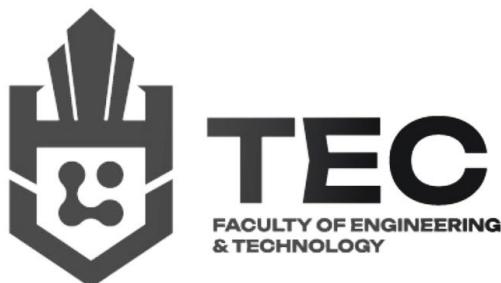


CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  
Độc lập - Tự do - Hạnh phúc

**TRƯỜNG ĐẠI HỌC HÙNG VƯƠNG TP.HCM  
KHOA KỸ THUẬT CÔNG NGHỆ**



**CODE TRÍ TUỆ, KHÍ PHÁCH VỊ VUA**  
WRITING INTELLIGENT CODE, DISPLAYING KING'S METTLE

**TÀI LIỆU THỰC HÀNH  
MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

HỌ VÀ TÊN: .....  
LỚP: .....  
MSSV: .....

(LUU HÀNH NỘI BỘ)

## NỘI QUY PHÒNG MÁY TÍNH

1. Tuyệt đối không được hút thuốc, ăn, uống, xả rác gây mất vệ sinh trong phòng máy. Phải đóng cửa khi ra vào phòng máy có máy lạnh.
2. Tuyệt đối không được mang vào phòng máy các chất hay vật dụng có khả năng gây cháy, nổ, hung khí.
3. Đi đúng giờ, đúng ca quy định của mình. Khi vào phòng máy thực hành sinh viên phải đeo thẻ sinh viên (dùng dây mang trên cổ hoặc kẹp trên túi áo)
4. Không gây ồn ào mất trật tự, không được sử dụng các trang thiết bị của phòng máy vào các mục đích khác với việc học tập và nghiên cứu.
5. Cuối ca thực hành phải sắp xếp lại ghế, chuột, bàn phím theo đúng vị trí, tắt máy tính.
6. Không di chuyển các máy móc, thiết bị... khi không được sự đồng ý của người có thẩm quyền (bộ phận quản lý phòng máy).
7. **Chuẩn bị đầy đủ bài thực hành mà Giảng viên đã quy định. Bài chuẩn bị phải viết bằng tay và nộp cho giảng viên trước khi vào phòng máy tính.**

TUỲ THEO MỨC ĐỘ VI PHẠM, SẼ CÓ HÌNH THỨC XỬ LÝ KỶ LUẬT

---

## Kế hoạch thực hành

### CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

*“Học đi đôi với hành. Bất cứ lý thuyết nào cũng cần phải được áp dụng vào thực tiễn, như thế mới có thể lưu lại trong tâm trí của chúng ta thật lâu.”*

Tuần	Nội dung
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

## Chương 0

### **GIỚI THIỆU VỀ VŨ TRỤ JAVA**

*“Cho dù bạn là người mới bắt đầu, nhà phát triển, lập trình viên hay người viết blog – kiến thức là điểm bắt đầu cho việc thực hành thực tế. Vì thế cần nắm vững lý thuyết trước khi vào thực hành”*

Trong phần thực hành này, chúng ta sẽ tìm hiểu:

- Java là gì?
- Ngôn ngữ lập trình Java có công dụng gì?
- Các tính năng của Java
- Ưu & nhược điểm của Java
- Java được sử dụng để làm gì?
- Sự khác biệt giữa C++ và Java là gì?

## I. GIỚI THIỆU VỀ JAVA

### I.1 Java là gì?

Vào năm 1991, một nhóm các kỹ sư bao gồm James Gosling, Mike Sheridan, và Patrick Naughton tại Sun Microsystems đã bắt tay cùng nhau tạo ra một ngôn ngữ lập trình cho các sản phẩm điện gia dụng. Ngôn ngữ lập trình này ban đầu được gọi bằng cái tên Greentalk, sau đó là Oak, và sau cùng là tên Java mà chúng ta biết ngày nay.

Ngôn ngữ Java được James Gosling phát triển và được Sun Microsystems cho ra mắt Java vào 1995. Hiện nay, bản quyền Java đã thuộc về Oracle. Ngôn ngữ lập trình này được thiết kế để có thể chạy trên các nền tảng khác nhau, từ máy tính cá nhân đến thiết bị di động, các máy chủ và thiết bị nhúng.

Java sử dụng cấu trúc lập trình hướng đối tượng (object-oriented programming - OOP) và được xây dựng trên cơ sở của ngôn ngữ lập trình C++. Nó cung cấp một môi

trường chạy ảo (virtual machine) gọi là Java Virtual Machine (JVM), giúp các chương trình Java có thể chạy trên nhiều nền tảng khác nhau mà không cần phải biên dịch lại.



Java là một ngôn ngữ lập trình được sử dụng rộng rãi để viết mã cho các ứng dụng web. Ngôn ngữ này là lựa chọn phổ biến của các nhà phát triển trong hơn 2 thập kỷ. Hiện nay có hàng triệu ứng dụng Java đang được sử dụng. Java là một ngôn ngữ đa nền tảng, hướng đến đối tượng, lấy mạng làm trung tâm và có thể được sử dụng như một nền tảng. Đây là một ngôn ngữ lập trình nhanh, bảo mật, đáng tin cậy dùng để viết mã cho mọi thứ từ ứng dụng di động, phần mềm doanh nghiệp cho đến các ứng dụng dữ liệu lớn và công nghệ phía máy chủ.

## I.2 Các tính năng của Java



### a) Đa nền tảng

Java được thiết kế để có thể chạy trên nhiều nền tảng khác nhau, vì vậy nó rất phù hợp cho việc phát triển các ứng dụng đa nền tảng. Java sử dụng một máy ảo (JVM - Java Virtual Machine) để chạy mã nguồn, vì vậy mã nguồn được viết một lần và có thể chạy trên nhiều hệ điều hành khác nhau mà không cần thay đổi.

### b) Quản lý bộ nhớ tự động

Java có tính năng tự động quản lý bộ nhớ, tức là nó tự động thu dọn các vùng nhớ không sử dụng nữa để giảm thiểu các lỗi bộ nhớ. Điều này giúp cho các ứng dụng được viết bằng Java có thể chạy ổn định và tránh các lỗi liên quan đến bộ nhớ.

### c) Hỗ trợ đa luồng

Java có thể xử lý đa luồng, cho phép chương trình thực hiện nhiều tác vụ cùng một lúc. Điều này giúp cho các ứng dụng có thể chạy nhanh và hiệu quả hơn, đặc biệt là khi phải xử lý nhiều tác vụ cùng một lúc.



### d) Tính bảo mật cao

Java có các tính năng bảo mật như kiểm tra kiểu tĩnh và kiểm tra lỗi trên đường dẫn. Java được thiết kế để giảm thiểu các lỗ hổng bảo mật và các vấn đề liên quan đến an ninh.

### e) Hỗ trợ các thư viện và công cụ phát triển mạnh mẽ

Java có các thư viện tiêu chuẩn và các công cụ phát triển như IDE (Integrated Development Environment) để giúp các nhà phát triển tạo ra các ứng dụng dễ dàng hơn.

Các thư viện và công cụ này cung cấp các tính năng như: tạo giao diện người dùng, xử lý dữ liệu, kết nối cơ sở dữ liệu và nhiều hơn nữa,...

**f) Tính di động**

Java được sử dụng rộng rãi trong lĩnh vực di động bao gồm: các thư viện hỗ trợ việc phát triển ứng dụng di động, đóng gói ứng dụng thành các file .jar hoặc .apk, cung cấp các tính năng như xử lý đa nhiệm, kết nối mạng và tích hợp với các thiết bị phần cứng như máy ảnh.

**g) Tính độc lập với nền tảng**

Java có thể chạy trên nhiều nền tảng khác nhau và không phụ thuộc vào bất kỳ nền tảng cụ thể nào. Điều này giúp cho các ứng dụng Java có thể được triển khai trên nhiều hệ thống khác nhau mà không cần sửa đổi mã nguồn.

**h) Tính kế thừa và đa hình**

Java là ngôn ngữ lập trình hướng đối tượng (OOP), vì vậy nó có các tính năng như kế thừa, đa hình và đóng gói. Các tính năng này giúp cho mã nguồn được tái sử dụng và giảm thiểu sự trùng lặp trong mã nguồn.

**i) Tính mở rộng**

Java có tính năng mở rộng, cho phép các nhà phát triển thêm các tính năng mới vào ngôn ngữ bằng cách tạo các thư viện và API riêng. Điều này giúp cho Java có thể được sử dụng trong nhiều lĩnh vực khác nhau và được phát triển theo các hướng khác nhau.



### II.3 Ưu & nhược điểm của Java



### **Ưu điểm của Java**

- Độ tin cậy cao
- Tính đa nền tảng
- Quản lý bộ nhớ tự động
- Công cụ phát triển phong phú
- Hỗ trợ đa luồng

### **Nhược điểm của Java**

- Tốc độ chậm hơn so với các ngôn ngữ lập trình gần sát với phần cứng, chẳng hạn như C hoặc C++.
- Java có thể chạy trên nhiều nền tảng khác nhau, nhưng ứng dụng này có thể cần đến một trình biên dịch hoặc máy ảo Java riêng biệt để có thể chạy trên các thiết bị di động.
- Sử dụng bộ nhớ lớn hơn so với một số ngôn ngữ lập trình khác.
- Cú pháp phức tạp hơn so với một số ngôn ngữ lập trình khác

## **III. NGÔN NGỮ LẬP TRÌNH JAVA CÓ CÔNG DỤNG GÌ?**

### **III.1 Ngôn ngữ lập trình Java có công dụng gì?**

Vì Java là một ngôn ngữ miễn phí và linh hoạt, nó có thể được dùng để phát triển các phần mềm cục bộ và phân tán. Một số công dụng phổ biến của Java bao gồm:

#### **1. Phát triển trò chơi**

Nhiều trò chơi điện tử, trò chơi máy tính và di động nổi tiếng được phát triển bằng Java. Ngay cả những trò chơi hiện đại được tích hợp công nghệ tiên tiến như máy học hay thực tế ảo cũng được xây dựng bằng công nghệ Java.

## 2. Điện toán đám mây

Java thường được gọi là ngôn ngữ WORA (Viết một lần, chạy ở mọi nơi), khiến nó trở thành ngôn ngữ lý tưởng cho các ứng dụng phi tập trung dựa trên đám mây. Các nhà cung cấp đám mây chọn ngôn ngữ Java để chạy các chương trình trên nhiều nền tảng cơ sở khác nhau.

## 3. Dữ liệu lớn

Java được dùng cho các công cụ xử lý dữ liệu có thể làm việc với những tập dữ liệu phức tạp và số lượng dữ liệu thời gian thực không lồ.

## 4. Trí tuệ nhân tạo

Java là một trung tâm thư viện máy học đồ sộ. Sự ổn định và tốc độ của ngôn ngữ này rất lý tưởng cho việc phát triển ứng dụng trí tuệ nhân tạo như xử lý ngôn ngữ tự nhiên và học sâu.

## 5. Internet of Things

Java đã được sử dụng để lập trình các cảm biến và phần cứng trong thiết bị biên có thể kết nối một cách độc lập với Internet.

## III.2 Java được sử dụng để làm gì?

### 1. Phát triển ứng dụng máy tính

Java là một trong những ngôn ngữ lập trình phổ biến nhất để phát triển các ứng dụng máy tính đa nền tảng. Java có thể được sử dụng để phát triển các ứng dụng trên máy tính để bàn, máy chủ và các ứng dụng trên nền web. Nó cung cấp các thư viện mạnh mẽ để xử lý các nhu cầu của các ứng dụng máy tính như xử lý dữ liệu, đồ họa, đầu vào và đầu ra.

### 2. Phát triển ứng dụng di động

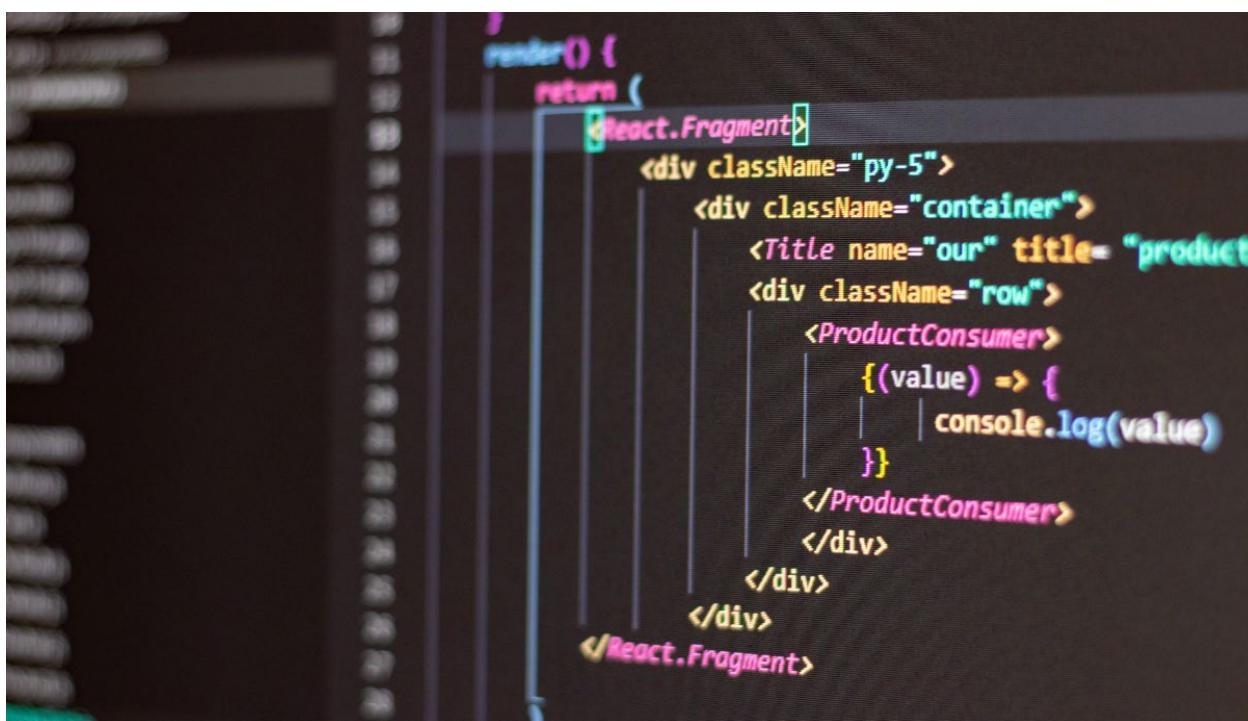
Java là một trong những ngôn ngữ lập trình phổ biến nhất để phát triển các ứng dụng di động cho hệ điều hành Android. Nó được sử dụng để phát triển các ứng dụng cho điện thoại di động và máy tính bảng. Java cung cấp một số khung như Android SDK để phát triển ứng dụng di động.

### 3. Phát triển game

Java cũng được sử dụng để phát triển các game trên nhiều nền tảng khác nhau, bao gồm các game trên máy tính, điện thoại di động và máy tính bảng. Các tính năng của Java như: tính đa luồng, xử lý đồ họa âm thanh đã được sử dụng để phát triển các game phức tạp.

### 4. Phát triển web

Java cung cấp các công cụ để phát triển các ứng dụng web động và tĩnh. Servlets, JSP (Java Server Pages), Spring Framework, Struts Framework,... là một số ví dụ về các công cụ phổ biến được sử dụng để phát triển các ứng dụng web bằng Java. Java cũng được sử dụng để phát triển các ứng dụng trên máy chủ và trình duyệt web.



### 5. Phát triển các hệ thống quản lý cơ sở dữ liệu

Java được dùng để phát triển các hệ thống quản lý cơ sở dữ liệu như Oracle và MySQL. Đồng thời, Java cũng cung cấp các API để kết nối và tương tác với các cơ sở dữ liệu, cho phép người phát triển tạo các ứng dụng quản lý cơ sở dữ liệu phức tạp và linh hoạt.

### 6. Phát triển các ứng dụng IoT

Java có thể phát triển các ứng dụng IoT (Internet of Things). Java có tính năng đa nền tảng, cho phép các ứng dụng được phát triển trên các thiết bị di động và chạy trên các thiết bị IoT khác nhau. Ngôn ngữ này cung cấp các công cụ và thư viện để phát triển các ứng dụng IoT, dùng để kết nối các thiết bị và xử lý dữ liệu.

## 7. Phát triển các ứng dụng trí tuệ nhân tạo

Java cũng được sử dụng để phát triển các ứng dụng trí tuệ nhân tạo (AI). Các tính năng của Java như xử lý đa luồng và tính năng hướng đối tượng cho phép người phát triển tạo các ứng dụng AI phức tạp.

## 8. Phát triển các ứng dụng blockchain

Java cũng có thể được sử dụng để phát triển các ứng dụng blockchain. Các tính năng của Java như độ bảo mật cao, độ tin cậy và khả năng xử lý các giao dịch trên mạng đã được sử dụng để phát triển các ứng dụng blockchain phức tạp.

### III.3 Một số lưu ý về cú pháp Java

- Các dấu ngoặc {} đánh dấu điểm khởi đầu và kết thúc của code
- System là lớp Java có sẵn bao gồm đối tượng out (viết tắt cho output). System, out và println() đều là những thành phần cần thiết để bạn có thể xuất các giá trị lên màn hình
- Mỗi dòng code cần được kết thúc bằng dấu chấm phẩy ;
- Mọi dòng code trong Java đều cần được đặt trong một lớp (class). Mỗi class đều cần được bắt đầu với chữ viết hoa, vì Java phân biệt chữ viết hoa và viết thường. Ví dụ, “MyFirstJavaProgram” khác với “myfirstjavaprogram”
- Cần nhớ tập tin Java và lớp Java cần có cùng một cái tên. Bạn có thể lưu tên file bằng tên class với đuôi “.java”

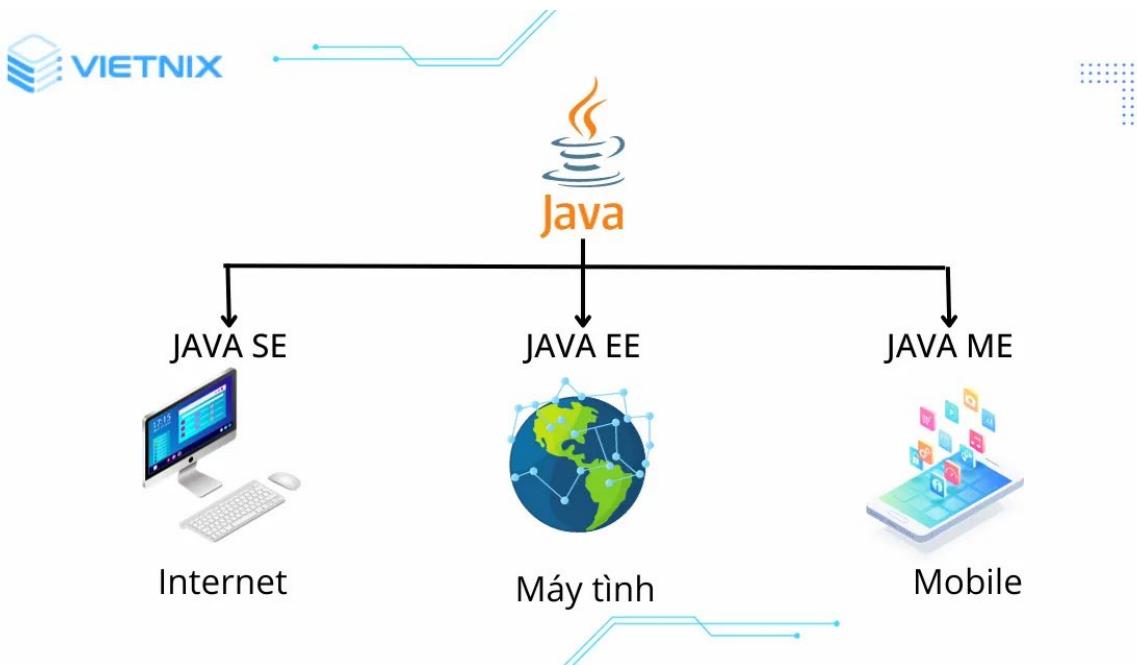
## IV. Các công cụ Java phổ biến

Khi phát triển các ứng dụng Java, những công cụ phổ biến nhất có thể hỗ trợ bạn gồm có:

- Phiên bản Java phổ biến nhất: Java 8 và Java 11. Ngoài ra, Java 21 (phát hành vào tháng 9 năm 2021) chính là phiên bản Java LTS mới nhất. LTS viết tắt cho long term support – hỗ trợ dài hạn. Java LTS chỉ các phiên bản Java được cập nhật thường xuyên và là phiên bản tiêu chuẩn của ngành trong một thời gian dài vì tính ổn định của nó. Một số phiên bản Java LTS là Java 11, Java 17, và mới nhất là Java 21.
- IDE: Eclipse, IntelliJ IDEA, Netbeans,...
- Framework: AngularJS, Spring MVC, Java Server Faces,...

- Server: SpringBoot, Tomcat và Wildfly,...
- Framework kiểm thử: JUnit, Mockito, Selenium,...

## V. CÁC THUẬT NGỮ CƠ BẢN CẦN BIẾT VỀ JAVA



Để có thể hiểu biết một cách đầy đủ về Java, chúng ta cần hiểu được ba thuật ngữ cơ bản gắn liền với ngôn ngữ lập trình này, bao gồm **Java virtual machine (JVM)**, **Java Runtime Environment (JRE)**, và **Java Development Kit (JDK)**.



## Java Development Kit (JDK)

Java development kit (JDK) là một bộ công cụ đa nền tảng cung cấp các công cụ và thư viện (libraries) cần thiết để phát triển và triển khai các ứng dụng và chương trình Java. JDK là một gói (package) hoặc kit (bộ công cụ) quan trọng của Java. JDK bao gồm hai thành phần quan trọng:

- Công cụ lập trình để phát triển các chương trình Java: trình diễn giải (compiler) Javac, trình biên dịch (interpreter) Java, trình lưu trữ (archiver) jar và nhiều công cụ khác
- Java Runtime Environment, tức môi trường để triển khai và test chương trình

## Java Runtime Environment (JRE)

Những người mới bắt đầu làm quen với Java thường nhầm lẫn Java Runtime Environment (JRE) với JDK. Có một sự thật là nếu người dùng chỉ quan tâm đến việc chạy chương trình Java trên máy thì kiến thức về JRE là gì là đủ. Tuy nhiên, nếu người dùng muốn viết các phần mềm sử dụng Java thì cần hiểu được cả JRE và JDK.

Tóm lại, trong khi JDK là một bộ công cụ được dùng để phát triển ứng dụng/ chương trình thì JRE chỉ cung cấp môi trường để chạy chúng trên máy.

JRE bao gồm:

- Java Virtual Machine (JVM)
- Các thư viện class có trong môi trường sân xuất
- Các thư viện bổ sung cho developer, ví dụ: thư viện quốc tế hóa và thư viện IDL

### **Java Virtual Machine (JVM)**

Java Virtual Machine là một chương trình có sẵn trong JDK và JRE. JVM thực tế là một trình dịch (interpreter) giúp người dùng chạy ứng dụng Java trên các nền tảng ứng dụng khác nhau.

Nguyên nhân đằng sau tên gọi này là do JVM là một máy ảo cho phép chạy các chương trình Java ngay cả trên các phần cứng hay hệ điều hành khác nhau mà không cần biên dịch sang code bytecode của Java. Đây cũng chính là đặc điểm tạo nên phương châm WORA của Java – Code một lần, chạy ở bất cứ đâu (Write Once, Run Anytime).

## **VI. SỰ KHÁC BIỆT GIỮA C++ VÀ JAVA LÀ GÌ?**



Java là ngôn ngữ lập trình hướng tới đối tượng dựa trên ngôn ngữ C/ C++, tuy nhiên Java và C++ vẫn có điểm khác nhau. Vậy **sự khác biệt giữa C++ và Java là gì?**

- Java không nạp chồng toán tử như C ++.
- C++ cung cấp các lớp mẫu còn Java thì không.
- C++ hỗ trợ con trỏ rõ ràng, nhưng Java lại không.
- Java không hỗ trợ các biến toàn cục như C ++.
- Java sử dụng hàm finalize () còn C ++ sử dụng hàm hủy.
- Java không có tệp tiêu đề như C ++.
- Java hỗ trợ tương tự thông qua các giao diện còn C ++ hỗ trợ kế thừa thông qua các lớp.
- Java không có hỗ trợ các cấu trúc và liên hiệp như C ++.
- Java không hỗ trợ từ khóa “ảo” như C++.

## Chương 1

---

### JAVA CƠ BẢN

---

*“Cho dù bạn là người mới bắt đầu, nhà phát triển, lập trình viên hay người viết blog – kiến thức là điểm bắt đầu cho việc thực hành thực tế. Vì thế cần nắm vững lý thuyết trước khi vào thực hành”*

Trong phần thực hành này, chúng ta sẽ tìm hiểu:

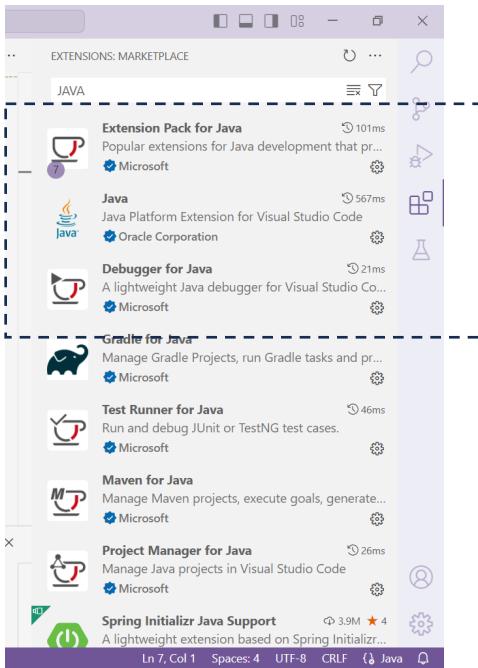
- *Cài đặt Java trong Visual Studio Code*
- *Chuỗi ký tự*
- *System.out.println*
- *print và println*
- *Ứng dụng hiển thị chữ*
- *Vẽ hình bằng cú pháp println*

## I. CÀI ĐẶT JAVA TRONG MÔI TRƯỜNG VISUAL STUDIO CODE

### I.1 Cài đặt Java trong Visual Studio Code

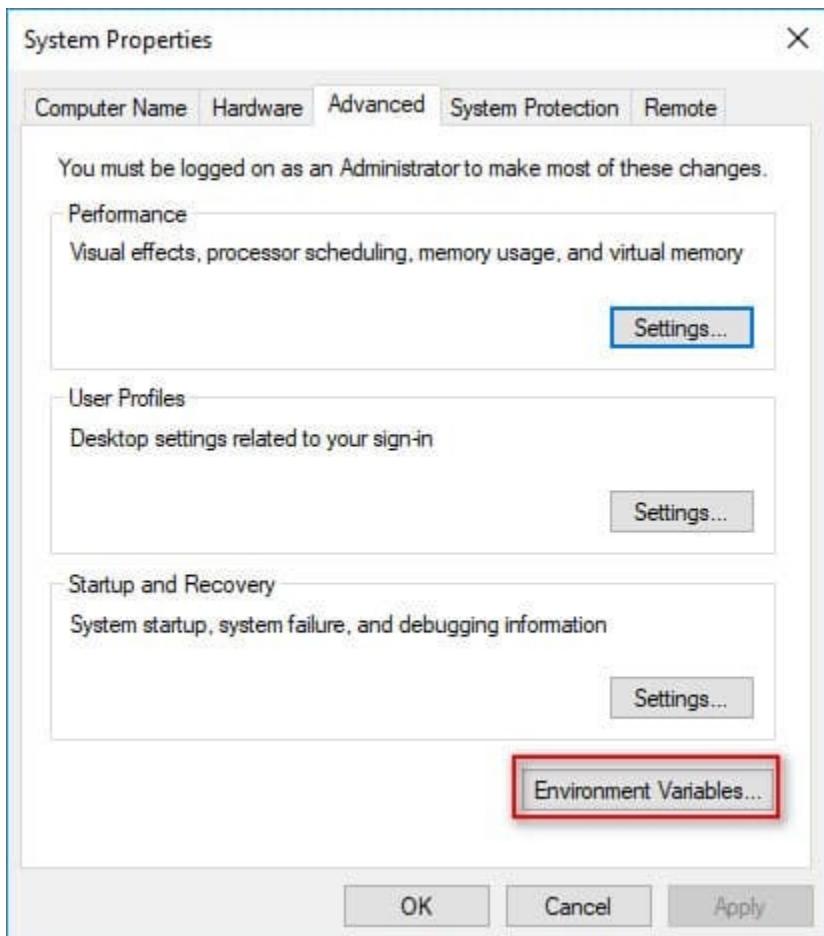
Bước 1: Trong visual Studio Code bạn tiến hành Install

- a) Java
- b) Java JDK SE



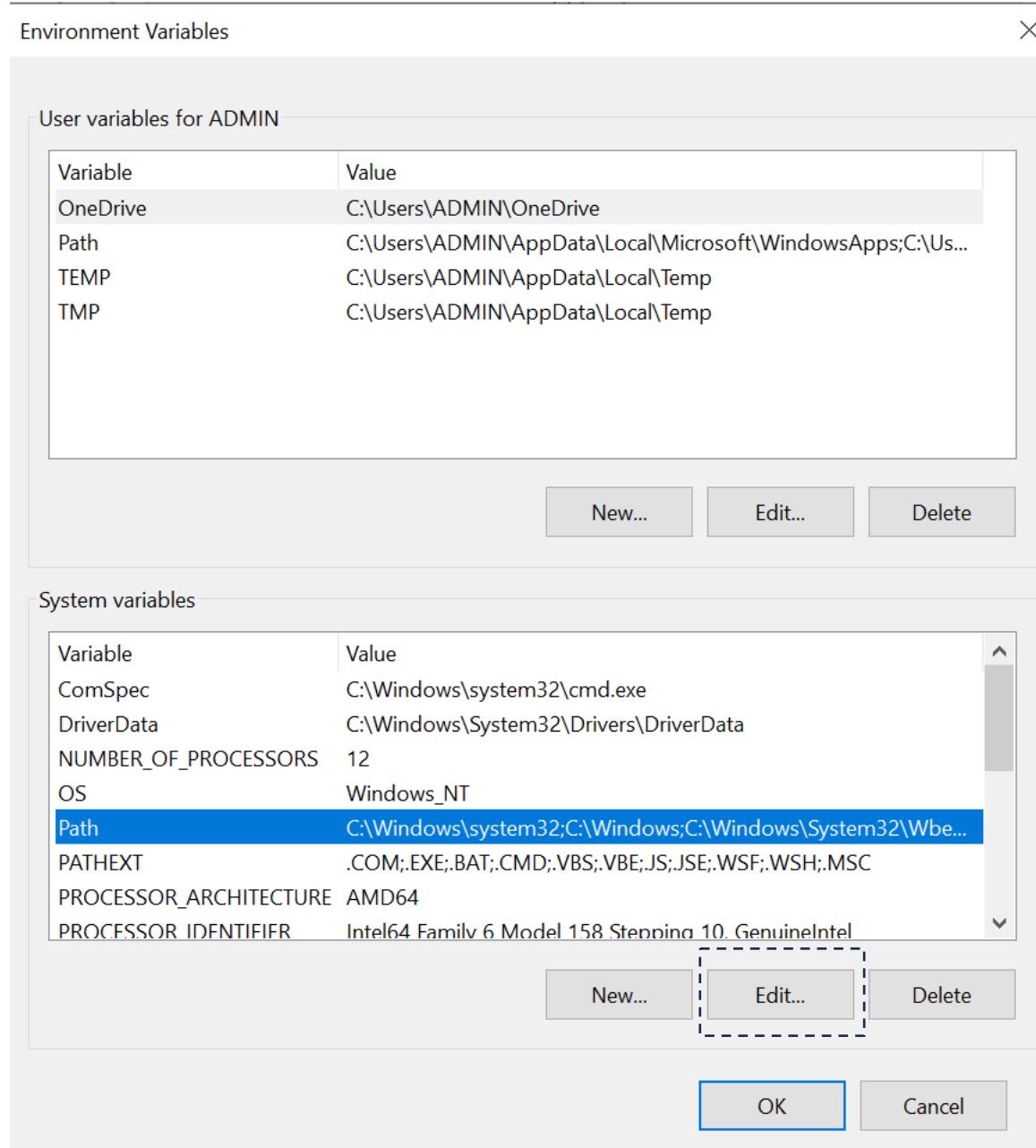
### 2) Cài đặt biến môi trường (Environment Variables) trong Java: PATH và CLASSPATH

- Bước 1: Nhấp chuột phải vào **My Computer** > **Chọn Properties**
- Bước 2: Click vào **Advanced system settings**
- Bước 3: Click vào **Environment Variables**

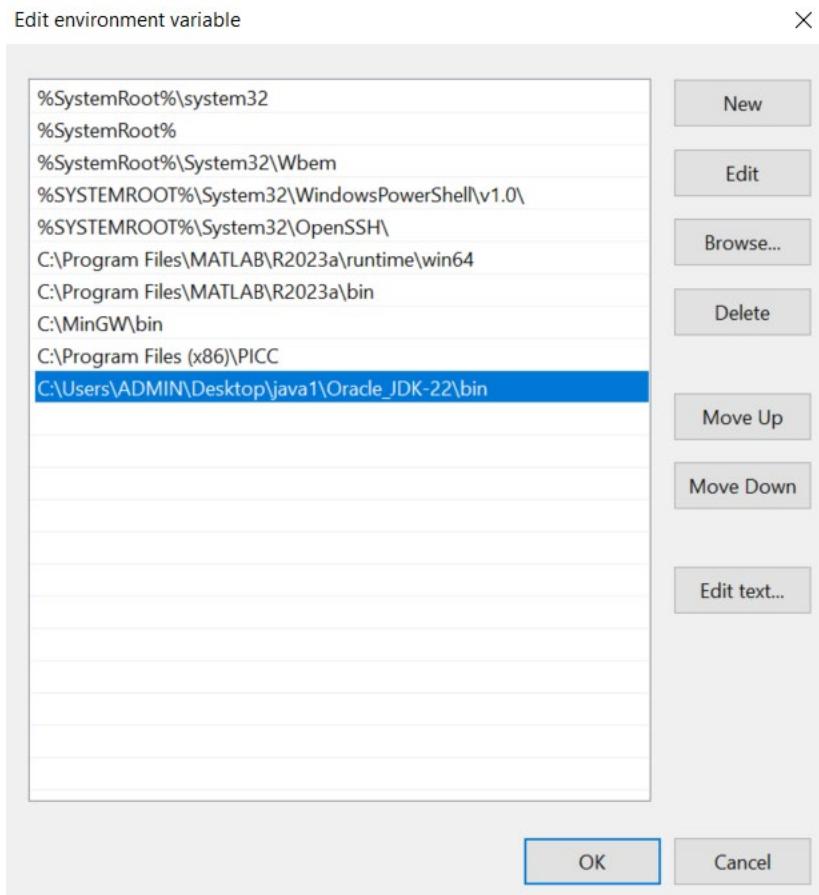


- Bước 4: Tại Khung làm việc System variables > Path

## Thực hành: Cấu trúc Dữ liệu và Giải thuật



- Bước 5: Trong cửa sổ Edit environment variable chọn New
- Bước 6: Sao chép đường dẫn của thư mục bin được cài đặt trong thư mục JDK
- Bước 7: Dán đường dẫn thư mục bin vào mục Variable value và nhấp vào nút OK



## II. DẠNG CƠ BẢN CỦA LỚP JAVA

Dạng cơ bản của lớp Java như sau:

```
public class <name> {  
    <method>  
    <method>  
    ...  
    <method>  
}
```

Dòng đầu tiên của lớp được gọi là tiêu đề lớp class header. Từ public trong tiêu đề cho biết lớp này có sẵn cho bất kỳ ai sử dụng. Lưu ý rằng mã chương trình trong một lớp được bao trong các ký tự dấu ngoặc nhọn ( { } ) .

Tối thiểu, một chương trình hoàn chỉnh yêu cầu một hàm đặc biệt được gọi là phương pháp chính main. Nó có cú pháp sau:

```
public static void main(String[] args) {  
    <statement>;  
    <statement>;  
    ...  
    <statement>;  
}
```

### System.out.println

Một trong những câu lệnh đơn giản và phổ biến nhất là System.out.println, được sử dụng để tạo ra một dòng đầu ra. Đây là một "câu thần chú" khác mà bạn nên ghi nhớ. Tính đến thời điểm viết bài này, Google liệt kê khoảng 8.000.000 trang web có đề cập đến System.out.println. Điều quan trọng cần nhớ về câu lệnh này là nó được sử dụng để tạo ra một dòng đầu ra được gửi đến cửa sổ bảng điều khiển.

Các ví dụ:

```
System.out.println();  
System.out.println("Hello, world!");  
System.out.println("This program produces four");  
System.out.println("lines of output.");
```

### Bài thực hành số 1: “Hello World!”



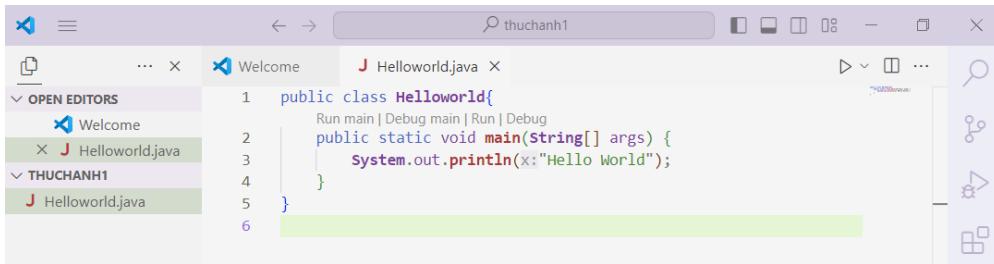
Bước 1: Chúng ta tạo folder “Thuchanh1”

Bước 2: Chạy Visual Studio Code, Open folder “Thuchanh1”, tạo New file có tên là thuchanh1.java

Bước 3: Đánh nội dung sau:

```
public class HelloWorld{  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

Bước 4: Khi đó trên vùng làm việc Visual Studio Code xuất hiện như sau:

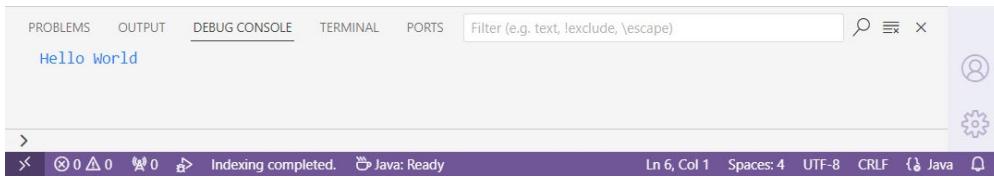


A screenshot of the Visual Studio Code interface. The title bar says "thuchanh1". The left sidebar shows "OPEN EDITORS" with "Welcome" and "Helloworld.java" listed under "THUCHANH1". The main editor area contains the following Java code:

```
public class Helloworld{
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

Chúng ta bấm vào Run main.

Bước 5: Kết quả nhận được như sau:



A screenshot of the Visual Studio Code terminal tab. It shows the output of the Java program "Hello World". The terminal interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The DEBUG CONSOLE tab is active. The output window displays:

```
Hello World
```

Indexing completed. Java: Ready

### III. CHUỖI KÝ TỰ ĐẶC BIỆT PHỐ BIỀN

Ký tự	Ý nghĩa
\t	Nhảy (tab) văn bản
\n	Sang dòng mới
\”	Dấu ngoặc kép
\\\	Ký tự dấu gạch chéo ngược

### Bài thực hành số 2

Tiếp tục bài thực hành số 1:



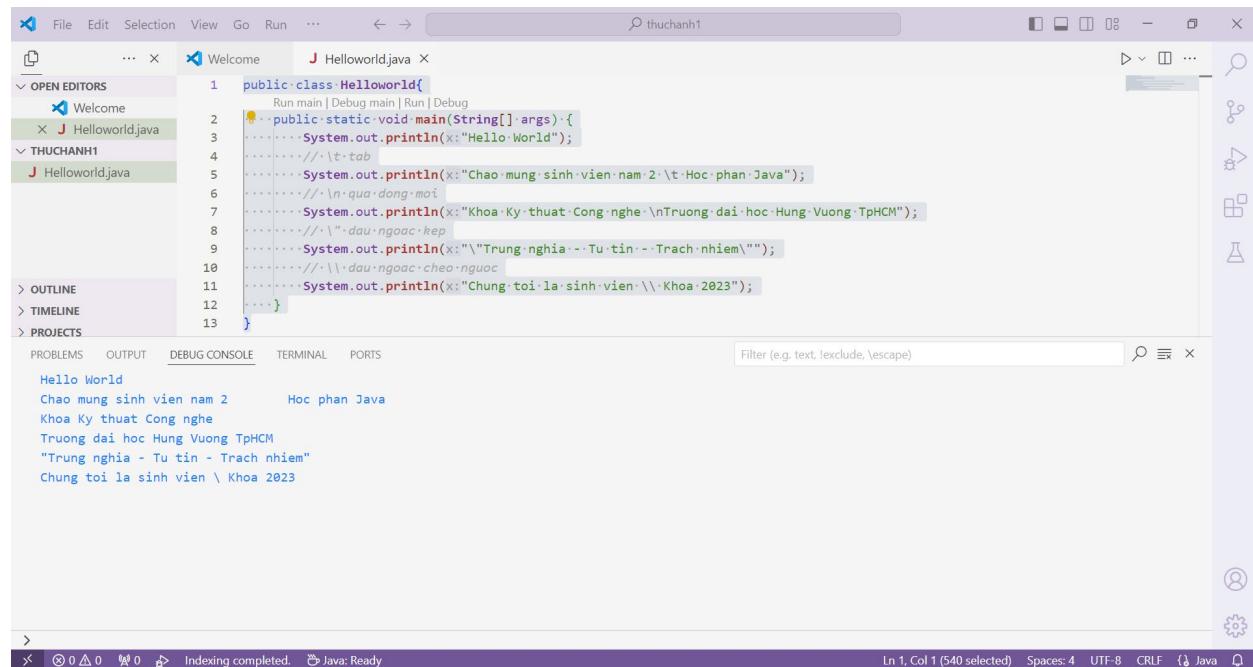
Bước 6:

Thực hiện các lệnh sau:

```
public class Helloworld{
    public static void main(String[] args) {
        System.out.println("Hello World");
        // \t tab
        System.out.println("Chao mung sinh vien nam 2 \t Hoc phan Java");
        // \n qua dong moi
```

```
System.out.println("Khoa Ky thuat Cong nghe \nTruong dai hoc Hung Vuong  
TpHCM");  
    // \" dau ngoac kep  
    System.out.println("\\"Trung nghia - Tu tin - Trach nhanh\"");  
    // \\ dau ngoac cheo nguoc  
    System.out.println("Chung toi la sinh vien \\ Khoa 2023");  
}  
}
```

Bước 7: Kết quả thu được như sau:



The screenshot shows the Eclipse IDE interface. In the center, there is a code editor window titled "Helloworld.java" with the following content:

```
public class Helloworld{  
    public static void main(String[] args){  
        System.out.println("Hello World");  
        System.out.println("Chao mung sinh vien nam 2 \t Hoc phan Java");  
        System.out.println("//\nqua dong moi");  
        System.out.println("\\"Khoa Ky thuat Cong nghe \nTruong dai hoc Hung Vuong TpHCM\"");  
        //\" dau ngoac kep  
        System.out.println("\\"Trung nghia - Tu tin - Trach nhanh\"");  
        //\\ dau ngoac cheo nguoc  
        System.out.println("\\"Chung toi la sinh vien \\ Khoa 2023\"");  
    }  
}
```

Below the code editor, the "OUTPUT" tab is selected, displaying the following text:

```
Hello World  
Chao mung sinh vien nam 2      Hoc phan Java  
Khoa Ky thuat Cong nghe  
Truong dai hoc Hung Vuong TpHCM  
"Trung nghia - Tu tin - Trach nhanh"  
Chung toi la sinh vien \ Khoa 2023
```

## IV. PRINT VÀ PRINTLN

**Println:** Nó hiện thị dữ liệu tại dòng hiện tại và sau đó nó sẽ di chuyển sang dòng mới.

**Print:** Nó hiện thị dữ liệu tại dòng hiện tại.

### Bài thực hành số 3:

Tiếp tục bài thực hành số 2.

```
// dung Len print  
System.out.print("Chung toi luon tu hao");  
System.out.print("la nhung sinh vien cham chi nhat \n");
```



```
System.out.println("");

// dung len println
System.out.println("Chung toi luon tu hao");
System.out.println("la nhung sinh vien cham chi nhat");
```

### Bước 8: Kết quả thu được

The screenshot shows the Eclipse IDE interface. On the left, there's a sidebar with 'OUTLINE', 'TIMELINE', 'PROJECTS', 'RUN CONFIGURATION', and 'JAVA PROJECTS'. The main area contains Java code:

```
12
13
14 // dung Len print
15 System.out.print(s:"Chung toi luon tu hao");
16 System.out.print(s:"la nhung sinh vien cham chi nhat \n");
17
18 System.out.println(x:"");
19 // dung Len println
20 System.out.println(x:"Chung toi luon tu hao");
21 System.out.println(x:"la nhung sinh vien cham chi nhat");
22
23 }
```

Below the code, the 'OUTPUT' tab is selected, showing the console output:

```
Chung toi luon tu hao la nhung sinh vien cham chi nhat
Chung toi luon tu hao
la nhung sinh vien cham chi nhat
```

At the bottom, status bar messages include 'Indexing completed.', 'Java: Ready', 'In 25, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', and icons for Java and Q.

## IV.2 Các từ khoá thông dụng trong Java

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

### Bài thực hành số 4: Sử dụng lệnh `print` hay `println` vẽ hình

Đề bài: Vẽ hình sau



```
^
/ \
\ /
\ \
\ \
```

```
\ 
\ /
\ \
\ \
\ \
```

```
^
/ \
/ \
+---+
|   |
|   |
+---+
| Viet|
| Nam |
+---+
|   |
|   |
+---+
^
/ \
/ \
```

Bước 1: Tạo file “vehinh1.java”

Bước 2: Tạo hình đầu tiên

```
public class Vehinh1 {
    public static void main(String[] args) {
        System.out.println(" ^");
        System.out.println(" / \\");
        System.out.println("/   \\");
        System.out.println("\\\\   /");
        System.out.println(" \\ /");
```

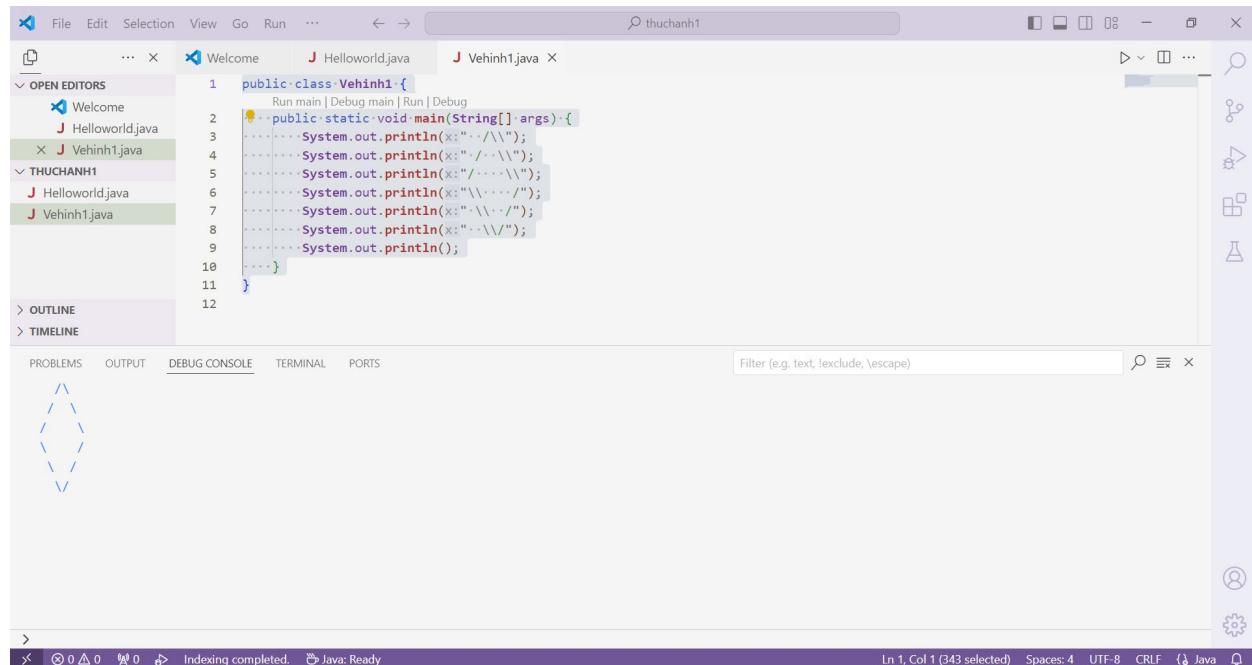
```
System.out.println(" \\");

System.out.println();

}

}
```

### Bước 3: Kết quả thu được

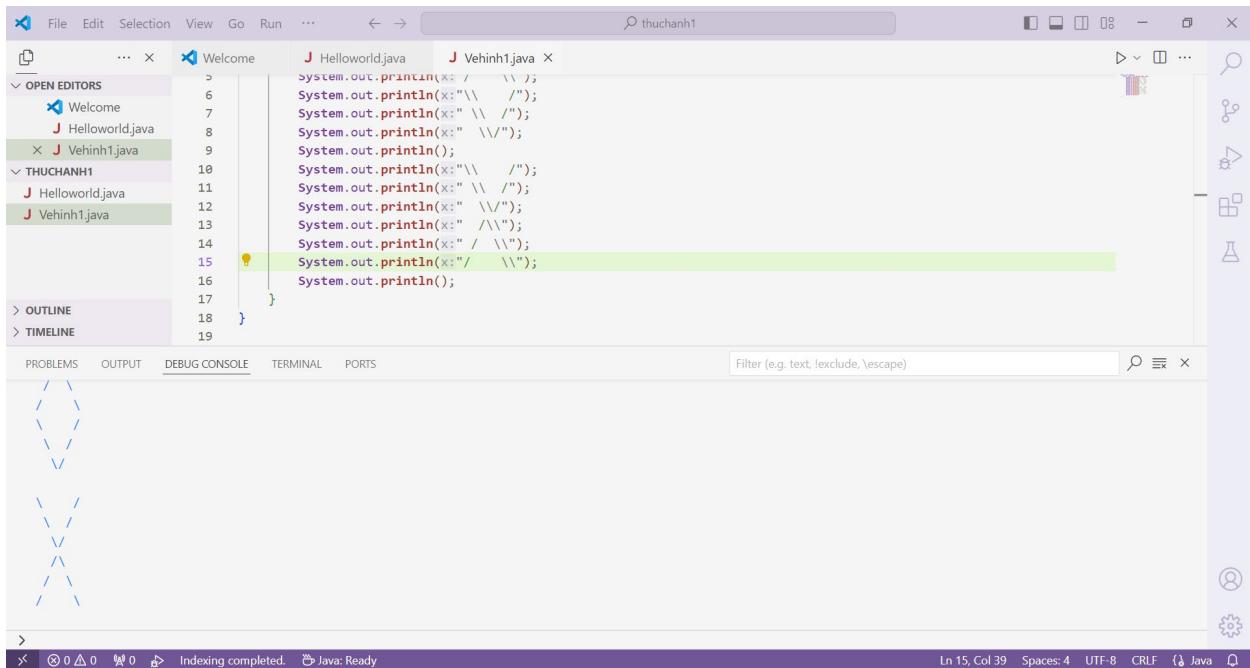


Lưu ý: Ký tự dấu gạch chéo ngược ta thêm sử dụng //

Bước 4: Tương tự hình số 2

```
System.out.println("\\   /");
System.out.println(" \\  /");
System.out.println("  \\\/");
System.out.println(" /  \\\/");
System.out.println(" /      \\\/");
System.out.println();
```

## Thực hành: Cấu trúc Dữ liệu và Giải thuật



The screenshot shows the Eclipse IDE interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Title Bar:** thuchanh1
- Left Sidebar:** OPEN EDITORS (Welcome, Helloworld.java, Vehinh1.java), THUCHANH1 (Helloworld.java, Vehinh1.java).
- Central Area:** Two code editors are open:
  - J Helloworld.java:** Prints various escape sequences like \n, \t, \\, \\", \\\\", etc.
  - J Vehinh1.java:** Prints the same escape sequences as Helloworld.java.
- Bottom Status Bar:** Indexing completed. Java: Ready, Line 15, Column 39, Spaces: 4, UTF-8, CRLF, Java.

Tương tự sinh viên cạnh chính ta sẽ có:

## Thực hành: Cấu trúc Dữ liệu và Giải thuật

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with the following details:

- File Explorer (Left):** Shows open files: `Welcome`, `Helloworld.java`, `Vehinh1.java`, and a folder `THUCHANH1` containing `Helloworld.java` and `Vehinh1.java`. The `Vehinh1.java` file is currently selected.
- Code Editor (Center):** Displays the `Vehinh1.java` code. The code prints a diamond shape and a welcome message ("Hello World!"). A yellow warning icon is present on line 20. The code is as follows:

```
1 public class Vehinh1 {  
2     Run main | Debug main | Run | Debug  
3     public static void main(String[] args) {  
4         System.out.println("//");  
5         System.out.println("// ");  
6         System.out.println("//\\ ");  
7         System.out.println("\\// ");  
8         System.out.println("\\//\\ ");  
9         System.out.println("\\\\");  
10        System.out.println("\\\\//");  
11        System.out.println("\\\\// ");  
12        System.out.println("\\\\//\\ ");  
13        System.out.println("\\\\//\\//");  
14        System.out.println("\\\\//\\// ");  
15        System.out.println("\\\\//\\//\\ ");  
16        System.out.println();  
17        System.out.println("\\\\//\\//");  
18        System.out.println("\\\\//\\// ");  
19        System.out.println("\\\\//\\//\\ ");  
20        System.out.println("-----+");  
21        System.out.println("|   |");  
22        System.out.println("|   |");  
23        System.out.println("-----+");  
24        System.out.println("|| Viet |");  
25        System.out.println("|| Nam |");  
26        System.out.println("-----+");  
27        System.out.println("||   |");  
28        System.out.println("||   |");  
29        System.out.println("-----+");  
30        System.out.println("\\\\//");  
31        System.out.println("\\\\// ");  
32        System.out.println("\\\\//\\ ");  
33    }  
34}  
35
```

- Bottom Status Bar:** Shows indexing status: "Indexing completed.", Java: Ready, and file statistics: Ln 20, Col 35, Spaces: 4, CRLF, Java.

## V. CHÚ THÍCH

Chú thích dùng để diễn giải hoặc ghi chú cho code. Trình biên dịch sẽ bỏ qua nó khi thực thi.

Có 2 cách để tạo ghi chú

1. Dùng /\* và \*/ chú thích cho đoạn dài
2. Dùng // chú thích cho 1 dòng

## VI. HÀM TĨNH (STATIC METHOD)

Cấu trúc của hàm tĩnh như sau:

```
public static void <name>() {  
    <statement>;  
    <statement>;  
    ...  
    <statement>;  
}
```

Gọi hàm là một lệnh để thực thi một phương thức khác, khiến tất cả các câu lệnh bên trong phương thức đó được thực thi.

### Bài thực hành số 5:

Chúng ta muốn tạo ra 2 hình vuông như sau:



```
+---+  
|   |  
|   |  
+---+  
  
+---+  
|   |  
|   |  
+---+
```

Các làm thông thường như sau:

1. Tạo file “vehinh2.java”
2. Viết code như sau

```
public class vehinh2 {  
    public static void main(String[] args) {  
        System.out.println("-----+");  
        System.out.println("|      |");  
        System.out.println("|      |");  
        System.out.println("-----+");  
        System.out.println();  
        System.out.println("-----+");  
        System.out.println("|      |");  
        System.out.println("|      |");  
        System.out.println("-----+");  
    }  
}
```

Kết quả thu được:

The screenshot shows the Visual Studio Code interface. The code editor displays the Java code for `vehinh2.java`. The terminal below shows the output of the program, which is a 2x2 square made of '+' and vertical bar characters.

```
+----+  
|    |  
|    |  
+----+  
  
+----+  
|    |  
|    |  
+----+
```

## Bài thực hành số 6: Phát triển khi sử dụng hàm tĩnh



Nhận xét: Hình trong bài tập số 5 là 1 hình vuông nhưng được lặp lại 2 lần. Vì thế chúng ta sử dụng hàm tĩnh để giản lược code.

1. Tạo file “vehinh3.java”
2. Tạo hàm tĩnh như sau:

```
public static void drawBox() {  
    System.out.println("-----+");  
    System.out.println(" | | ");  
    System.out.println(" | | ");  
    System.out.println("-----+");  
}
```

3. Trong main chúng ta chỉ cần gọi hàm ra như sau:

```
public class vehinh3 {  
    public static void main(String[] args) {  
        drawBox();  
        System.out.println();  
        drawBox();  
    }  
}
```

4. Kết quả thu được

The screenshot shows the Eclipse IDE interface with the following details:

- Open Editors:** Shows files Helloworld.java, Vehinh1.java, vehinh2.java, and vehinh3.java. The file vehinh3.java is currently selected.
- Outline:** Shows the class structure and method definitions.
- Terminal:** Displays the output of the program, which consists of two identical boxes separated by a blank line. Each box is a square made of four lines: a top horizontal line ("-----+"), a left vertical line (" | | "), a right vertical line (" | | "), and a bottom horizontal line ("-----+").

```
+----+  
| |  
| |  
+----+  
  
+----+  
| |  
| |  
+----+
```

Quá trình thực hiện code trên như sau:

```
public static void main(String[] args) {  
    drawBox();  
  
    public static void drawBox() {  
        System.out.println("-----");  
        System.out.println("|      |");  
        System.out.println("|      |");  
        System.out.println("-----");  
    }  
  
    System.out.println();  
    drawBox();  
  
    public static void drawBox() {  
        System.out.println("-----");  
        System.out.println("|      |");  
        System.out.println("|      |");  
        System.out.println("-----");  
    }  
}
```

Lưu ý: Việc khai báo hàm con trong JAVA cho phép bạn khai báo bất kì đâu bạn muốn mà không cần quan tâm đến thứ tự.

## VII. HÀM GỌI HÀM

Chúng ta xem xét một ví dụ sau:

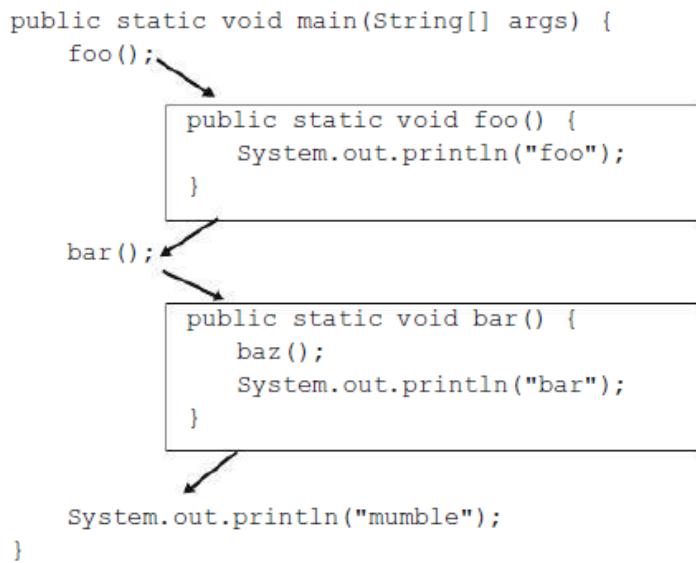
Để hiện thị các chữ (“foo,” “bar,” “baz,” and “mumble”) ta sử dụng hàm tĩnh

```
foo  
baz  
bar  
mumble
```

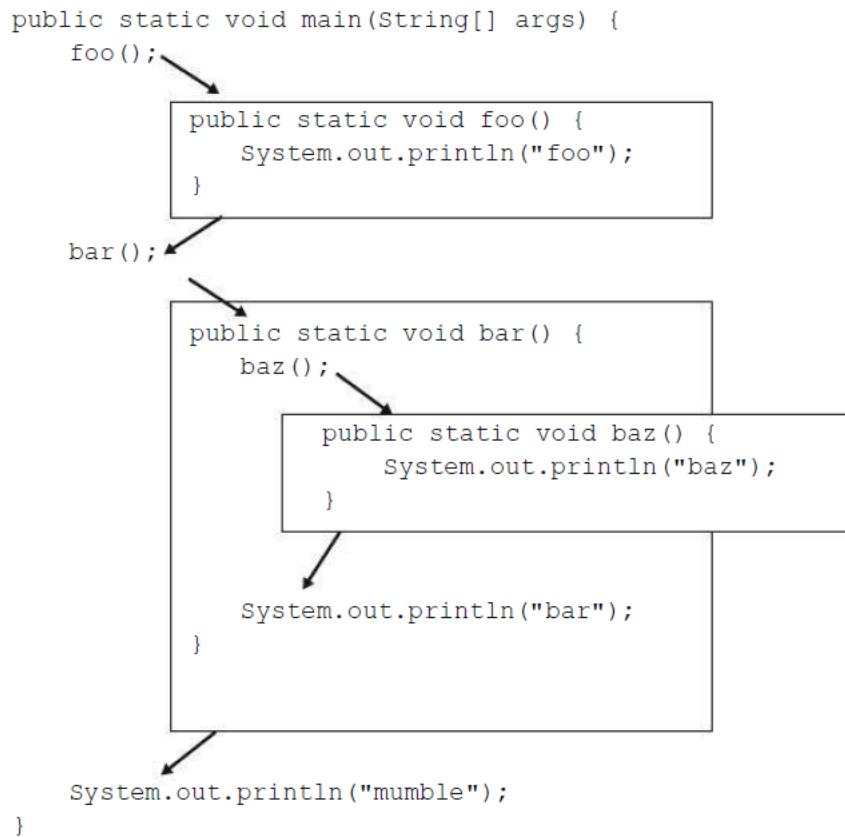
Code như sau:

```
public class FooBarBazMumble {  
    public static void main(String[] args) {  
        foo();  
        bar();  
        System.out.println("mumble");  
    }  
  
    public static void foo() {  
        System.out.println("foo");  
    }  
  
    public static void bar() {  
        baz();  
        System.out.println("bar");  
    }  
  
    public static void baz() {  
        System.out.println("baz");  
    }  
}
```

Quá trình chạy chương trình như sau:



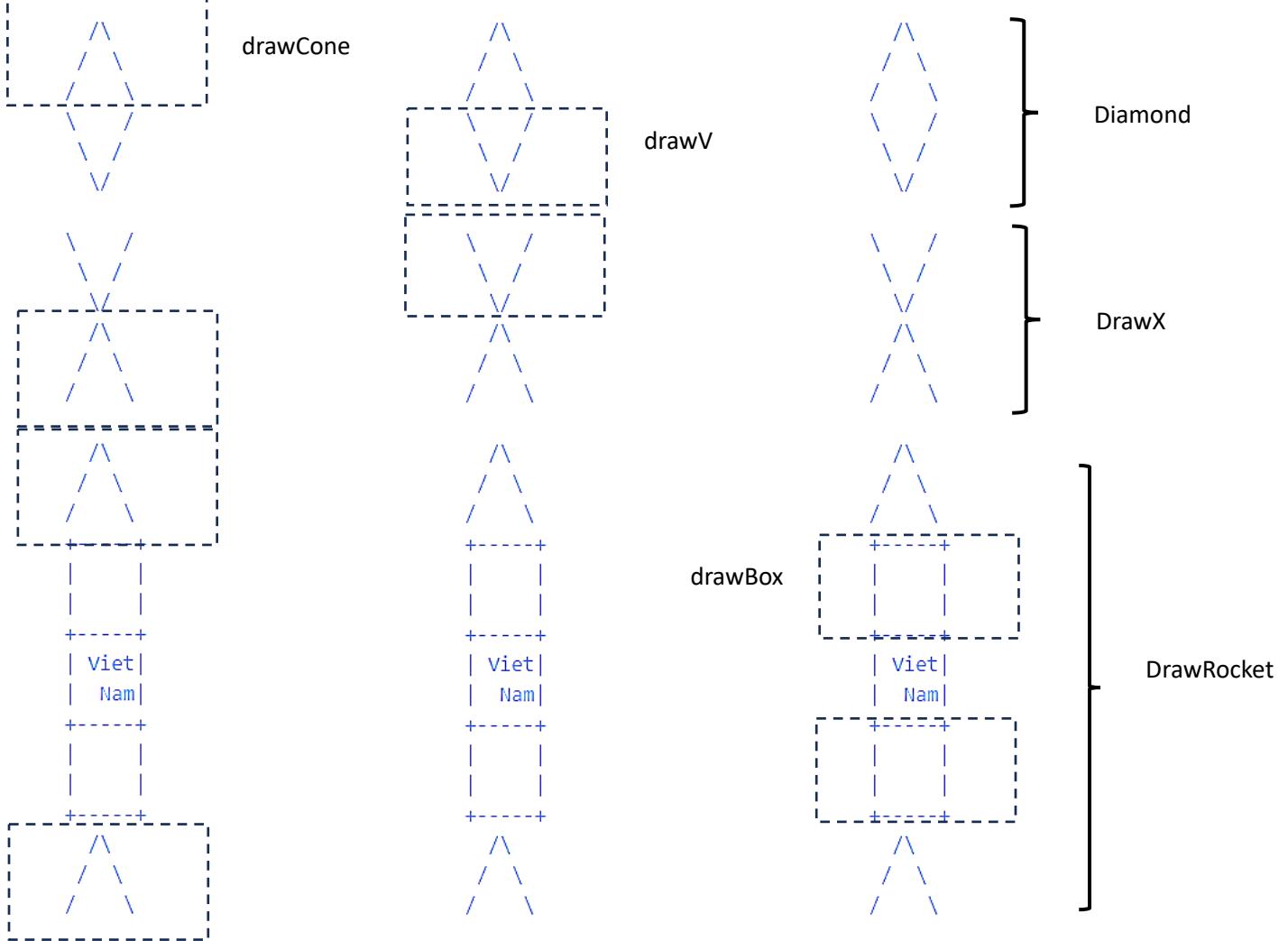
Trong đó hàm bar gọi hàm bar tiếp



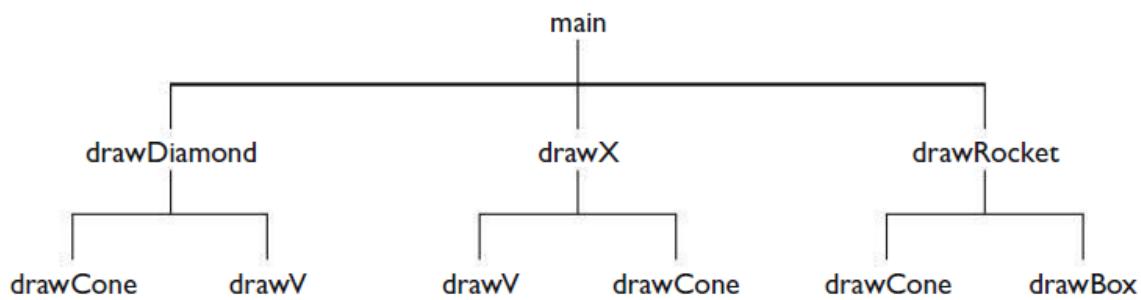


## Bài thực hành số 7: Vẽ hình dùng hàm tĩnh

Đầu tiên ta phân tích hình trên thành các hình giống nhau



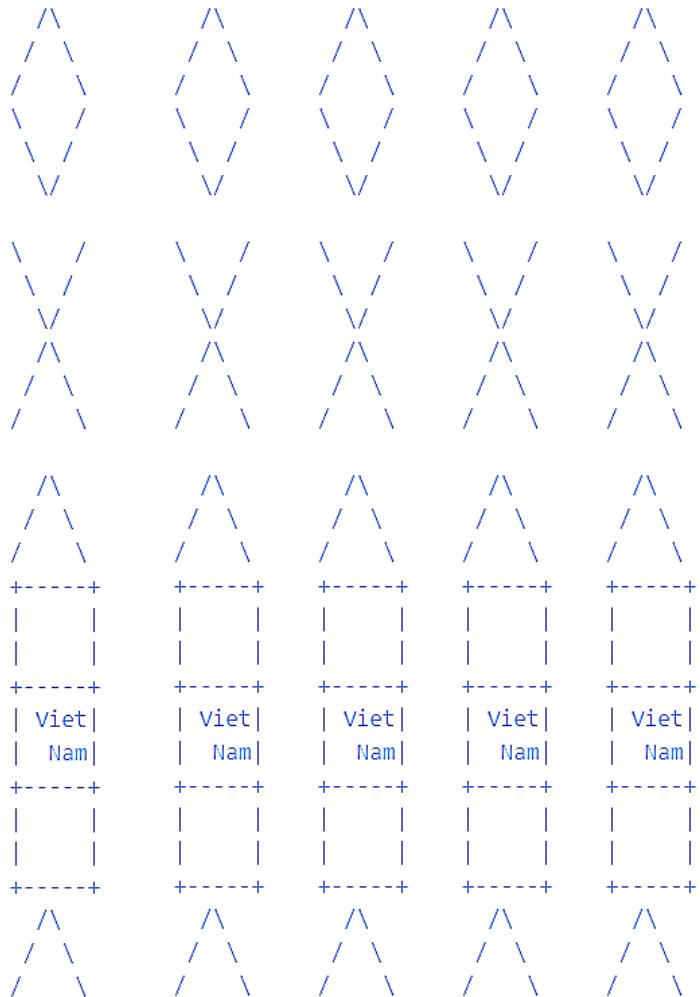
Vậy ta phân tích hình trên thông qua các hình vẽ sau



Phần này sinh viên tự viết code và thực hành.

Bài tập thực hành tại nhà:

Bài tập 1: Sinh viết code thể hiện 5 hình tên lửa sau



Bài tập 2. Sinh viên vẽ hình sau

\* \* \* \* \*

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

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

\* \* \* \* \*

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

\* | | | | | \*

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

\* \* \* \* \*

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

\* \* \* \* \*

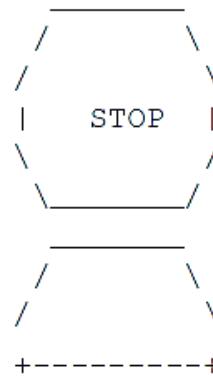
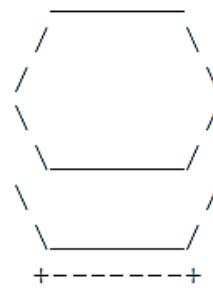
\* | | | | | \*

\* | | | | | \*

\* \* \* \* \*

\* \* \* \* \*

### Bài tập 3: Sinh viên vẽ hình sau:



## Chương 2

### DỮ LIỆU CƠ BẢN VÀ VÒNG LẶP

*“Cho dù bạn là người mới bắt đầu, nhà phát triển, lập trình viên hay người viết blog – kiến thức là điểm bắt đầu cho việc thực hành thực tế. Vì thế cần nắm vững lý thuyết trước khi vào thực hành”*

Trong phần thực hành này, chúng ta sẽ tìm hiểu:

- Định nghĩa các giới hạn cơ bản
- Biến trong Java
- Loop
- For
- Ứng dụng cơ bản trong vẽ hình

## I. CÁC KIỂU DỮ LIỆU

### I.1 Kiểu dữ liệu cơ bản

Kiểu	Ý nghĩa	Ví dụ
<b>int</b>	Số nguyên	42, -3, 18, 20493, 0
<b>double</b>	Số thực	7.35, 14.9, -19.83423
<b>char</b>	Kí tự đơn	'a', 'X', '!'
<b>boolean</b>	Luật lý	True, False

### I.2 Biểu thức

Biểu thức là quá trình thu được giá trị của một biểu thức.

Các phép toán trong Java

Kiểu	Ý nghĩa	Ví dụ	Kết quả
<b>+</b>	Cộng	2+2	4
<b>-</b>	Trừ	53-18	35
<b>*</b>	Nhân	3*8	24
<b>/</b>	Chia	4.8/2.0	2.4
<b>%</b>	Lấy phần dư (mod)	19%5	4

### Phép toán ưu tiên

Miêu tả	Kiểu
Phép đơn lẻ	$+, -, *, /, \%$
Đa nhiệm	$*, /, \%$
Phép cộng	$+, -$

Ví dụ:

$$\begin{array}{ccccccccccccc}
 13 & * & 2 & + & 239 & / & 10 & \% & 5 & - & 2 & * & 2 \\
 \underbrace{13 \quad 2}_{26} & + & \underbrace{239 \quad 10}_{23} & \% & 5 & - & 2 & * & 2 \\
 26 & + & \underbrace{23}_{3} & \% & 5 & - & 2 & * & 2 \\
 26 & + & 3 & - & 2 & * & 2 \\
 26 & + & 3 & - & \underbrace{2}_{4} & * & 2 \\
 \\[10pt]
 26 & + & 3 & - & 4 \\
 \underbrace{26 \quad 3}_{29} & - & \underbrace{4}_{25}
 \end{array}$$

## II. BIẾN

II.1 Khai báo biến: Yêu cầu đặt riêng một biến mới có tên và kiểu dữ liệu đã cho.

`<type> <name>;`

Ví dụ:

`double height;`

`double weight;`

`double bmi;`

## II.2 Gán dữ liệu cho biến:

**<variable> = <expression>;**

Ví dụ:

**height = 70;**

## II.3 Thiết lập ban đầu cho biến

**<type> <name> = <expression>;**

Ví dụ:

**double height = 70;**

**double weight = 195;**

**double bmi = weight / (height \* height) \* 703;**

Mở rộng: Chúng ta có thể khởi động nhiều biến cùng 1 lúc

Cú pháp:

**<type> <name>, <name>, <name>, ..., <name>;**

Ví dụ:

**double height, weight;**

**double height = 70, weight = 195, bmi;**

## Bài thực hành số 1: Tính giá trị BMI

Viết chương trình tính BMI (Body Mass Index) theo công thức sau:

$$BMI = \frac{Weight}{Height * Height} * 703$$



Đối với người lớn từ 20 tuổi trở lên, Sử dụng bảng phân loại chuẩn cho cả nam và nữ để đánh giá chỉ số BMI.

- **BMI <16:** Gầy độ III
- **16 ≤ BMI <17:** Gầy độ II
- **17 ≤ BMI <18.5:** Gầy độ I
- **18.5 ≤ BMI <25:** Bình thường
- **25 ≤ BMI <30:** Thừa cân
- **30 ≤ BMI 35:** Béo phì độ 1
- **35 ≤ BMI <40:** Béo phì độ II
- **BMI >40:** Béo phì độ III

Bước 1: Tạo folder “thuchanh2”

Bước 2: Trong Visual Studio Code tạo file thuchanh1.java

Bước 3: Sinh viên tự viết code.

Bước 4: Kết quả như sau:

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "THUCHANH2" containing ".vscode" and "thuchanh1.java".
- Code Editor:** Displays the Java code for "thuchanh1.java".

```
public class thuchanh1{
    public static void main(String[] args) {
        // khai bao bien
        double height;
        double weight;
        double bmi;

        // tinh toan BMI
        // gan du lieu
        height=70;
        weight=195;
        bmi=weight/(height*height)*703;

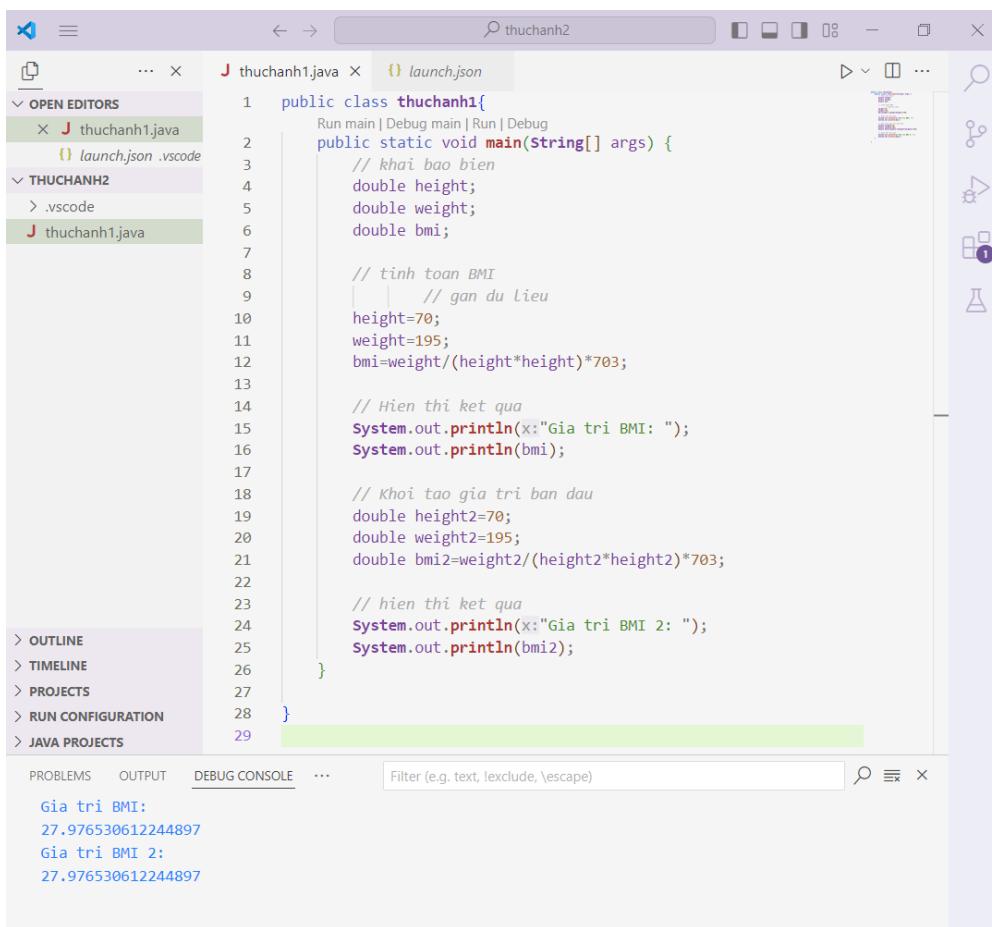
        // Hien thi ket qua
        System.out.println("Gia tri BMI: ");
        System.out.println(bmi);
    }
}
```
- Terminal:** Shows the output of the program:

```
Gia tri BMI:
27.976530612244897
```

Bước 5: Chúng ta có thể dùng khởi tạo giá trị ban đầu như sau:

```
// Khoi tao gia tri ban dau  
  
double height2=70;  
  
double weight2=195;  
  
double bmi2=weight2/(height2*height2)*703;  
  
  
// hien thi ket qua  
  
System.out.println("Gia tri BMI 2: ");  
  
System.out.println(bmi2);
```

Bước 6: Kết quả thu được



The screenshot shows the Microsoft Visual Studio Code interface with the following details:

- Editor Area:** Displays the Java code for the `thuchanh1.java` file. The code defines a class `thuchanh1` with a `main` method. It initializes variables `height`, `weight`, and `bmi`, calculates `bmi2` using the formula `weight2/(height2*height2)*703`, and prints the results.
- Terminal:** Shows the output of the program:

```
Gia tri BMI:  
27.976530612244897  
Gia tri BMI 2:  
27.976530612244897
```
- Sidebar:** Includes sections for `OPEN EDITORS`, `THUCHANH2`, `OUTLINE`, `TIMELINE`, `PROJECTS`, `RUN CONFIGURATION`, and `JAVA PROJECTS`.
- Bottom Bar:** Contains tabs for `PROBLEMS`, `OUTPUT`, `DEBUG CONSOLE`, and a search bar.

## II.4 Nối chuỗi

Kết hợp nhiều chuỗi thành một chuỗi duy nhất hoặc kết hợp một chuỗi với dữ liệu khác thành một chuỗi mới dài hơn.

Ví dụ:

```
System.out.println("I have " + 3 + " things to concatenate");
```

### Bài thực hành số 2: Nối chuỗi



Tiếp tục bài thực hành số 1 như sau:

```
System.out.println("Gia tri BMI 2: " + bmi2);
```

The screenshot shows the Visual Studio Code interface with the following details:

- Left Sidebar:** Shows the project structure with files: `thuchanh1.java`, `launch.json`, `THUCHANH2`, `.vscode`, and `J thuchanh1.java`.
- Editor Area:** Displays Java code for calculating BMI and concatenating strings.

```
1 public class thuchanh1{  
2     public static void main(String[] args) {  
3         // khai bao bien  
4         double height;  
5         double weight;  
6         double bmi;  
7  
8         // tinh toan BMI  
9         // gan du lieu  
10        height=70;  
11        weight=195;  
12        bmi=weight/(height*height)*703;  
13  
14        // Hien thi ket qua  
15        System.out.println("Gia tri BMI: ");  
16        System.out.println(bmi);  
17  
18        // Khoi tao gia tri ban dau  
19        double height2=70;  
20        double weight2=195;  
21        double bmi2=weight2/(height2*height2)*703;  
22  
23        // hien thi ket qua  
24        System.out.println("Gia tri BMI 2: ");  
25        System.out.println(bmi2);  
26        System.out.println();  
27        // dung noi chuoi  
28        System.out.println("Gia tri BMI 2: " + bmi2);  
29    }  
30 }  
31 }
```
- Bottom Status Bar:** Shows the output of the code execution:

```
Gia tri BMI:  
27.976530612244897  
Gia tri BMI 2:  
27.976530612244897  
Gia tri BMI 2: 27.976530612244897
```

Chú ý: Khi thực hiện phép toán trong nối chuỗi phải thêm dấu ( và ) để tránh kết quả bị sai.

Xét ví dụ sau:

**System.out.println(5 + " hello " + 7 + 6);**

Kết quả sẽ là:

**5 hello 76**

Lý do:

5 + " hello " + 7 + 6  
  \underbrace{"hello"}\_{"5 hello"} + 7 + 6

"5 hello " + 7 + 6  
  \underbrace{"5 hello"}\_{"5 hello 7"} + 6

"5 hello 7" + 6  
  \underbrace{"5 hello 7"}\_{"5 hello 76"} + 6

Sinh viên tự check lại.

Tiếp tục, nếu chúng ta thêm ( và ) sẽ được

**System.out.println(5 + " hello " + (7 + 6));**

Kết quả sẽ là:

**5 hello 13**

### III. TĂNG (++) VÀ GIẢM (--)

Ví dụ:

**x++;**

**y--;**

**x += 1;**

**y -= 1;**

**$z += 2;$**

**$x *= 2;$**

**$y *= 3;$**

**$z /= 2;$**

Thứ tự ưu tiên

Miêu tả	Kiểu
Phép đơn lẻ	<b><math>++, --, +, -</math></b>
Đa nhiệm	<b><math>*, /, %</math></b>
Phép cộng	<b><math>+, -</math></b>
Phép gán	<b><math>=, +=, -=, *=, /=, %=</math></b>

### Bài thực hành số 3: Phép toán ++ và --



Sinh viên thực hiện code test cho giá trị z dòng sau

```
int x = 10;
int y = 20;
int z = ++x * y--;
```

Sinh viên hiển thị kết quả của các bài toán sau: (có thể ở lớp làm 1 số câu, về nhà sinh viên hoàn thành phần còn lại)

- a)  $(5 * 7.0 / 2 - 2.5) / 5 * 2$
- b)  $41\% 7 * 3 / 5 + 5 / 2 * 2.5$ 
  - a.  $/2/4$
- c)  $8 / 5 + 13 / 2 / 3.0$
- d)  $(2.5 + 3.5) / 2$
- e)  $9 / 4 * 2.0 - 5 / 4$
- f)  $9 / 2.0 + 7 / 3 - 3.0 / 2$
- g)  $813 \% 100 / 3 + 2.4$
- h)  $27 / 2 / 2.0 * (4.3 + 1.7) - 8 / 3$
- i)  $53 / 5 / (0.6 + 1.4) / 2 + 13 / 2$
- j)  $2 * 3 / 4 * 2 / 4.0 + 4.5 - 1$
- k)  $89 \% 10 / 4 * 2.0 / 5 + (1.5 + 1.0 / 2) * 2$

- l) "2 + 2 " + (3 + 4)
- m) "(2 + 2) " + (3 + 4)
- n) "hello 34 " + 2 \* 4
- o) 2 + "(int) 2.0" + 2 \* 2 + 2
- p) 4 + 1 + 9 + "." + (-3 + 10) + 11 / 3
- q) 8 + 6 \* -2 + 4 + "0" + (2 + 5)
- r) 1 + 1 + "8 - 2" + (8 - 2) + 1 + 1
- s) 5 + 2 + "(1 + 1)" + 4 + 2 \* 3
- t) "1" + 2 + 3 + "4" + 5 \* 6 + "7" + (8 + 9)

### III.2 Biến và trộn các định dạng

Chúng ta xét ví dụ sau:

*int x;*

*double y;*

*x = 2 + 3;*

*y = 3.4 \* 2.9;*

Nhưng khi chúng ta khai báo như sau:

*x = 3.4 \* 2.9; // báo lỗi*

Vì sao? .....

Để chính xác ta cần thay đổi như sau:

*x = (int) (3.4 \* 2.9);*

## IV. VÒNG LẶP *FOR*

Cú pháp tổng quát của vòng lặp *for* như sau:

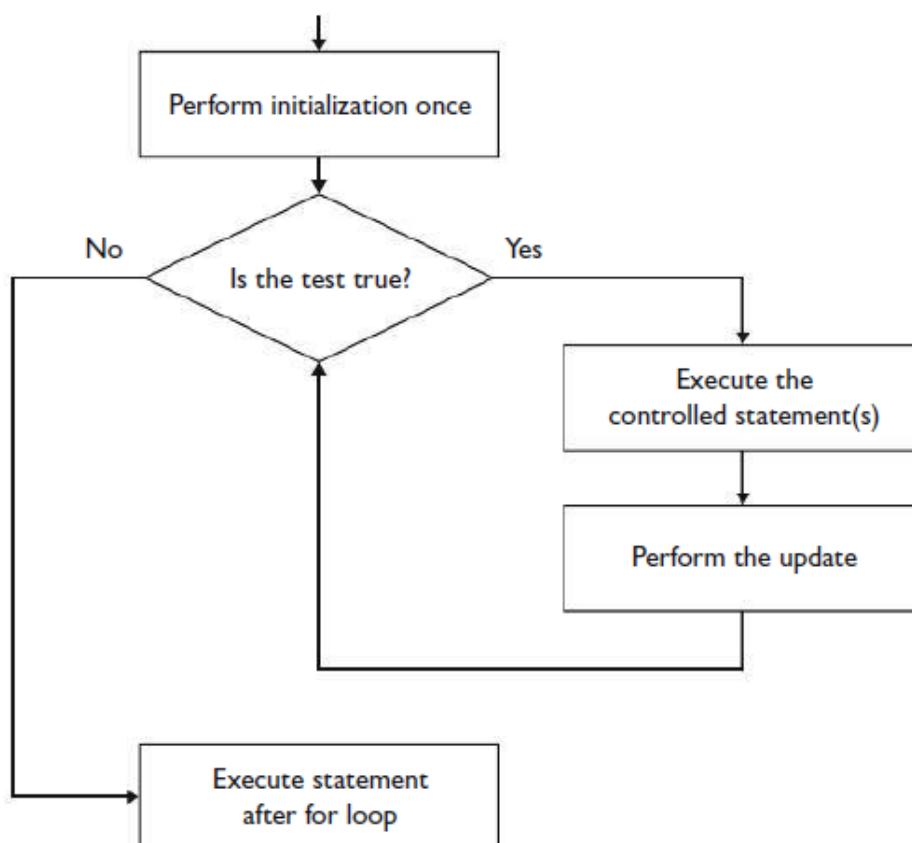
```
for (<initialization>; <continuation test>; <update>) {  
    <statement>;  
    <statement>;  
    ...
```

```
<statement>;  
}  
  
for (int <variable> = n; <variable> >= 1; <variable>--) {  
<statement>;  
<statement>;  
...  
<statement>;  
}
```

Số vòng lặp được tính theo công thức sau:

$<\text{ending value}> - <\text{starting value}> + 1$

Quá trình xử lý như sau:



## Bài thực hành số 4: Tính bình phương từ 1 tới 10



Phân tích bài toán:

- Chúng ta thực hiện phép toán giống nhau là lấy bình phương:  $x*x$
- Bài toán lặp lại từ 1 tới 10

Bước 1: Tạo file “thuchanh2.java”

Bước 2: Tạo code như sau:

```
public class thuchanh2 {  
    public static void main(String[] args) {  
        for (int i=1;i<=10;i++)  
        {  
            System.out.println(i+ " Bình phuong la: " + (i * i));  
        }  
    }  
}
```

Bước 3: Kết quả thu được

```
public class thuchanh2 {  
    public static void main(String[] args) {  
        for (int i=1;i<=10;i++)  
        {  
            System.out.println(i+ " Bình phuong la: " + (i * i));  
        }  
    }  
}
```

Output in Terminal:

```
1 Bình phuong la: 1  
2 Bình phuong la: 4  
3 Bình phuong la: 9  
4 Bình phuong la: 16  
5 Bình phuong la: 25  
6 Bình phuong la: 36  
7 Bình phuong la: 49  
8 Bình phuong la: 64  
9 Bình phuong la: 81  
10 Bình phuong la: 100
```

### Bài thực hành số 5: Vẽ hình sau sử dụng vòng lặp



```
+-----+  
+-----+  
+-----+  
+-----+  
+-----+  
+-----+  
+-----+  
+-----+  
+-----+  
+-----+
```

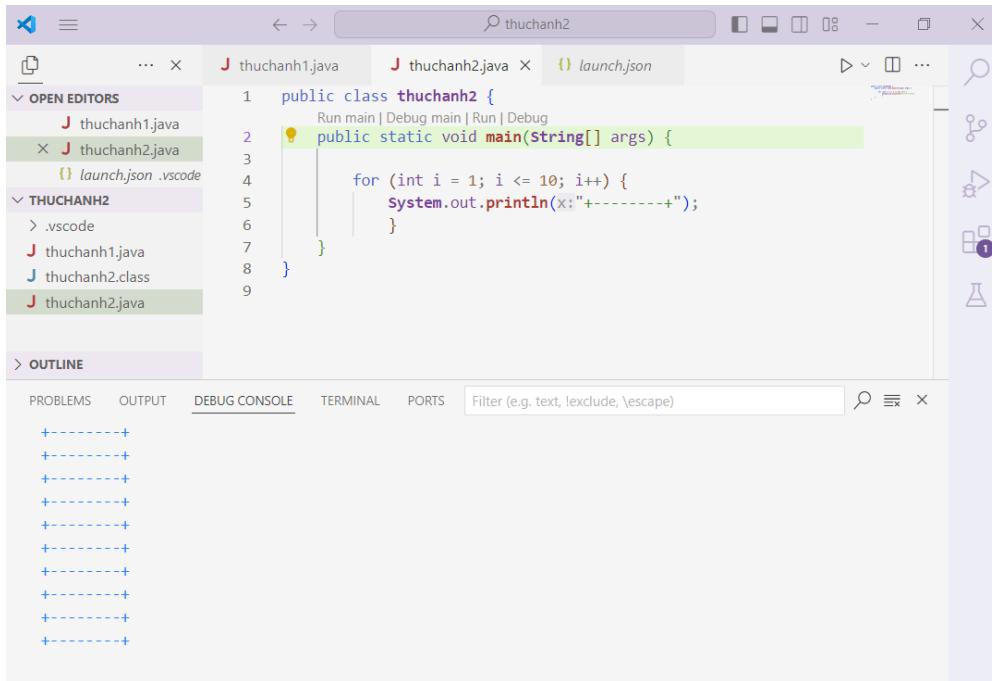
Phân tích bài toán:

- Số dòng lặp lại là **10** dòng
- Dòng lặp lại “+-----+

Bước 1: Tạo code sau

```
for (int i = 1; i <= 10; i++) {  
    System.out.println("+-----+");  
}
```

Bước 2: Sinh viên kiểm tra kết quả.



The screenshot shows a Java program in a VS Code editor. The code prints a diamond shape using nested loops. The editor interface includes tabs for thuchanh1.java, thuchanh2.java, and launch.json, and a sidebar with an outline of files like .vscode, thuchanh1.java, thuchanh2.java, and thuchanh2.class.

```
public class thuchanh2 {
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i++) {
            System.out.println("-----");
            for (int j = 1; j <= i; j++) {
                System.out.print(j + " ");
            }
            for (int k = i - 1; k >= 1; k--) {
                System.out.print(k + " ");
            }
            System.out.println();
        }
    }
}
```

Bài tập tương tự: Sinh viên in hình sau

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

## V. VÒNG LẶP TRONG VÒNG LẶP

Ví dụ như sau:

```
for (int i = 1; i <= 6; i++) {
    for (int j = 1; j <= 3; j++) {
        System.out.print(j + " ");
    }
}
```

}

? Sinh viên chạy và kiểm tra kết quả

### Bài thực hành số 6: Vẽ hình sau sử dụng vòng lặp của vòng lặp



```
*****  
*****  
*****  
*****  
*****  
*****  
*****
```

Phân tích:

- Có 6 hàng cần hiển thị
- Trong 6 hàng thì có 10 dấu \* xuất hiện

Bước 1: Tiếp tục file “thuchanh1.java”

Bước 2: Sử dụng 2 biến i, j để tạo vòng lặp

```
for (int i = 1; i <= 6; i++) {
```

```
    for (int j = 1; j <= 10; j++) {  
        System.out.print("*");  
    }
```

```
    System.out.println();
```

```
}
```

Bước 3: Sinh viên chạy và kiểm tra kết quả

### Bài thực hành số 7: Vẽ hình sao sử dụng vòng lặp của vòng lặp



\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*

Phân tích bài toán: Ở đây chúng ta đếm và thống kê qua bảng sau

Dòng	Số khoảng trắng	Số ký tự *
1	0	9
2	1	7
3	2	5
4	3	3
5	4	1

Bây giờ: Bằng phương pháp toán học, hãy tìm mối liên hệ giữa cột khoảng trắng và số ký tự \* theo dòng.

- 1) Số khoảng trắng: **dòng – 1**
- 2) Số ký tự \* = **11 - 2 \* dòng**

Tips: *đặt số ký tự = a \* dòng – b; Sau đó tìm a và b ta được biểu thức trên*

Code như sau:

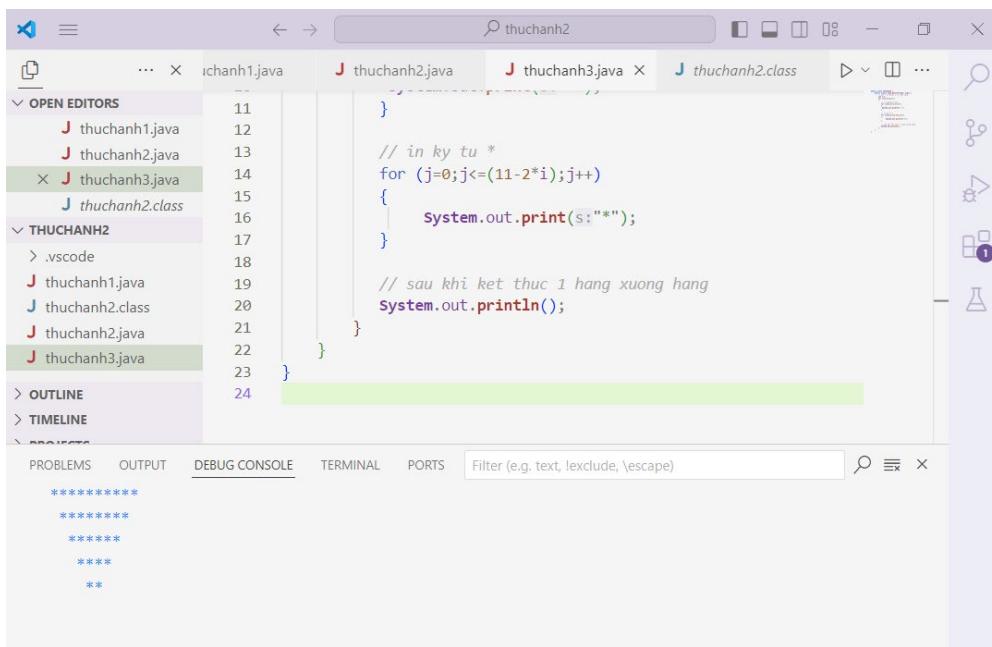
Bước 1: Tạo file mới “thuchanh3.java”

Bước 2:

```
public class thuchanh3 {  
    public static void main(String[] args) {  
        // dat i la dong; j la bien chay  
        int i,j;  
        for (i=1;i<=5;i++)
```

```
{  
    // in khoang trang  
    for (j=I;j<=(i-1);j++)  
    {  
        System.out.print(" ");  
    }  
  
    // in ky tu *  
    for (j=1;j<=(11-2*i);j++)  
    {  
        System.out.print("*");  
    }  
  
    // sau khi ket thuc 1 hang xuong hang  
    System.out.println();  
}  
}  
}
```

Kết quả thu được như sau:



The screenshot shows a VS Code interface with the following details:

- Open Editors:** thuchanh1.java, thuchanh2.java, thuchanh3.java, thuchanh2.class.
- Code Editor:** The code for `thuchanh3.java` is displayed, which prints a diamond shape. The code uses nested loops and `System.out.print` to output asterisks (\*).
- Terminal:** Shows the output of the program:

```
*****
 ****
  ***
   *
  ***
 ****
*****
```

### Bài thực hành số 8: Vẽ hình sao sử dụng vòng lặp của vòng lặp



```
*  
***  
*****  
*****  
*****
```

Phân tích bài toán: Bài toán này là ngược bài toán trên nên thay vì chạy từ 1 tới 5 thì chúng ta chỉnh ngược lại chạy từ 5 mà giảm xuống. Cụ thể như sau:

*for (i=5;i>=1;i--)*

```
{  
    // in khoang trang  
    for (j=1;j<=(i-1),j++)  
    {  
        System.out.print(" ");  
    }
```

```
// in ky tu *
for (j=I;j<=(11-2*i);j++)
{
    System.out.print("*");
}

// sau khi ket thuc 1 hang xuong hang
System.out.println();
}
```

Sinh viên tự làm và kết quả thu được như sau:

## Thực hành: Cấu trúc Dữ liệu và Giải thuật

The screenshot shows a Java code editor in VS Code displaying a program named `thuchanh3.java`. The code prints a diamond pattern of asterisks (\*) to the console. The editor interface includes tabs for `thuchanh1.java`, `thuchanh2.java`, `thuchanh3.java`, and `thuchanh2.class`. The code uses nested loops to print the pattern. The terminal below shows the output of running the program.

```
public class thuchanh3 {
    public static void main(String[] args) {
        // dat i la dong; j la bien chay
        int i,j;
        for (i=1;i<=5;i++)
        {
            // in khoang trang
            for (j=1;j<=(i-1);j++)
            {
                System.out.print(" ");
            }

            // in ky tu *
            for (j=1;j<=(11-2*i);j++)
            {
                System.out.print("*");
            }

            // sau khi ket thuc 1 hang xuong hang
            System.out.println();
        }
        for (i=5;i>=1;i--)
        {
            // in khoang trang
            for (j=1;j<=(i-1);j++)
            {
                System.out.print(" ");
            }

            // in ky tu *
            for (j=1;j<=(11-2*i);j++)
            {
                System.out.print("*");
            }

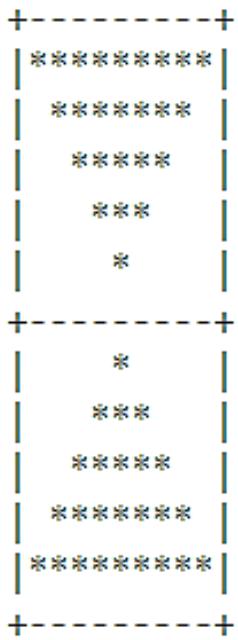
            // sau khi ket thuc 1 hang xuong hang
            System.out.println();
        }
    }
}
```

The terminal window shows the output:

```
'C:\Users\ADMIN\AppData\Roaming\Code\User\workspaceStorage\aa688243a4beff2152eadcd7736ff1a5fc\redhat.java\jdt_ws\thuchanh2_7ca0cb4b\bin' 'thuchanh3'
*****
*****
*
*
***
*****
*****
*****
PS C:\Users\ADMIN\Desktop\thuchanh2>
```

Bài tập thực hành tại lớp:

Dựa vào bài thực hành 6 phát triển thành hình sau:



## Thực hành: Cấu trúc Dữ liệu và Giải thuật

The screenshot shows a Java code editor in VS Code displaying a program to print a diamond shape. The code uses nested loops and `System.out.println` statements. The editor interface includes a sidebar with project navigation, a bottom status bar showing file paths and indexing status, and a terminal window at the bottom.

```
// dat i la dong; j la bien chay
int i,j;
System.out.println("-----");
for (i=1;i<=5;i++)
{
    // in khoang trang
    System.out.print(" ");
    for (j=1;j<=(i-1);j++)
    {
        System.out.print(" ");
    }

    // in ky tu *
    for (j=1;j<=(11-2*i);j++)
    {
        System.out.print("*");
    }

    for (j=1;j<=(i-1);j++)
    {
        System.out.print(" ");
    }

    System.out.print(" ");
    // sau khi ket thuc 1 hang xuong hang
    System.out.println();
}

System.out.println("-----");

for (i=5;i>=1;i--)
{
    System.out.print(" ");
    // in khoang trang
    for (j=1;j<=(i-1);j++)
    {
        System.out.print(" ");
    }

    // in ky tu *
    for (j=1;j<=(11-2*i);j++)
    {
        System.out.print("*");
    }

    for (j=1;j<=(i-1);j++)
    {
        System.out.print(" ");
    }

    System.out.print(" ");
    // sau khi ket thuc 1 hang xuong hang
    System.out.println();
}

System.out.println("-----");
```

PROBLEMS    2    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

3a4beff2152eacd7736ff1a5fc\redhat.java\jdt\_ws\thuchanh2\_7ca0cb4b\bin' 'thuchanh3'

+-----+  
| \*\*\*\*\* |  
| \* \* \* \* |  
| \* \* \* |  
| \* \* |  
| \* |  
+-----+  
| \* |  
| \* \* |  
| \* \* \* |  
| \* \* \* \* |  
| \*\*\*\*\* |  
+-----+

PS C:\Users\ADMIN\Desktop\thuchanh2>

X ⑧ 2 △ 0 ⌂ 0 Current File (thuchanh2) Indexing completed. ☰ Java: Ready Ln 33, Col 27 Spaces: 4 UTF-8 CRLF { } Java Q

### Bài thực hành số 9: Vẽ hình sau sử dụng vòng lặp của vòng lặp và hàm tĩnh



```
+-----+  
| \.... / |  
| \... / |  
| \ \ / |  
| \ / |  
| / ... \ |  
| /....\ |  
+-----+
```

Phân tích bài toán:

- 1) Chúng ta chia bài toán thành các phân khúc (hàm tĩnh) sau

+-----+	line
\.... /     \... /     \ \ /	top half
\ /     / ... \     /....\	bottom half
+-----+	line

- 2) Phân tích hàm line

Bắt đầu là ký tự “+”

Sau đó là 6 ký tự “-“

Kết thúc là ký tự “+”

Sinh viên dựa vào trên viết code

```
public static void drawLine() {
```

}

3) Phân tích top



Lập bảng như sau

Dòng	Khoảng trắng	Ký tự “.”	Khoảng trắng
1	1	4	1
2			
3			

Bây giờ: Bằng phương pháp toán học, hãy tìm mối liên hệ giữa cột khoảng trắng và số ký tự \* theo dòng.

- a) Số khoảng trắng: .....  
b) Số ký tự . = .....

Sinh viên tự làm code.

- 4) Phân tích bottom: Bottom là ngược của phần top (thực hành số 6)  
5) Kết thúc là line giống phần 1)

Sinh viên tự làm và kiểm tra kết quả.

**Bài thực hành số 10: Vẽ hình sau sử dụng vòng lặp của vòng lặp và hàm tĩnh**



```
+-----+
| \...../
| \..../
| \..../
| \.../
| \
| \
| /...
| /...
| /....\
| /....\
| /....\
+-----+
| \
| /...
| /....\
| /....\
| /....\
| \....\
| \....\
| \.../
| \
+-----+
```

## BÀI TẬP THỰC HÀNH TẠI NHÀ.

1) Vẽ hình sau:

!!!!!! !!!!! !!!!! !!!!!  
\\!!!!!! !!!!! //  
\\\\\\\\!!!!!! // / / /  
\\\\\\\\\\\\\\\\!!!! // / / / /

## Bài tập 2: Vẽ hình sau

-----1-----  
-----333-----  
-----55555-----  
--7777777--  
-999999999-

### Bài tập 3: Vẽ hình sau:

### Bài tập 4: Vẽ hình sau:

## Bài tập 5: Vẽ hình sau:

## Bài tập 6: Vẽ hình sau:

```

+-----+
|      ^^   |
|      ^     ^ |
| ^       ^    |
|      ^^   |
|      ^     ^ |
| ^       ^    |
+-----+
| v       v |
| v       v |
|      vv   |
| v       v |
| v       v |
|      vv   |
+-----+

```

Bài tập 7: Vẽ hình sau:

```
+-----+  
|      *      |  
|    /*\     |  
|   //*\\"    |  
|  ///*\\\"   |  
| \\\/*///   |  
| \\\*//    |  
|  \*/     |  
|    *     |  
+-----+  
| \\\/*///   |  
| \\\*//    |  
|  \*/     |  
|    *     |  
|    *     |  
|  /*\     |  
| //*\\"    |  
|  ///*\\\"   |  
+-----+
```

Bài tập 8: Vẽ hình sau:

```
| ##### |  
 \:::/:/  
  \:::/  
   \::/  
    \::/  
     ||  
      /:\\  
       /:::\  
        /:::::\  
         /:::::::\  
          /:::::::\  
| ##### |
```

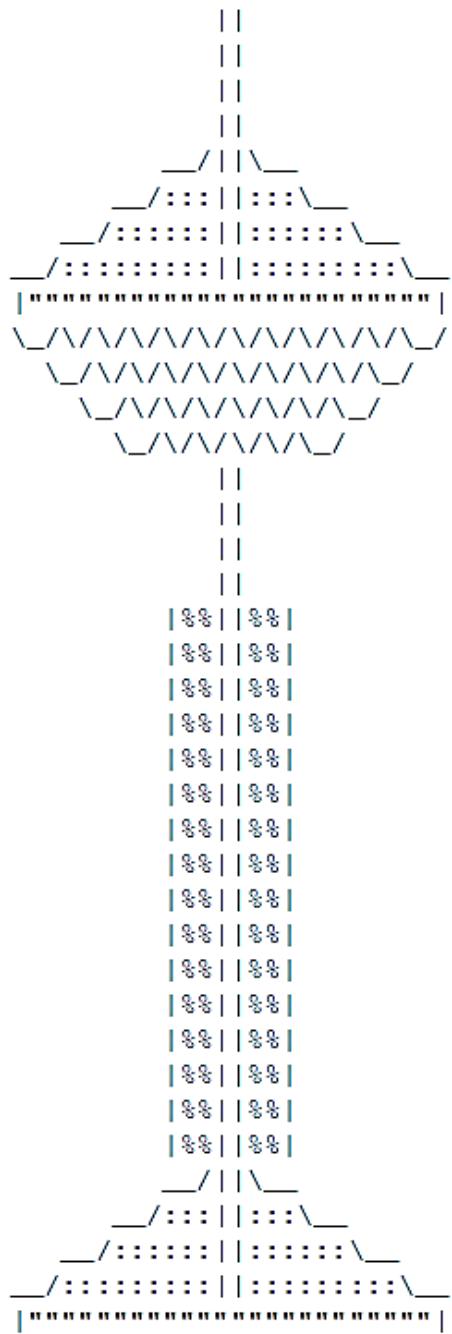
Bài tập 9: Vẽ hình sau

```
          * * * * * * *
          ○   / | \   *
          / \   *   *
          ○   * * * * * *
          / | \   *
          / \   *   *
          ○   * * * * * *
          / | \   *
          / \   *   *
          ○   * * * * * *
          / | \   *
          / \   *   *
          ○   * * * * * *
          / | \   *
          / \   *   *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

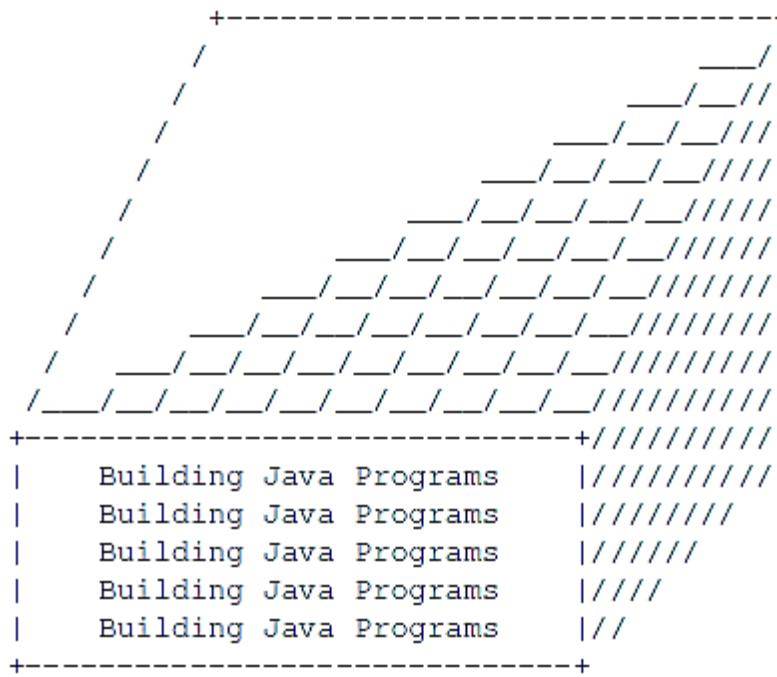
Bài tập 10: Vẽ hình sau

```
/* *\n//**\\ \n///**\\ \\ \n////**\\ \\ \\ \n//////**\\ \\ \\ \\ \n\n+==*=*=*=*=*=*=*\n| .. / \\ . . . / \\ . . |\n| . / \\ / \\ . . / \\ / \\ . |\n| / \\ / \\ / \\ / \\ / \\ / \\ |\n| \\ / \\ / \\ / \\ / \\ / \\ / |\n| . \\ / \\ / . . \\ / \\ / . |\n| .. \\ / . . . . \\ / . . |\n+==*=*=*=*=*=*=*\n| \\ / \\ / \\ / \\ / \\ / \\ / |\n| . \\ / \\ . . \\ / \\ / . |\n| .. \\ / . . . . \\ / . . |\n| .. / \\ . . . . / \\ . . |\n| . / \\ / \\ . . / \\ / \\ . |\n| / \\ / \\ / \\ / \\ / \\ / |\n+==*=*=*=*=*=*\n\n/* *\n//**\\ \n///**\\ \\ \n////**\\ \\ \\ \n//////**\\ \\ \\ \\ \n
```

Bài tập 11: Vẽ hình sau:



Bài tập 12: Vẽ hình sau:



Tài liệu tham khảo

- [1] <https://aws.amazon.com/vi/what-is/java/>
- [2] <https://topdev.vn/blog/tong-quan-ve-ngon-Ngu-lap-trinh-java/>
- [3] <https://fptshop.com.vn/tin-tuc/danh-gia/java-la-gi-156801>
- [4]
- [5] <https://vietnix.vn/java-la-gi/>