

ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG CAO ĐẲNG BÁCH KHOA NAM SÀI GÒN



GIÁO TRÌNH

MÔN HỌC/MÔ ĐUN: LẬP TRÌNH CĂN BẢN

NGÀNH/ NGHỀ: TIN HỌC ỨNG DỤNG

TRÌNH ĐỘ: CAO ĐẲNG

Ban hành kèm theo Quyết định số: 451/QĐ-NSG, ngày 08 tháng 08 năm 2023
của Hiệu trưởng Trường Cao Đẳng Bách Khoa Nam Sài Gòn

Tp.Hồ Chí Minh, năm 2023

TUYÊN BỐ BẢN QUYỀN

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

LỜI GIỚI THIỆU

Môn học này thuộc khối kiến thức chuyên môn nghề trong chương trình đào tạo ngành tin học ứng dụng hệ cao đẳng. Lập trình căn bản là môn học bắt buộc trong chương trình ngành Tin học ứng dụng trình độ cao đẳng.

Nội dung giáo trình gồm 5 bài:

- Bài 1: Giới thiệu ngôn ngữ lập trình
- Bài 2: Kiểu, toán tử, lệnh IF, SWITCH
- Bài 3: Mảng và lệnh lặp
- Bài 4: Lớp và đối tượng
- Bài 5: Kế thừa

Trong quá trình biên soạn chúng tôi đã kết hợp kinh nghiệm giảng dạy trong nhiều năm của nhóm tác giả với mong muốn có thể giúp cho học sinh – sinh viên dễ dàng nắm bắt được nội dung của môn học. Mặc dù, rất cố gắng trong quá trình biên soạn, nhưng chắc chắn không thể tránh khỏi những thiếu sót, vì vậy, nhóm tác giả chúng tôi rất mong nhận được ý kiến đóng góp của bạn đọc để giáo trình ngày một hoàn thiện hơn.

Tp.HCM , ngày 08 tháng 08 năm 2023

Tham gia biên soạn

1. Chủ biên: ThS Bùi Thị Mai Châu

2.

3.

.....

MỤC LỤC

LỜI GIỚI THIỆU.....	3
Bài 1: Giới thiệu ngôn ngữ lập trình	6
1.1. Giới thiệu ngôn ngữ	6
1.2. Biến, toán tử, hàm toán học và lệnh nhập xuất	8
Bài tập:	18
Bài 2: Cấu trúc IF, SWITCH.....	22
2.1. Lệnh IF	22
2.2. Lệnh SWITCH	24
Bài tập:	24
Bài 3: Mảng và lệnh lặp	27
3.1. Lệnh lặp.....	27
3.2. Mảng.....	30
Bài tập:	42
Bài 4: Lớp và đối tượng	46
4.1. Lớp và đối tượng.....	46
4.2. Phương thức	48
4.3. Package	52
Bài tập:	57
Bài 5: Kế thừa	60
5.1. Kế thừa	60
5.2. Lớp và phương thức trừu tượng.....	72
Bài tập:	92
TÀI LIỆU THAM KHẢO:.....	97

GIÁO TRÌNH MÔN HỌC/MÔ ĐUN

Tên môn học/mô đun: LẬP TRÌNH WEB CƠ BẢN

Tên môn học: Lập trình căn bản

Mã số môn học: MH09

Thời gian môn học: 60 giờ; (Lý thuyết: 30 giờ, Thực hành, thí nghiệm, bài tập, thảo luận: 28 giờ, Kiểm tra: 02 giờ)

Vị trí, tính chất của môn học:

- Vị trí: Môn Lập trình căn bản là môn chuyên ngành, trong chương trình đào tạo hệ cao đẳng Tin học ứng dụng.
- Tính chất: Môn Lập trình căn bản là môn học cung cấp kiến thức nền tảng về ngôn ngữ lập trình (từ định nghĩa đến kiểu dữ liệu, biến, câu lệnh, mảng, chuỗi ... cấu trúc điều khiển, phương thức, lớp....), có khả năng tiếp tục phát triển ngôn ngữ lập trình nâng cao.

Mục tiêu môn học:

1. Về kiến thức:

- Trình bày được các kiến thức cơ bản về ngôn ngữ lập trình;
- Phân biệt các biến, cấu trúc dữ liệu, toán tử, biên dịch và thực thi chương trình, câu lệnh, điều khiển;
- Áp dụng các kiến thức cơ bản về lập trình trong việc giải quyết các bài toán đơn giản.

2. Về kỹ năng:

- Nhận dạng được các lệnh cơ bản;
- Xây dựng được những chương trình ứng dụng đơn giản.

3. Về năng lực tự chủ và trách nhiệm:

- Rèn luyện lòng yêu nghề, tư thế tác phong công nghiệp, tính kiên trì, sáng tạo trong công việc.

Nội dung của môn học/mô đun:

Bài 1: Giới thiệu ngôn ngữ lập trình

1. Mục tiêu:

- Trình bày được khái niệm về biến, toán tử và các hàm toán học;
- Áp dụng để viết mã và thi hành chương trình đơn giản.

2. Nội dung:

1.1. Giới thiệu ngôn ngữ

Java là ngôn ngữ lập trình hướng đối tượng (tựa C++) do Sun Microsystem đưa ra vào giữa thập niên 90.

Chương trình viết bằng ngôn ngữ lập trình java có thể chạy trên bất kỳ hệ thống nào có cài máy ảo java (Java Virtual Machine).

Ngôn ngữ lập trình Java do James Gosling và các công sự của Công ty Sun Microsystem phát triển.

Đầu thập niên 90, Sun Microsystem tập hợp các nhà nghiên cứu thành lập nên nhóm đặt tên là Green Team. Nhóm Green Team có trách nhiệm xây dựng công nghệ mới cho ngành điện tử tiêu dùng. Để giải quyết vấn đề này nhóm nghiên cứu phát triển đã xây dựng một ngôn ngữ lập trình mới đặt tên là Oak tương tự như C++ nhưng loại bỏ một số tính năng nguy hiểm của C++ và có khả năng chạy trên nhiều nền phần cứng khác nhau. Cùng lúc đó world wide web bắt đầu phát triển và Sun đã thấy được tiềm năng của ngôn ngữ Oak nên đã đầu tư cải tiến và phát triển. Sau đó không lâu ngôn ngữ mới với tên gọi là Java ra đời và được giới thiệu năm 1995.

Java là tên gọi của một hòn đảo ở Indonexia, 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 café trong nhiều sản phẩm phần mềm, công cụ lập trình Java của Sun cũng như một số hãng phần mềm khác đưa ra.

Một số đặc điểm nổi bật của ngôn ngữ lập trình Java

Máy ảo Java (JVM - Java Virtual Machine)

Tất cả các chương trình muốn thực thi được thì phải được biên dịch ra mã máy. Mã máy của từng kiến trúc CPU của mỗi máy tính là khác nhau (tập lệnh mã máy của CPU Intel, CPU Solarix, CPU Macintosh ... là khác nhau), vì vậy trước đây một chương trình sau khi được biên dịch xong chỉ có thể chạy được trên một kiến trúc CPU cụ thể nào đó. Đối với CPU Intel chúng ta có thể chạy các hệ điều hành như Microsoft Windows, Unix, Linux, OS/2, ... Chương trình thực thi được trên Windows được biên dịch dưới dạng file có đuôi .EXE còn trên Linux thì được biên dịch dưới dạng file có đuôi .ELF, vì vậy trước đây một chương trình chạy được trên Windows muốn chạy được trên hệ điều hành khác như Linux chẳng hạn thì phải chỉnh sửa và biên dịch lại. Ngôn ngữ lập trình Java ra đời, nhờ vào máy ảo Java mà khó khăn nêu trên đã được khắc phục. Một chương trình viết bằng ngôn ngữ lập trình Java sẽ được biên dịch ra mã của máy ảo java (mã java bytecode). Sau đó máy ảo Java chịu trách nhiệm chuyển mã java bytecode thành mã máy tương ứng. Sun Microsystems chịu trách nhiệm phát triển các máy ảo Java chạy trên các hệ điều hành trên các kiến trúc CPU khác nhau.

Thông dịch:

Java là một ngôn ngữ lập trình vừa biên dịch vừa thông dịch. Chương trình nguồn viết bằng ngôn ngữ lập trình Java có đuôi *.java đầu tiên được biên dịch thành tập tin có đuôi *.class và sau đó sẽ được trình thông dịch thông dịch thành mã máy.

Độc lập nền:

Một chương trình viết bằng ngôn ngữ Java có thể chạy trên nhiều máy tính có hệ điều hành khác nhau (Windows, Unix, Linux, ...) miễn sao ở đó có cài đặt máy ảo java (Java Virtual Machine). Viết một lần chạy mọi nơi (write once run anywhere).
Hướng đối tượng:

Hướng đối tượng trong Java tương tự như C++ nhưng Java là một ngôn ngữ lập trình hướng đối tượng hoàn toàn. Tất cả mọi thứ đề cập đến trong Java đều liên quan đến các đối tượng được định nghĩa trước, thậm chí hàm chính của một chương

trình viết bằng Java (đó là hàm main) cũng phải đặt bên trong một lớp. Hướng đối tượng trong Java không có tính đa kế thừa (multi inheritance) như trong C++ mà thay vào đó Java đưa ra khái niệm interface để hỗ trợ tính đa kế thừa. Vấn đề này sẽ được bàn chi tiết trong chương 3.

Đa nhiệm - đa luồng (MultiTasking - Multithreading):

Java hỗ trợ lập trình đa nhiệm, đa luồng cho phép nhiều tiến trình, tiểu trình có thể chạy song song cùng một thời điểm và tương tác với nhau. Khả chuyển (portable):

Chương trình ứng dụng viết bằng ngôn ngữ Java chỉ cần chạy được trên máy ảo Java là có thể chạy được trên bất kỳ máy tính, hệ điều hành nào có máy ảo Java. “Viết một lần, chạy mọi nơi” (Write Once, Run Anywhere).

Hỗ trợ mạnh cho việc phát triển ứng dụng:

Công nghệ Java phát triển mạnh mẽ nhờ vào “đại gia Sun Microsystem” cung cấp nhiều công cụ, thư viện lập trình phong phú hỗ trợ cho việc phát triển nhiều loại hình ứng dụng khác nhau cụ thể như: J2SE (Java 2 Standard Edition) hỗ trợ phát triển những ứng dụng đơn, ứng dụng client-server; J2EE (Java 2 Enterprise Edition) hỗ trợ phát triển các ứng dụng thương mại, J2ME (Java 2 Micro Edition) hỗ trợ phát triển các ứng dụng trên các thiết bị di động, không dây, ...

1.2. Biến, toán tử, hàm toán học và lệnh nhập xuất

1.2.1. Biến, quy tắc đặt tên.

- Biến là vùng nhớ dùng để lưu trữ các giá trị của chương trình. Mỗi biến gắn liền với một kiểu dữ liệu và một định danh duy nhất gọi là tên biến.
- Tên biến thông thường là một chuỗi các ký tự (Unicode), ký số. o Tên biến phải bắt đầu bằng một chữ cái, một dấu gạch dưới hay dấu dollar.
- Tên biến không được trùng với các từ khóa (xem phụ lục các từ khóa trong java).
- Tên biến không có khoảng trắng ở giữa tên.

- Trong java, biến có thể được khai báo ở bất kỳ nơi đâu trong chương trình.

Cách khai báo

```
<kiểu_dữ_liệu> <tên_biến>;
```

```
<kiểu_dữ_liệu> <tên_biến> = <giá_trị>;
```

Gán giá trị cho biến

```
<tên_biến> = <giá_trị>;
```

Biến công cộng (toàn cục): là biến có thể truy xuất ở khắp nơi trong chương trình, thường được khai báo dùng từ khóa public, hoặc đặt chúng trong một class.

Biến cục bộ: là biến chỉ có thể truy xuất trong khối lệnh nó khai báo.

Lưu ý: Trong ngôn ngữ lập trình java có phân biệt chữ in hoa và in thường. Vì vậy chúng ta cần lưu ý khi đặt tên cho các đối tượng dữ liệu cũng như các xử lý trong chương trình.

Ví dụ:

```
import java.lang.*; import java.io.*;
```

```
class VariableDemo
```

```
{
```

```
    static int x, y;
```

```
    public static void main(String[] args)
```

```
    {
```

```
        x    = 10;
```

```
        y    = 20;
```

```
        int z = x+y;
```

```
        System.out.println("x = " + x);
```

```
        System.out.println("y = " + y);
```

```

        System.out.println("z = x + y =" + z);

System.out.println("So nho hon la so:" +

Math.min(x, y));

        char c = 80;

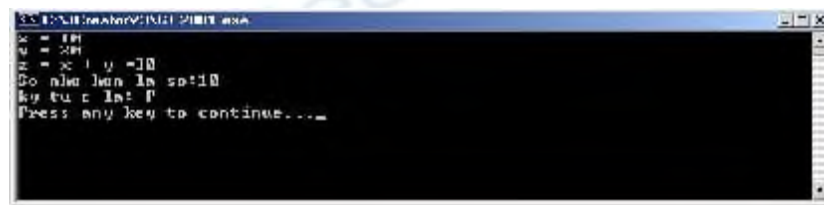
        System.out.println("ky tu c la: " + c);

    }

}

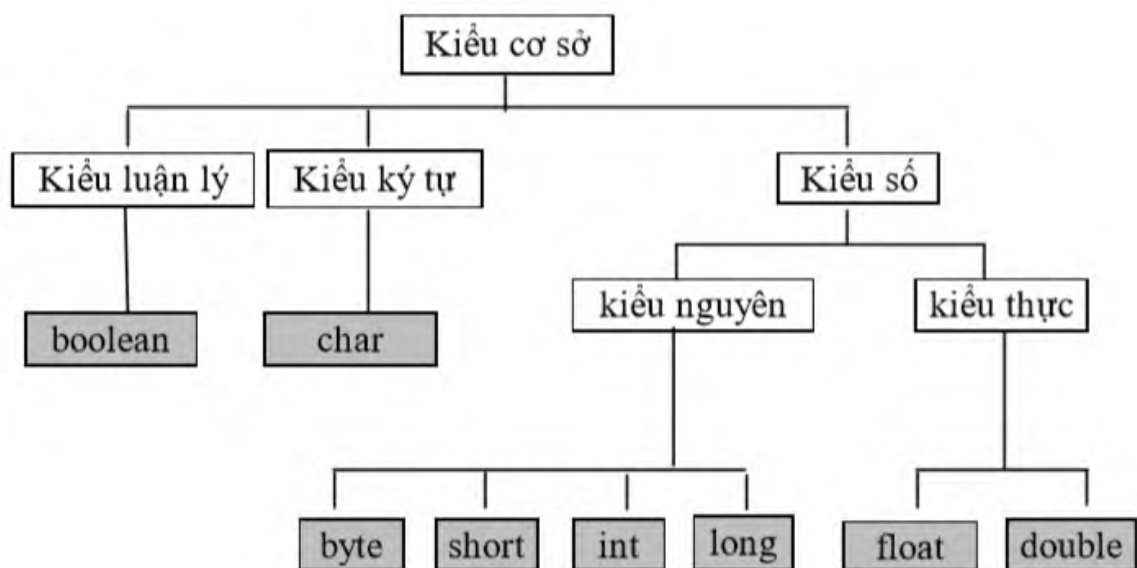
```

Kết quả chương trình



1.2.2. Các kiểu dữ liệu cơ sở

Ngôn ngữ lập trình java có 8 kiểu dữ liệu cơ sở: byte, short, int, long, float, double, boolean và char



Kiểu		Giá trị min	Giá trị max	
------	--	-------------	-------------	--

	Kích thước (bytes)			Giá trị mặc định
byte	1	-256	255	0
short	2	-32768	32767	0
int	4	-2^{31}	$2^{31} - 1$	0
long	8	-2^{63}	$2^{63} - 1$	0L
float	4			0.0f
double	8			0.0d

Kiểu số nguyên:

- Java cung cấp 4 kiểu số nguyên khác nhau là: byte, short, int, long. Kích thước, giá trị nhỏ nhất, lớn nhất, cũng như giá trị mặc định của các kiểu dữ liệu số nguyên được mô tả chi tiết trong bảng trên.
- Kiểu mặc định của các số nguyên là kiểu int.
- Các số nguyên kiểu byte và short rất ít khi được dùng.
- Trong java không có kiểu số nguyên không dấu như trong ngôn ngữ C/C++.

Khai báo và khởi tạo giá trị cho các biến kiểu nguyên:

```
int x = 0;
```

```
long y = 100;
```

Một số lưu ý đối với các phép toán trên số nguyên:

- Nếu hai toán hạng kiểu long thì kết quả là kiểu long. Một trong hai toán hạng không phải kiểu long sẽ được chuyển thành kiểu long trước khi thực hiện phép toán.

- Nếu hai toán hạng đầu không phải kiểu long thì phép tính sẽ thực hiện với kiểu int.
- Các toán hạng kiểu byte hay short sẽ được chuyển sang kiểu int trước khi thực hiện phép toán.
- Trong java không thể chuyển biến kiểu int và kiểu boolean như trong ngôn ngữ C/C++.

Ví dụ: có đoạn chương trình như sau `boolean b = false; if (b == 0)`

```
{  
  
    System.out.println("Xin chào");  
  
}
```

Lúc biên dịch đoạn chương trình trên trình dịch sẽ báo lỗi: không được phép so sánh biến kiểu boolean với một giá trị kiểu int.

Kiểu dấu chấm động:

Đối với kiểu dấu chấm động hay kiểu thực, java hỗ trợ hai kiểu dữ liệu là float và double.

Kiểu float có kích thước 4 byte và giá trị mặc định là 0.0f

Kiểu double có kích thước 8 byte và giá trị mặc định là 0.0d

Số kiểu dấu chấm động không có giá trị nhỏ nhất cũng không có giá trị lớn nhất. Chúng có thể nhận các giá trị:

- Số âm
- Số dương
- Vô cực âm
- Vô cực dương

Khai báo và khởi tạo giá trị cho các biến kiểu dấu chấm động: `float x = 100.0/7;` `double y = 1.56E6;`

Một số lưu ý đối với các phép toán trên số dấu chấm động:

- Nếu mỗi toán hạng đều có kiểu dấu chấm động thì phép toán chuyển thành phép toán dấu chấm động.
- Nếu có một toán hạng là double thì các toán hạng còn lại sẽ được chuyển thành kiểu double trước khi thực hiện phép toán.
- Biến kiểu float và double có thể ép chuyển sang kiểu dữ liệu khác trừ kiểu boolean.

Kiểu ký tự (char):

Kiểu ký tự trong ngôn ngữ lập trình java có kích thước là 2 bytes và chỉ dùng để biểu diễn các ký tự trong bộ mã Unicode. Như vậy kiểu char trong java có thể biểu diễn tất cả $2^{16} = 65536$ ký tự khác nhau.

Giá trị mặc định cho một biến kiểu char là null.

Kiểu luận lý (boolean):

- Kiểu boolean chỉ nhận 1 trong 2 giá trị: true hoặc false.
- Trong java kiểu boolean không thể chuyển thành kiểu nguyên và ngược lại.
- Giá trị mặc định của kiểu boolean là false.

Hằng:

- Hằng là một giá trị bất biến trong chương trình
- Tên hằng được đặt theo qui ước giống như tên biến.
- Hằng số nguyên: trường hợp giá trị hằng ở dạng long ta thêm vào cuối chuỗi số chữ "l" hay "L". (ví dụ: 1L)
- Hằng số thực: trường hợp giá trị hằng có kiểu float ta thêm tiếp vĩ ngữ "f" hay "F", còn kiểu số double thì ta thêm tiếp vĩ ngữ "d" hay "D".
- Hằng Boolean: java có 2 hằng boolean là true, false.
- Hằng ký tự: là một ký tự đơn nằm giữa 2 dấu ngoặc đơn.

- Ví dụ: ‘a’: hằng ký tự a
- Một số hằng ký tự đặc biệt

Ký tự	Ý nghĩa
\b	Xóa lùi (BackSpace)
\t	Tab
\n	Xuống hàng
\r	Dấu enter
\”	Nháy kép
\’	Nháy đơn
\\	Số ngược
\f	Đẩy trang
\uxxxx	Ký tự unicode

- Hằng chuỗi: là tập hợp các ký tự được đặt giữa hai dấu nháy kép “”. Một hằng chuỗi không có ký tự nào là một hằng chuỗi rỗng.

- Ví dụ: “Hello Wolrd”
- Lưu ý: Hằng chuỗi không phải là một kiểu dữ liệu cơ sở nhưng vẫn được khai báo và sử dụng trong các chương trình.

Lệnh, khối lệnh trong java :

Giống như trong ngôn ngữ C, các câu lệnh trong java kết thúc bằng một dấu chấm phẩy (;).

Một khối lệnh là đoạn chương trình gồm hai lệnh trở lên và được bắt đầu bằng dấu mở ngoặc nhọn ({) và kết thúc bằng dấu đóng ngoặc nhọn (}).

Bên trong một khối lệnh có thể chứa một hay nhiều lệnh hoặc chứa các khối lệnh khác.

```
{ // khối 1

    { // khối 2

        lệnh 2.1

        lệnh 2.2

        ...

    } // kết thúc khối lệnh 2

    lệnh 1.1    lệnh 1.2

    ...

} // kết thúc khối lệnh 1

{ // bắt đầu khối lệnh 3

    // Các lệnh thuộc khối lệnh 3

    // ...

} // kết thúc khối lệnh 3
```

1.2.3. Toán tử và biểu thức.

Toán tử số học:

Toán tử	Ý nghĩa
+	Cộng
-	Trừ
*	Nhân
/	Chia nguyên
%	Chia dư
++	Tăng 1

--	Giảm 1
----	--------

Toán tử trên bit:

Toán tử	Ý nghĩa
&	AND
	OR
^	XOR
<<	Dịch trái
>>	Dịch phải
>>>	Dịch phải và điền 0 vào bit trống
~	Bù bit

Toán tử quan hệ & logic:

Toán tử	Ý nghĩa
==	So sánh bằng
!=	So sánh khác
>	So sánh lớn hơn
<	So sánh nhỏ hơn
>=	So sánh lớn hơn hay bằng
<=	So sánh nhỏ hơn hay bằng
	OR (biểu thức logic)
&&	AND (biểu thức logic)
!	NOT (biểu thức logic)

Toán tử ép kiểu:

- Ép kiểu rộng (widening conversion): từ kiểu nhỏ sang kiểu lớn (không mất mát thông tin)
- Ép kiểu hẹp (narrow conversion): từ kiểu lớn sang kiểu nhỏ (có khả năng mất mát thông tin) <tên biến> = (kiểu_dữ_liệu) <tên biến>; Ví dụ:

```
float fNum = 2.2;
```

```
int iCount = (int) fNum; // (iCount = 2)
```

Toán tử điều kiện:

Cú pháp: <điều kiện> ? <biểu thức 1> : <biểu thức 2> Nếu điều kiện đúng thì có giá trị, hay thực hiện <biểu thức 1>, còn ngược lại là <biểu thức 2>. <điều kiện>: là một biểu thức logic

<biểu thức 1>, <biểu thức 2>: có thể là hai giá trị, hai biểu thức hoặc hai hành động.

Ví dụ:

```
int x = 10; int y = 20; int Z = (x < y) ? 30 : 40;
```

// Kết quả z = 30 do biểu thức (x < y) là đúng.

Thứ tự ưu tiên:

Thứ tự ưu tiên tính từ trái qua phải và từ trên xuống dưới

Cao nhất			
()	[]	.	
++	--	~	!
*	/	%	
+	-		
>>	>>> (dịch phải và điền 0 vào bit trống)	<<	

>	>=	<	<=
==	!=		
&			
^			
&&			
?:			
=	<toántử>=		
Thấp nhất			

1.2.4. Lệnh xuất ra màn hình và nhập từ bàn phím.

Bài tập:

Câu 1. Kích thước của 1 Char là bao nhiêu?

- a. 4 bit b. 7 bit c. 8 bit d. 16 bit

Câu 2. 1 class không thể được khai báo ở chế độ

- a. Static b. Private c. Default

Câu 3. Kết quả sau khi biên dịch mã int a = 3.5

- a. Compilation error
b. Runtime error
c. 3.5
d. 3

Câu 4. Kết quả khi biên dịch mã int a1 = 5; double a2 = (float)a1

- a. Compilation error

- b. Runtime error
- c. No error

Câu 5. Kết quả sau khi biên dịch mã `int a = 9/0`

- a. Compilation error: Phép tính chia phải nằm trong cú pháp try
- b. Compilation error: `DivideByZeroException`
- c. Runtime Exception
- d. No Error: kết quả là NaN

Câu 6. Kết quả sau khi biên dịch mã: `float a = 9/0`

- a. Compilation error: Phép tính chia phải nằm trong cú pháp try
- b. Compilation error: `DivideByZeroException`
- c. Runtime Exception
- d. No Error: Kết quả là NaN

Câu 7: Đây là câu SAI về ngôn ngữ Java?

- a. Ngôn ngữ Java có phân biệt chữ hoa – chữ thường
- b. Java là ngôn ngữ lập trình hướng đối tượng
- c. Dấu chấm phẩy được sử dụng để kết thúc lệnh trong java
- d. Chương trình viết bằng Java chỉ có thể chạy trên hệ điều hành win

Câu 8: Đây không phải là một kiểu dữ liệu nguyên thủy trong Java?

- a. double b. int c. long d. long float

Câu 9: Trong câu lệnh sau: `public static void main(String[] args)` thì phần tử `args[0]` chứa giá trị gì?

- a. Tên của chương trình
- b. Số lượng tham số

- c. Tham số đầu tiên của danh sách tham số
- d. Không câu nào đúng

Câu 10: Phương thức next() của lớp Scanner dùng để làm gì?

- a. Nhập một số nguyên
- b. Nhập một ký tự
- c. Nhập một chuỗi
- d. Không có phương thức này

Câu 11: Muốn chạy được chương trình java, chỉ cần cài phần mềm nào sau đây?

- a. Netbeans
- b. Eclipse
- c. JDK
- d. Java Platform

Câu 12: Gói nào trong java chứa lớp Scanner dùng để nhập dữ liệu từ bàn phím?

- a. java.net
- b. java.io
- c. java.util
- d. java.awt

Câu 13: Phương thức nextLine() thuộc lớp nào ?

- A. String
- B. Scanner
- C. Integer
- D. System

Câu 14: Tên đầu tiên của Java là gì?

- A. Java
- B. Oak
- C. Cafe
- D. James golings

Câu 15: Đâu KHÔNG phải là thành phần trong cấu trúc của lớp trong java.

- A. Tên lớp
- B. Thuộc tính
- C. Phương thức
- D. Biến

Câu 16: Đọc đoạn mã lệnh sau. Sau khi thực thi chương trình sẽ in ra kết quả gì ?

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int x = -1;  
  
        String y = x + 3;  
  
        System.out.println("x = " + x + "y = " + y);  
    }  
}
```