatpoint.com/features-of-java#Simple)
2. [Hướng đối tượng](https://www.javatpoint.com/features-of-java#Object-Oriented)
3. [Portable](https://www.javatpoint.com/features-of-java#Portable)
4. [Nền tảng độc lập](https://www.javatpoint.com/features-of-java#Platform-independent)
5. [Bảo đảm](https://www.javatpoint.com/features-of-java#Secured)
6. [Mạnh mẽ](https://www.javatpoint.com/features-of-java#Robust)
7. [Kiến trúc trung lập](https://www.javatpoint.com/features-of-java#Architecture-neutral)
8. [Giải thích](https://www.javatpoint.com/features-of-java#Interpreted)
9. [Hiệu suất cao](https://www.javatpoint.com/features-of-java#High-Performance)
10. [Đa luồng](https://www.javatpoint.com/features-of-java#Multithreaded)
11. [Phân phối](https://www.javatpoint.com/features-of-java#Distributed)
12. [Năng động](https://www.javatpoint.com/features-of-java#Dynamic)

*Ví dụ về chương trình Java:*

class Simple{

    public static void main(String args[]){

     System.out.println("Hello Java");

    }

}

**Sự khác biệt giữa JDK, JRE và JVM**

JVM

JVM (Java Virtual Machine) là một máy trừu tượng. Nó được gọi là máy ảo vì nó không tồn tại về mặt vật lý. Nó là một đặc tả cung cấp một môi trường thời gian chạy trong đó Java bytecode có thể được thực thi. Nó cũng có thể chạy các chương trình được viết bằng các ngôn ngữ khác và được biên dịch thành Java bytecode.

JVM có sẵn cho nhiều nền tảng phần cứng và phần mềm. JVM, JRE và JDK phụ thuộc vào nền tảng vì cấu hình của mỗi [hệ điều hành](https://www.javatpoint.com/os-tutorial) khác nhau. Tuy nhiên, Java là nền tảng độc lập. Có ba khái niệm về JVM: đặc tả, triển khai và ví dụ.

JVM thực hiện các nhiệm vụ chính sau:

* Tải mã
* Xác minh mã
* Thực thi mã
* Cung cấp môi trường thời gian chạy

JRE

JRE (Java Runtime Environment). Nó cũng được viết là Java RTE. Java Runtime Environment là một tập hợp các công cụ phần mềm được sử dụng để phát triển các ứng dụng Java. Nó được sử dụng để cung cấp môi trường thời gian chạy. Đó là việc thực hiện JVM. Nó tồn tại về mặt vật lý. Nó chứa một tập hợp các thư viện + các tệp khác mà JVM sử dụng trong thời gian chạy.

Việc triển khai JVM cũng được tích cực phát hành bởi các công ty khác ngoài Sun Micro Systems.



JDK

JDK (Java Development Kit) là một môi trường phát triển phần mềm được sử dụng để phát triển các ứng dụng và [applet](https://www.javatpoint.com/java-applet) Java. Nó tồn tại về mặt vật lý. Nó chứa các công cụ phát triển JRE +.

JDK là một triển khai của bất kỳ một trong những Nền tảng Java dưới đây do Tập đoàn Oracle phát hành:

Nền tảng Java phiên bản tiêu chuẩn

Nền tảng Java phiên bản doanh nghiệp

Nền tảng Java phiên bản Micro

JDK chứa một Máy ảo Java riêng (JVM) và một vài tài nguyên khác như trình thông dịch / trình tải (java), trình biên dịch (javac), trình lưu trữ (jar), trình tạo tài liệu (Javadoc), v.v. để hoàn thành việc phát triển Ứng dụng Java.



# Biến Java

Có ba loại biến trong [Java](https://www.javatpoint.com/java-tutorial):

* biến cục bộ
* biến phiên bản
* biến tĩnh

# Kiểu dữ liệu trong Java

Có hai loại [kiểu dữ liệu trong Java](https://www.javatpoint.com/java-data-types): nguyên thủy và không nguyên thủy.

* -8 loại kiểu nguyên thủy: boolean, byte, char, short, int, long, float, double

**Toán tử trong Java**

* Toán tử đơn nhất,
* Toán tử số học,
* Người điều khiển ca,
* Toán tử quan hệ,
* Toán tử bitwise,
* Toán tử logic,
* Toán tử bậc ba và
* Toán tử nhiệm vụ.

#### Control Statements

[**Câu lệnh kiểm soát Java**](https://www.javatpoint.com/control-flow-in-java)

Decision Making statements

- if

- switch

Loop statements

- do while

- while

- for

- for-each

Jump statements

- Break

- continue

[**Java Bình luận**](https://www.javatpoint.com/java-comments)

Ví dụ:

//cach1: comment

/\*cach2: comment

Comment

\*/

/\*\*

\* cach3:

\*comment

\*comment

\*

\*/

1. **Nội dung thực hành**

***Bài 1:*** viết chương trình nhập từ bàn phím 1 chuỗi bất kì, hiện thị chuỗi lên màn hình(2 cách)

* Kết quả:

1. Thiết kế ứng dụng

|  |
| --- |
|  |

1. Kết quả chạy chương trình

|  |
| --- |
|  |

***Bài 2***: Viết chương trình nhập 2 số nguyên từ bàn phím, tính tổng tích thương hiệu và phần dư, hiện thị kết quả ra màn hình

- Kết quả:

1. Thiết kế ứng dụng

|  |
| --- |
|  |

1. Kết quả chạy chương trình

|  |
| --- |
|  |

***Bài 3:*** Viết chương trình nhập 1 dãy số nguyên từ bàn phím, tính tổng dãy số, sắp xếp dãy số theo chiều tăng dần, giảm dần, tách dãy số thành 2 dãy số con:dãy các số nguyên tố và dãy các số không nguyên tố, sau đó hiện thị các dãy số ra màn hình theo chiều ngang

* Kết quả:

1. Thiết kế ứng dụng

|  |
| --- |
|  |

1. Kết quả chạy chương trình

|  |
| --- |
|  |

***Bài 4:*** Viết chương trình nhập vào 1 ma trận có m\*n phần tử(m,n là nguyên dương nhập từ bàn phím), hiện thị ma trận ra màn hình dưới dạng bảng(có hàng , cột)

* Kết quả:

1. Thiết kế ứng dụng

|  |
| --- |
|  |

1. Kết quả chạy chương trình

|  |
| --- |
|  |

***Bài 5:*** Viết chương trình nhập 2 ma trận a, b từ bàn phím (a: m\*k, b:k\*n) tính tích 2 ma trận và hiện thị ma trận a, b và kết quả ra màn hình dưới dạng bảng

* Kết quả:

1. Thiết kế ứng dụng

|  |
| --- |
|  |

1. Kết quả chạy chương trình

|  |
| --- |
|  |

***Hà Nội, ngày 19 tháng 9 năm 2024***

***Nguyễn Thanh Liêm***