

Bài 1

TỔNG QUAN VỀ JAVA

Company: DEVPRO VIỆT NAM

Website: devpro.edu.vn

Design by Minh An

Nội dung

- Giới thiệu ngôn ngữ lập trình java
- Nền tảng java – java platform
- Một số khái niệm
- Chương trình java đầu tiên
- Hằng và biến trong java
- Các kiểu dữ liệu cơ bản
- Định dạng dữ liệu đầu ra
- Định dạng dữ liệu đầu vào

Design by Minh An

Giới thiệu ngôn ngữ lập trình java

- **Java là gì?**

- ❑ Java là một ngôn ngữ lập trình cao cấp, ban đầu được phát triển bởi Sun Microsystems và được phát hành vào năm 1995. Java chạy trên nhiều nền tảng, chẳng hạn như Windows, Mac OS và các phiên bản khác nhau của UNIX.
- ❑ Phiên bản mới nhất của Java Standard Edition là Java SE 8. Với sự tiến bộ của Java và sự phổ biến rộng rãi của nó, nhiều cấu hình đã được xây dựng để phù hợp với nhiều loại nền tảng khác nhau. Ví dụ: J2EE cho các ứng dụng doanh nghiệp, J2ME cho các ứng dụng di động.
- ❑ Các phiên bản J2 mới đã được đổi tên thành Java SE, Java EE và Java ME. Java được đảm bảo là Write Once, Run Anywhere.

Design by Minh An

Biểu tượng java

- Java là tên gọi của một hòn đảo ở Indonesia, Đây là nơi nhóm nghiên cứu phát triển đã chọn để đặt tên cho ngôn ngữ lập trình Java trong một chuyến đi tham quan và làm việc trên hòn đảo này.
- Hòn đảo Java này là nơi rất nổi tiếng với nhiều khu vườn trồng cafe, đó chính là lý do chúng ta thường thấy biểu tượng ly cafe trong nhiều sản phẩm phần mềm, công cụ lập trình Java của Sun cũng như một số hãng phần mềm khác đưa ra.



Design by Minh An

Lịch sử ra đời



- James Gosling đã bắt đầu dự án ngôn ngữ Java vào tháng 6 năm 1991 để sử dụng trong một dự án set-top box của ông. Ngôn ngữ, ban đầu được gọi là 'Oak' (cây sồi nằm bên ngoài văn phòng của Gosling), cũng có tên 'Green' và sau đó được đổi tên thành Java, từ một danh sách các từ ngẫu nhiên.
- Sun lần đầu tiên công khai Java 1.0 vào năm 1995. Nó hứa hẹn một khái niệm Write Once, Run Anywhere (WORA), cung cấp thời gian chạy không tốn kém trên nền tảng phổ biến.
- Ngày 13 tháng 11 năm 2006, Sun Microsystems phát hành Java như là phần mềm miễn phí và mã nguồn mở theo các điều khoản của GNU General Public License (GPL).
- Vào ngày 8 tháng 5 năm 2007, Sun hoàn tất quá trình này, tạo ra toàn bộ mã nguồn mở Java miễn phí, ngoại trừ một phần nhỏ mã mà Sun không giữ bản quyền.

Design by Minh An

Đặc trưng cơ bản của java



- **Hướng đối tượng** – Trong Java, mọi thứ đều là một Object. Java có thể dễ dàng mở rộng vì nó được dựa trên mô hình Object.
- **Độc lập nền tảng** – Không giống nhiều ngôn ngữ lập trình khác như C và C ++, khi Java được biên dịch, nó không được biên dịch vào nền tảng máy tính cụ thể, thay vào đó là mã byte nền tảng độc lập. Mã byte này được phân phát trên web và được thông dịch bởi Virtual Machine (JVM) trên nền tảng nào đó mà nó đang chạy.
- **Đơn giản** – Java được thiết kế rất dễ học. Nếu bạn hiểu khái niệm cơ bản của OOP Java, bạn sẽ rất dễ làm chủ nó.
- **Bảo mật** – Với tính năng an toàn của Java, nó cho phép phát triển các hệ thống không có virus. Các kỹ thuật xác thực dựa trên key mã hoá khóa công khai.

Design by Minh An

Đặc trưng cơ bản của java



- **Di động** – Là kiến trúc trung lập và không bị phụ thuộc làm cho Java có thể mang đi dễ dàng. Trình biên dịch trong Java được viết bằng ANSI C với khả năng di chuyển sạch, đó là một tập hợp con POSIX.
- **Mạnh mẽ** – Java nỗ lực để loại trừ các tình huống dễ bị lỗi bằng cách nhấn mạnh việc kiểm tra lỗi thời gian biên dịch và kiểm tra thời gian chạy.
- **Đa luồng** – Với tính năng đa luồng của Java, có thể viết các chương trình có thể thực hiện nhiều tác vụ đồng thời. Tính năng thiết kế này cho phép các nhà phát triển xây dựng các ứng dụng tương tác có thể chạy trơn tru.

Design by Minh An

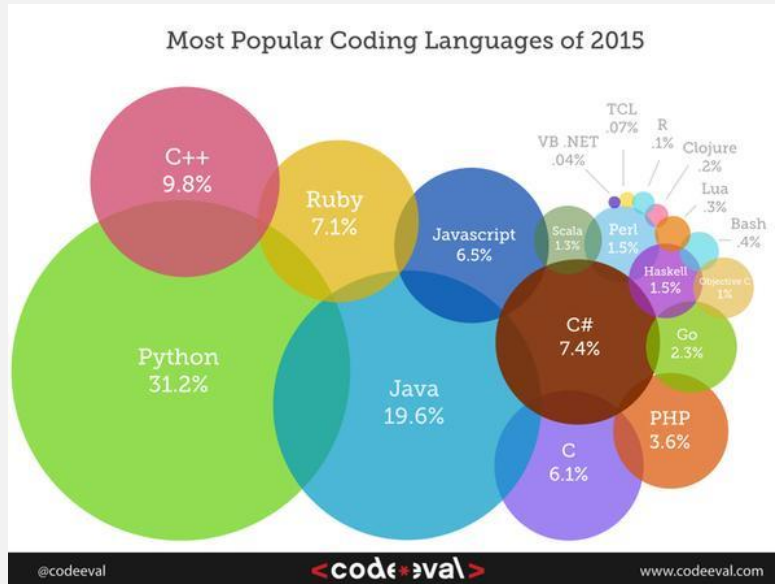
Đặc trưng cơ bản của java



- **Phiên dịch** – Mã byte Java được dịch trực tiếp tới các hướng dẫn máy tính và không được lưu trữ ở bất cứ đâu. Quá trình phát triển nhanh hơn và phân tích hơn.
- **Hiệu năng cao** – Với việc sử dụng trình biên dịch Just-In-Time, Java cho phép thực hiện chương trình với hiệu năng cao.
- **Phân phối** – Java được thiết kế cho môi trường phân tán của internet.
- **Năng động** – Java được xem là năng động hơn C hoặc C ++ vì nó được thiết kế để thích nghi với môi trường đang phát triển. Các chương trình Java có thể mang một lượng lớn thông tin run-time, có thể được sử dụng để xác minh và giải quyết các truy cập vào các đối tượng trong thời gian chạy.

Design by Minh An

Thị phần



Design by Minh An

Nền tảng java – Java Platform



- Nền tảng là một môi trường phần cứng hoặc phần mềm mà ở đó các chương trình có thể chạy.
- Nền tảng Java có thể được xem như là một bộ máy thực thi giống như một máy ảo vì nó không phải là một hệ điều hành hay phần cứng cụ thể.
- Nền tảng Java gồm có hai thành phần quan trọng:
 - a. Máy ảo Java (Java Virtual Machine - JVM)
 - b. Giao tiếp lập trình ứng dụng Java (API).

Design by Minh An

Máy ảo java – Java Virtual Machine



- Máy ảo Java (JVM) là một môi trường thực thi và nó đã có sẵn cho các hệ điều hành khác nhau.
- Máy ảo Java thực thi các chương trình Java đã được biên dịch (dạng byte codes).
- Nó tạo thành một lớp trừu tượng đối với:
 - Nền tảng phần cứng bên dưới.
 - Hệ điều hành.
 - Các mã đã được biên dịch.

Design by Minh An

Java API – Application Programming Interface



- Bộ giao tiếp lập trình ứng dụng (APIs) chứa một lượng lớn các thư viện các lớp và các thành phần phần mềm khác như là các giao tiếp.
- Nó là một phần của bộ Java SDK.

Design by Minh An

Các nền tảng Java khác nhau



- **Có 4 nền tảng cơ bản:**
 - ❑ Java Platform, Standard Edition (Java SE)
 - ❑ Java Platform, Enterprise Edition (Java EE)
 - ❑ Java Platform, Micro Edition (Java ME)
 - ❑ JavaFX

Design by Minh An

Các nền tảng Java khác nhau



- Tất cả các platforms đều bao gồm một Máy Ảo Java (JVM - Java Virtual Machine) và một bộ thư viện API riêng của nó (application programming interface).
- Máy ảo java là một chương trình chạy trên một nền tảng phần cứng, và hệ điều hành cụ thể, máy ảo sẽ chạy các ứng dụng Java.
- API là bộ thư viện lập trình mà chúng ta sử dụng để tạo nên các thành phần khác nhau của ứng dụng java của mình. Mỗi nền tảng java sẽ cung cấp một máy ảo và một API riêng, việc này giúp cho các ứng dụng viết cho các nền tảng đó có thể chạy được trên bất cứ hệ thống thích hợp nào với tất cả ưu điểm của ngôn ngữ java: độc lập nền tảng, mạnh mẽ, ổn định, dễ học, và bảo mật.

Design by Minh An

Java SE – Java Platform, Standard Edition



- Java SE là phiên bản mà hầu hết mọi người nghĩ tới, khi nói tới ngôn ngữ lập trình Java.
- Java SE API cung cấp các tính năng cốt lõi của Java. Nó định nghĩa mọi thứ từ các kiểu dữ liệu cơ bản cho đến các đối tượng, và các class bậc cao dùng cho việc kết nối mạng, xử lý bảo mật, truy cập database, bóc tách XML, và xử lý giao diện đồ họa (GUI).
- Ngoài API lõi, Java SE còn bao gồm cả máy ảo, bộ công cụ phát triển, và các thư viện hỗ trợ cho việc lập trình ứng dụng Java.

Design by Minh An

Java EE – Java Platform, Enterprise Edition



- Java EE được xây dựng dựa trên sự mở rộng từ Java SE.
- Java EE platform cung cấp API, runtime environment chuyên biệt cho việc phát triển và thực thi các ứng dụng quy mô lớn (large-scale), đa tầng (multi-tiered), có độ tin cậy cao (reliable), bảo mật tốt (secure network).

Design by Minh An

Java ME – Java Platform, Micro Edition



- Java ME là phiên bản rút gọn của Java SE.
- Java ME platform cung cấp một bộ API và máy ảo thực thi nhỏ gọn phù hợp để chạy các ứng dụng java trên các thiết bị nhỏ như điện thoại, máy nhắn tin, máy tính bỏ túi...
- Java ME là tập con của Java SE cộng thêm với một số class đặc biệt phục vụ cho việc phát triển ứng dụng chạy trên các thiết bị nhỏ.
- Các ứng dụng Java ME thường là client của các dịch vụ chạy trên Java EE.

Design by Minh An

JavaFX



- JavaFX là nền tảng dùng để phát triển các ứng dụng java chạy trên nền Web sử dụng một giao diện đồ họa nhỏ gọn.
- JavaFX sử dụng các engine tăng tốc phần cứng đồ họa và xử lý media để đạt hiệu quả thực thi cao hơn cùng với một giao diện mượt mà.
- JavaFX cũng cung cấp một tập các API chuyên biệt để xử lý truyền tải dữ liệu qua mạng.
- Các ứng dụng JavaFX thường là client của các dịch vụ web chạy trên nền Java EE.

Design by Minh An

Một số khái niệm



- SDK (Software Development Kit) thường là khái niệm chỉ một bộ công cụ lập trình, tiện ích, tài liệu và các thư viện API. Đối với java ta có JDK
- JVM (Java virtual machine) - Máy ảo java.
- JRE (Java Runtime Environment) – môi trường thực thi Java, cung cấp các Java API, máy ảo Java (Java Virtual Machine hay JVM) và các thành phần cần thiết khác.
- Java Runtime library là thư viện các class đã được compiled sẵn và đặt trong file `jre\lib\rt.jar`. Các class `java.lang.String`, `java.lang.Object`, ... nằm trong gói jar này.
- JRE = JVM + Java Runtime Library: bạn sẽ thấy `java.exe`, `javaw.exe` và `rt.jar` trong thư mục cài đặt jre.
- JDK = JRE + tools (`javac.exe`, `keytool.exe`, ...) + document (help, samples,...) + `src.zip`.

Design by Minh An

Chương trình java đầu tiên



- Một chương trình java luôn phải có 1 hàm main, và chỉ có duy nhất 1 hàm main().
- Hàm main là nơi mà chương trình bắt đầu chạy.
- Hàm main() này phải được đặt trong 1 class, class được khai báo trong 1 file đuôi `.java` có tên file trùng với tên class.

```
1 package vn.devpro.example;
2
3 public class HelloWorld {
4
5     public static void main(String[] args) {
6         // in 1 câu hello ra màn hình
7         System.out.println("Hello World");
8     }
9
10 }
```

Design by Minh An

Chương trình java đầu tiên



- Để in 1 câu thông báo ra màn hình console ta dùng lệnh: `System.out.println("message");`
- Câu thông báo được đặt trong cặp ngoặc tròn và được bao bởi một cặp dấu nháy kép.

```
1 package vn.devpro.example;
2
3 public class HelloWorld {
4
5     public static void main(String[] args) {
6         // in 1 câu hello ra màn hình
7         System.out.println("Hello World");
8     }
9
10 }
```

Design by Minh An

Hằng và biến trong java



- Biến được sử dụng trong chương trình Java để lưu trữ dữ liệu mà giá trị của nó thay đổi trong quá trình thực thi chương trình.
- Biến phải được khai báo trước khi sử dụng
- Khai báo biến gồm 3 thành phần:
 - Kiểu dữ liệu
 - Tên biến
 - Giá trị khởi tạo (không bắt buộc)

- **Cú pháp:**

```
kiểu_dữ_liệu  tên_biến [ = giá_trị_khởi_tạo];
```

- Ví dụ: `int tuoi;`
`char giớiTinh='M';`

Design by Minh An

Hằng và biến trong java



- Tên biến có thể bao gồm các ký tự Unicode và các chữ số, dấu gạch dưới (_) và dấu dollar (\$)
- Tên của một biến phải được bắt đầu bằng một ký tự, dấu dollar (\$) hoặc dấu gạch dưới (_).
- Tên biến không được trùng với từ khóa.
- Tên biến có phân biệt chữ hoa, chữ thường.

Variable name	Valid / Invalid
rollNumber	Valid
a2x5_w7t3	Valid
\$yearly_salary	Valid
_2010_tax	Valid
\$\$_	Valid
amount#Balance	Invalid and contains the illegal character #
double	Invalid and is a keyword
4short	Invalid and the first character is a digit

Design by Minh An

Hằng trong java



- Khai báo hằng số bằng cách dùng từ khóa **final** đặt trước tên kiểu dữ liệu. Giá trị của hằng phải được gán giá trị ngay thời điểm khai báo.
- Và giá trị này sau đó sẽ không thể thay đổi được.
- **Cú pháp:**

```
final kiểu_dữ_liệu  tên_hằng =giá_trị;
```

- **Ví dụ:**

```
final int H = 10;
```

Design by Minh An

Các kiểu dữ liệu cơ bản



Kiểu dữ liệu nguyên thủy

- **byte**
- **char**
- **boolean**
- **short**
- **int**
- **long**
- **float**
- **double**

Kiểu dữ liệu tham chiếu

- **Array**
- **Class**
- **Interface**

Design by Minh An

Kiểu dữ liệu nguyên thủy



Data Type	Description	Range
byte	8-bit signed integer	-128 to 127
short	16-bit signed integer	-32768 to 32767
long	64-bit signed integer	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
int	32-bit signed integer	-2,147,483,648 to 2,147,483,647
float	32-bit floating-point variable	-3.40292347E+38 to +3.40292347E+38
boolean	Stores a true or false value	true or false
char	16-bit Unicode character	0 to 65535
double	64-bit floating-point variable	-1.79769313486231570E+308 to 1.79769313486231570E+308

Design by Minh An

Định dạng dữ liệu đầu ra



Converter	Flag	Ý nghĩa
d		in số nguyên
f		in số thực
n		đầu xuống dòng
tB		tên tháng đầy đủ theo Locale
td, te		ngày trong tháng, td có số 0, te không có số 0
ty, tY		ty - năm 2 chữ số, tY - năm 4 chữ số
tl		số chỉ giờ trong hệ 12 giờ
tM		số chỉ phút 2 con số, với số 0 đứng đầu
tp		am/pm theo Locale

Design by Minh An

Định dạng dữ liệu đầu ra



Converter	Flag	Ý nghĩa
tm		số tháng 2 ký tự với số 0 đứng trước
tD		kết hợp của %tm %td %ty
	08	độ rộng bằng 8 ký tự, điền số 0 vào chỗ trống
	+	hiện thị dấu + trước số dương
	,	hiển thị ký tự phân tách nhóm 3 chữ số
	-	căn lề trái, để chứa khoảng trống bên phải nếu có
	.3	lấy 3 chữ số phần lẻ thập phân
	10.3	độ rộng là 10 ký tự, lấy 3 chữ số phần lẻ thập phân

Design by Minh An

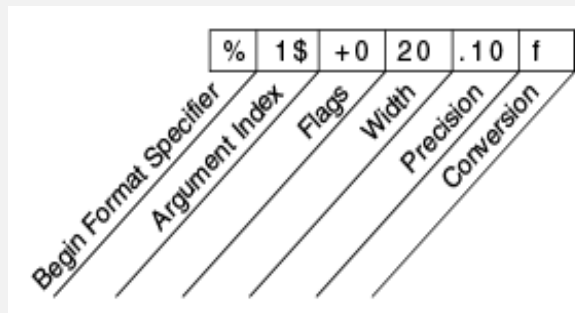
Định dạng dữ liệu đầu ra



```
public class Format {  
    public static void main(String[] args) {  
        System.out.format("%f, %1$+020.10f %n", Math.PI);  
    }  
}
```

Kết quả đầu ra:

3.141593, +00000003.1415926536



Design by Minh An

Định dạng dữ liệu đầu ra – Ví dụ



```
int num = 15;  
String str = String.format("%d %o %x", num, num, num);  
System.out.println(str);
```

```
long n = 461012;  
System.out.format("%n%d%n", n); // --> "461012"  
System.out.format("%08d%n", n); // --> "00461012"  
System.out.format("%8d%n", n); // ---> " 461012"  
System.out.format("%+8d%n", n); // --> " +461012"  
System.out.format("% ,8d%n", n); // --> " 461,012"  
System.out.format("%+,8d%n%n", n); // --> "+461,012"
```

Design by Minh An

Định dạng dữ liệu đầu ra – Ví dụ



```
double pi = Math.PI;

System.out.format("%f\n", pi);           // --> "3.141593"
System.out.format("%.3f\n", pi);         // --> "3.142"
System.out.format("%10.3f\n", pi);       // --> "      3.142"
System.out.format("%-10.3f\n", pi);      // --> "3.142      "
System.out.format(Locale.ENGLISH, "%-10.4f%n", pi);
// --> "3,1416"

Calendar c = Calendar.getInstance();
System.out.format("%tB %te, %tY%n", c, c, c); // --> "July
28, 2018"
System.out.format(Locale.CHINESE, "%tB %n", c); // --> "七月"
System.out.format(Locale.ITALIAN, "%tl:%tM %tp%n", c, c, c);
// --> "2:34 pm"
System.out.format("%tD%n", c);           // --> "07/28/18"
```

Design by Minh An

Định dạng dữ liệu đầu vào



- Lớp Scanner cho phép đọc giá trị của một vài kiểu dữ liệu.
- Để sử dụng lớp Scanner, chúng ta truyền vào đối tượng InputStream cho hàm constructor như sau:

```
Scanner input = new Scanner(System.in);
```

- Bảng thể hiện 1 số phương thức của lớp Scanner được sử dụng để nhận vào các giá trị dạng số từ người dùng.

Method	Description
nextByte()	Returns the next token as a byte value
nextInt()	Returns the next token as an int value
nextLong()	Returns the next token as a long value
nextFloat()	Returns the next token as a float value
nextDouble()	Returns the next token as a double value

Design by Minh An

Định dạng dữ liệu đầu vào



```
//tạo một đối tượng và truyền vào InputStream
Scanner input = new Scanner (System.in);

//Nhận giá trị từ người dùng
byte byteValue = input.nextByte();
int intValue = input.nextInt();
float floatValue = input.nextFloat();
long longValue = input.nextLong();
double doubleValue = input.nextDouble();

System.out.println("Values entered are: ");
System.out.println(byteValue + " " + intValue + " " +
floatValue + " " + longValue + " " + doubleValue);
```

Design by Minh An

Định dạng dữ liệu đầu vào



```
System.out.println("Enter a byte value: ");
byte byteValue = input.nextByte();
System.out.println("Enter an int value: ");
int intValue = input.nextInt();
System.out.println("Enter a float value: ");
float floatValue = input.nextFloat();
System.out.println("Enter a long value: ");
long longValue = input.nextLong();
System.out.println("Enter a double value: ");
double doubleValue = input.nextDouble();
input.nextLine();
```

Design by Minh An

Định dạng dữ liệu đầu vào



```
System.out.println("Enter a String: ");  
String str2 = s.nextLine();  
System.out.println("Values entered are: ");  
System.out.println(byteValue + " " + intValue  
+ " " + floatValue + " " + longValue + " " +  
doubleValue + " " + str2);
```

Design by Minh An

Lớp Math trong java



Tên thành phần	Công dụng
PI	Số pi
abs(X)	Giá trị tuyệt đối của X
max(X, Y)	Giá trị lớn nhất trong hai số
min(X, Y)	Giá trị nhỏ nhất trong hai số
sqrt(X)	Căn bậc hai
pow(X, Y)	X lũy thừa Y
sin(X)	sinX (X - radian)
cos(X)	cosX
tan(X)	tangX
log(X)	Loga cơ số e
log10(X)	Loga cơ số 10
...	

Design by Minh An

Cài đặt chương trình ứng dụng

1. Viết chương trình java giải quyết bài toán tính diện tích hình thang.
2. Viết chương trình java giải quyết bài toán tính chu vi, diện tích hình tròn.
3. Viết chương trình java nhập 2 số nguyên a, b và số thực x. Tính giá trị biểu thức dưới đây:

$$\frac{a}{10} + a^b - \frac{\sin \frac{\pi x}{3}}{\sqrt{(a+b)^2 + 5}} + \ln(a^2 + 1) - \frac{\log_3(4x^{2a} + 3)}{2.55}$$

Hiển thị kết quả ra màn hình.

4. Viết chương trình java giải quyết bài toán tính chu vi và diện tích của một tam giác.

Design by Minh An

Cài đặt chương trình ứng dụng (tt)

5. Viết chương trình java giải quyết bài toán tính diện tích xung quanh, diện tích toàn phần và thể tích của hình hộp chữ nhật.
6. Viết chương trình java giải quyết bài toán, nhập và hiển thị thông tin của một học sinh gồm: Họ tên, ngày, tháng, năm sinh, giới tính, điểm trung bình, xếp loại đạo đức.
7. Viết chương trình java thực hiện các yêu cầu:
 - Nhập thông tin của cán bộ bao gồm các thuộc tính: Mã cán bộ, họ và tên, ngày sinh, số ngày làm việc trong tháng, số giờ làm thêm.
 - Tính lương của cán bộ theo công thức lương = Số ngày làm việc trong tháng * 350.000 + số giờ làm thêm * 45000.
 - Hiển thị thông tin của cán bộ ra màn hình cùng với lương cán bộ.

Design by Minh An

Cài đặt chương trình ứng dụng (tt)

9. Viết chương trình java thực hiện các yêu cầu:

- Nhập thông tin của sinh viên bao gồm các thuộc tính: Mã sinh viên, họ tên, ngày sinh, tên của 5 môn học và số tín chỉ tương ứng của các môn học:
- Tính điểm trung bình của các môn học theo hệ số tín chỉ.
- Hiển thị thông tin của sinh viên ra màn hình.

10. Viết chương trình java thực hiện các yêu cầu:

- Nhập số nguyên dương n có 4 chữ số:
- Hiển thị ra màn hình các chữ số của n và tổng của chúng.