

Chương 2

Ngôn Ngữ Java



Trường Đại học Công nghệ Sài Gòn
Khoa Công nghệ Thông tin

Nội dung

- Tổng quan
- Cấu trúc chương trình Java
- Biến & Hằng
- Kiểu dữ liệu
- Toán tử, biểu thức
- Các cấu trúc điều khiển (chọn, rẽ nhánh, lặp)
- Lớp wrapper.
- Phương thức và cách sử dụng
- Một số lớp cơ bản
- Input/output

Java?

- Java là ngôn ngữ hướng đối tượng
- Java trong sáng: phát triển từ c, c++-> C#, python, scalar
- Java không phụ thuộc nền
- Java hỗ trợ phát triển app : web, mobile, Windows form, console, nhúng, phân bố, điều khiển
- Văn phạm Java giống c#
- Khó khăn: Công cụ đa dạng (intelij, myeclipse) (eclipse, apache netbean), các phiên bản java (jdk 8.→ jdk 21)

Cấu trúc chương trình Java

- Cấu trúc khối → **class**
- Khai báo biến
 - Toàn cục-→ **static**
 - Cục bộ
 - Kiểu dữ liệu
 - **Primitive**: nguyên thủy, cơ sở: int, double, char, boolean
 - Reference (Tham chiếu): đối tượng: String, Array, Object
 - Khai báo hàm và thủ tục
 - Tham số truyền
 - Tham trị
 - Tham khảo
 - Trả kết quả:
 - void
 - Kiểu dữ liệu trả về
 - Phương thức **main**

Biến

- Tên biến: *Phân biệt chữ hoa chữ thường (giống C)*
- Trong java, biến có thể được khai báo ở bất kỳ nơi đâu trong chương trình.
- Biến toàn cục:
 - Khai báo dùng từ khóa **public**, hoặc đặt chúng trong một class
- Biến cục bộ:
 - *Bắt buộc phải khởi tạo giá trị trước khi sử dụng.*
- Khai báo biến:
 - Cách khai báo biến:
 - `<kiểu_dữ_liệu> <tên_biến>;`
 - `<kiểu_dữ_liệu> <tên_biến> = <giá_trị>;`
 - Gán giá trị cho biến:
 - `<tên_biến> = <giá_trị>;`

Biến

- Lưu ý:

- Trong java nếu lúc khai báo không khởi tạo giá trị cho biến thì nó sẽ nhận 1 giá trị mặc định. Mỗi kiểu dữ liệu có 1 giá trị mặc định khác nhau.
 - 0 nếu kiểu dữ liệu là kiểu số
 - '\0' nếu kiểu dữ liệu là ký tự
 - false nếu kiểu dữ liệu là boolean
 - null nếu kiểu dữ liệu là class
- Nên chủ động khởi tạo giá trị ban đầu!!!!

Hằng

- Hằng?
 - Tên đặt theo qui ước như tên biến
 - Khai báo dùng từ khóa **final**
- Ví dụ:
 - `final int x = 10; // khai báo hằng số nguyên x = 10`
 - `final long y = 20; // khai báo hằng số long y = 20`
- Hằng ký tự: đặt giữa cặp nháy đơn ‘ ’
- Hằng chuỗi: là một dãy ký tự đặt giữa cặp nháy đôi “ ”

Hằng kí tự đặc biệt

Ký tự	Ý nghĩa
\b	Xóa lùi (BackSpace)
\t	Tab
\n	Xuống hàng
\r	Dấu enter
\"	Nháy kép
'	Nháy đơn
\\	\
\f	đẩy trang
\uxxxx	Ký tự unicode

Kiểu dữ liệu

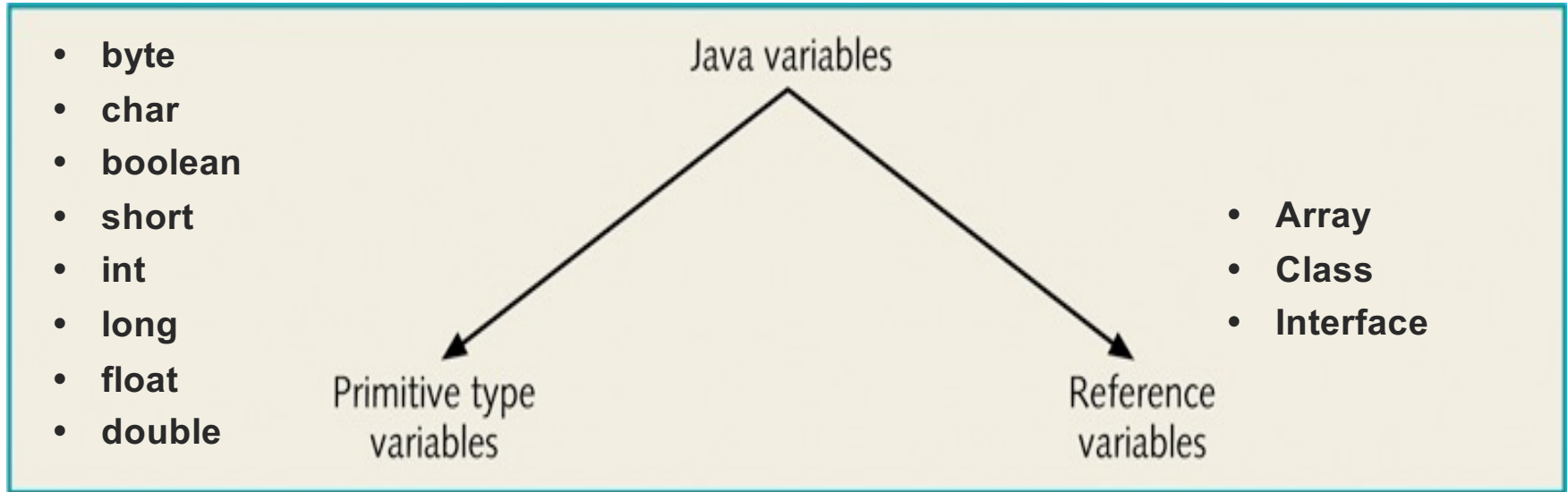


Figure 3-6 Java variables

Kiểu dữ liệu

```
int x;           //primitive type
String str;      //Reference type
x = 45;          //gán trị
str = "Java Programming"; //gán trị
```

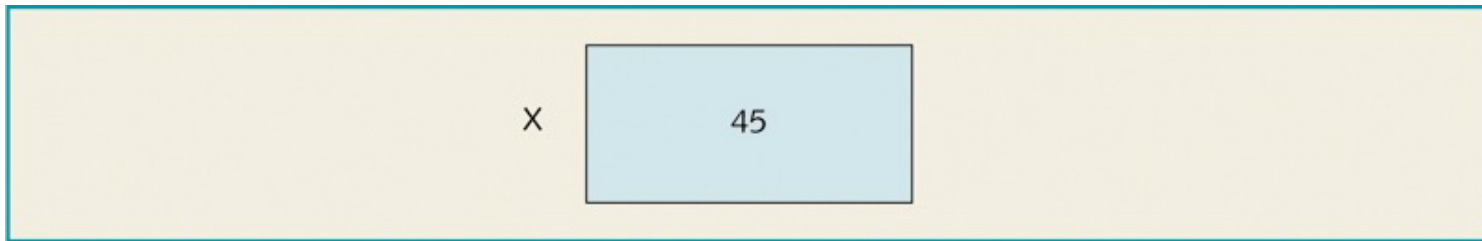


Figure 3-1 Variable `x` and its data

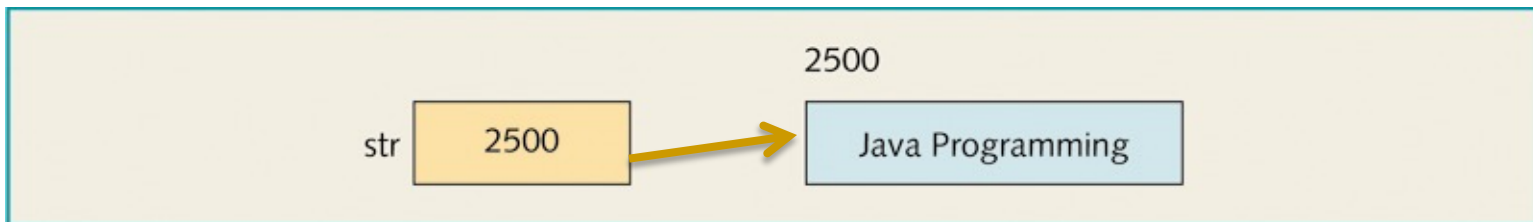
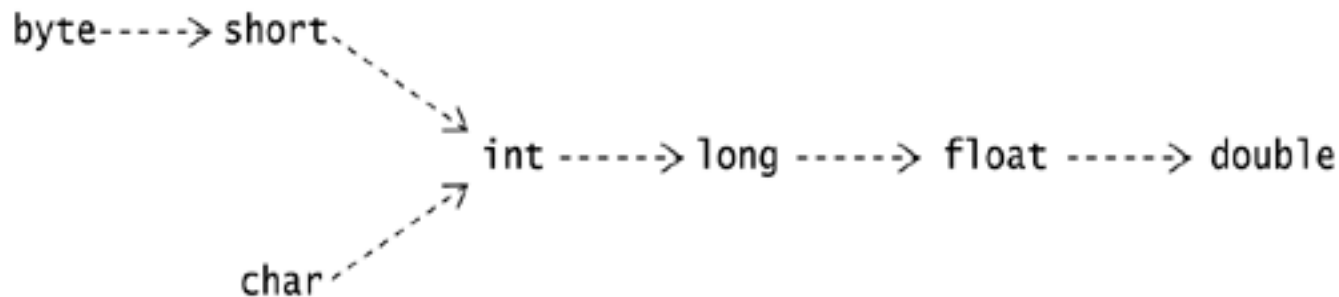


Figure 3-2 Variable `str` and the data it points to

Chuyển đổi kiểu dữ liệu

- Khi có sự **không tương thích** về kiểu dữ liệu (gán, tính toán biểu thức, truyền đối số gọi phương thức)
 - Chuyển kiểu hẹp (lớn → nhỏ): **cần ép kiểu**
 - **<tên biến 2> = (kiểu dữ liệu) <tên biến 1>;**
 - Chuyển kiểu rộng (nhỏ → lớn): **tự động chuyển**



Kiểu dữ liệu tham chiếu

- Kiểu mảng:
 - Khai báo:
 - `<kiểu dữ liệu>[] <tên mảng>; //mảng 1 chiều`
 - `<kiểu dữ liệu> <tên mảng>[]; //mảng 1 chiều`
 - `<kiểu dữ liệu>[][] <tên mảng>; //mảng 2 chiều`
 - `<kiểu dữ liệu> <tên mảng>[][]; //mảng 2 chiều`
 - Khởi tạo mảng:
 - `int arrInt[] = {1, 2, 3};`
 - `char arrChar[] = {'a', 'b', 'c'};`
 - `String arrString[] = {"ABC", "EFG", "GHI"};`
 - Cấp phát & truy cập mảng:
 - `int arrInt = new int[100];`
 - `int arrInt[100];` // Khai báo này trong Java sẽ báo lỗi.
 - Chỉ số mảng n phần tử: từ 0 đến $n-1$
 - **Chú ý: Mảng trong java được xem là một đối tượng**

Kiểu dữ liệu tham chiếu

- Kiểu đối tượng:
 - Khai báo đối tượng
 - *<Kiểu đối tượng>* *<biến Đ T>;*
 - Khởi tạo đối tượng: từ khóa **new**
 - *<Kiểu đối tượng>* *<biến Đ T> = new <Kiểu đối tượng>;*
 - Truy xuất thành phần đối tượng:
 - *<biến Đ T>.<thuộc tính>*
 - *<biến Đ T>.<phương thức>([tham số])*

Một số toán tử

■ Toán tử số học:

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

Một số toán tử

■ 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)

Một số toán tử

■ Toán tử gán:

Toán tử	Ví dụ	Ý nghĩa
=	$a = b$	gán $a = b$
+=	$a += 5$	$a = a + 5$
-=	$b -= 10$	$b = b - 10$
*=	$c *= 3$	$c = c * 3$
/=	$d /= 2$	$d = d / 2$
%=	$e \% = 4$	$e = e \% 4$

Một số toán tử

- Toán tử điều kiện:
 - $\langle \text{điều kiện} \rangle ? \langle \text{biểu thức 1} \rangle : \langle \text{biểu thức 2} \rangle$
 - `int x = 10; int y = 20;`
 - `int Z = (x < y) ? 30 : 40;`
 - *// Kết quả z = 30 do biểu thức (x < y) là đúng.*

DEMO

Gói `java.lang`

- Gói cơ bản của ngôn ngữ java
- Chứa các class cơ bản đóng vai trò trung tâm đối với các tác vụ java
- Các class cơ sở: **Object (lớp cha của các lớp trong java), Class (đại diện cho một lớp trong java)**
- Nếu muốn xem các kiểu dữ liệu cơ bản như một đối tượng, sử dụng **class Wrapper**: Boolean, Integer, Double, Character...
- **class Void**: không thể khởi tạo bằng từ khóa new, dùng để biểu diễn cho kiểu void
- **class Math** chứa các phương thức xử lý toán học
 - Math.tên_hàm(tham_số): Math.abs(-5) → 5
- class Loader, Process, System, Runtime ... cung cấp các tác vụ mức hệ thống

Cấu trúc điều khiển

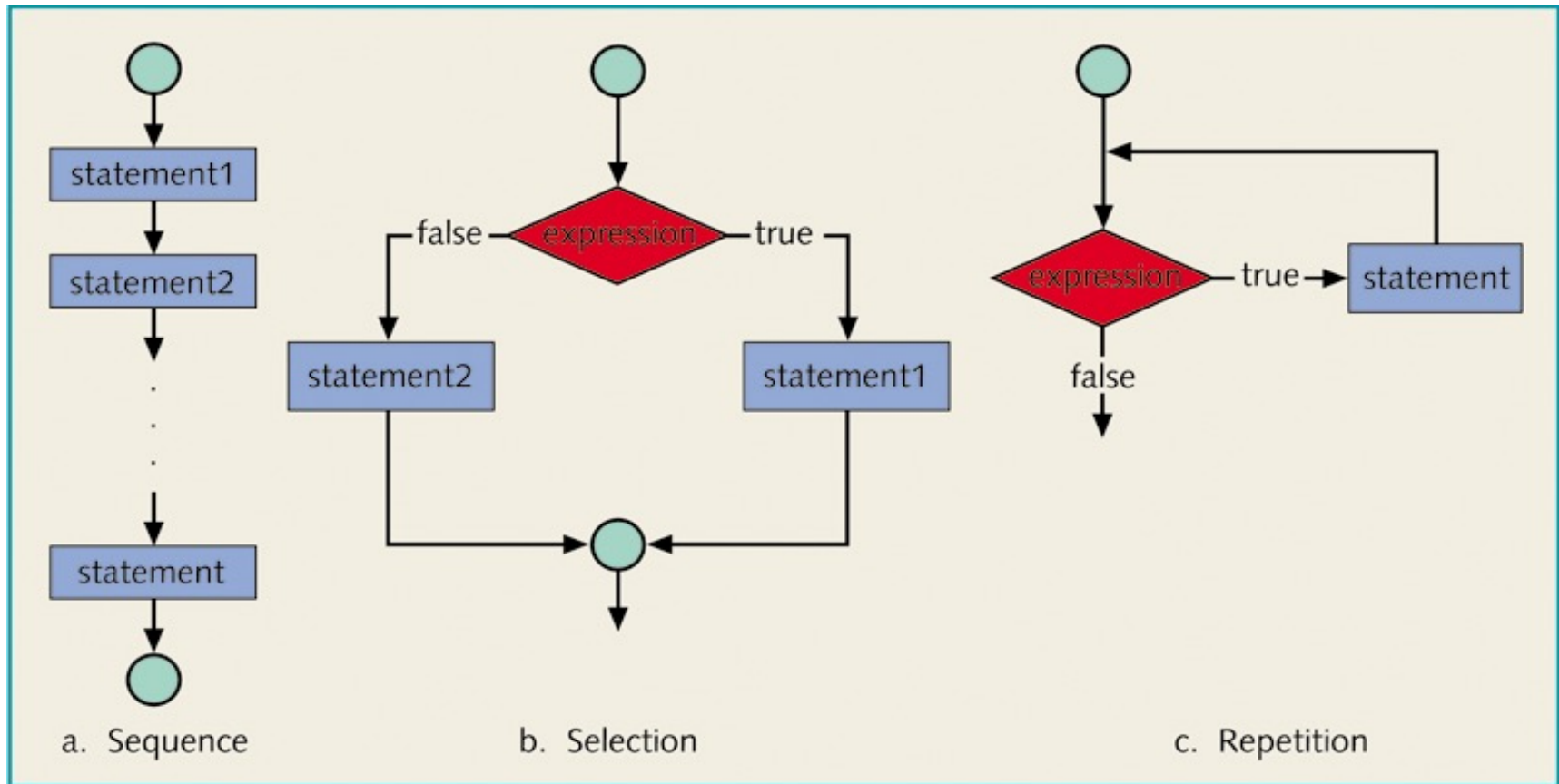


Figure 4-1 Flow of execution

Cấu trúc điều kiện

■ Cú pháp

if (condition)

```
{  
  action1 statements;  
}  
else  
{  
  action2 statements;  
}
```

■ Cú pháp

switch (expression)

```
{  
    case 'value1': action1 statement(s);  
                    break;  
    case 'value2': action2 statement(s);  
                    break;  
    :  
    :  
    case 'valueN': actionN statement(s);  
                    break;  
    default: default_action statement(s);  
}
```

Cấu trúc lặp

- **while loop**

```
while(condition)  
{  
    action statements;  
}
```

- **Do while**

```
do{  
    action statements;  
} while(condition);
```

- **For**

```
for(initialization statements; condition; increment statements){  
    action statements;}
```

Input/Output

- Nhập dữ liệu
- Xuất kết quả
- Format Xuất
- Đọc/ghi tập tin

Input

- Class **BufferedReader** trong Package **java.io**;
 - Khai báo: `BufferedReader dataIn = new BufferedReader(new InputStreamReader(System.in));`
 - Khai báo biến kiểu String chứa tạm dữ liệu nhập từ console:
 - `String temp = dataIn.readLine();`
- Class **Scanner** trong Package **java.util**;
 - Khai báo: `Scanner dataIn= new Scanner(System.in);`
 - Sử dụng phương thức `nextLine()`, `nextByte()`, `nextShort()`, `nextInt()`, `nextLong()`, `nextFloat()`, `nextDouble()`, hay `nextBoolean()` nhập dữ liệu dạng string, byte, short, int, long, float, double, hay boolean.
 - Ví dụ khai báo biến temp kiểu String nhận từ console:
 - `String temp=dataIn.nextLine();`

Định dạng xuất với printf

- Cú pháp:

- `System.out.printf(formatString)`
- `System.out.printf(formatString, argumentList)`

- `formatString`: mẫu định dạng các đối số output.

- `argumentList`: danh sách đối số xuất bao gồm constant values, variables, hoặc expressions.

- Các đối số ngăn cách bởi dấu ,

- Ví dụ:

- `System.out.printf("Hello there!");`
- `System.out.printf("There are %.2f inches in %d centimeters.%n", centimeters / 2.54, centimeters);`

Định dạng xuất với `printf`

- Cú pháp chuỗi định dạng:

`%[argument_index$][flags][width][.precision]conversion`

- *argument_index* : vị trí đối số : "1\$", "2\$,".
- *flags*: tập các kí tự thay đổi định dạng output.
- *width* : độ rộng output.
- *precision* : phần phân cách.
- *conversion* : cách đối số được định dạng.

Table 3-2 Some of the Supported Conversions

	Conversion	The result is
's'	general	a string
'c'	character	a Unicode character
'd'	integral	formatted as a (decimal) integer
'e'	floating point	formatted as a decimal number in computerized scientific notation
'f'	floating point	formatted as a decimal number
'%'	percent	'%'
'n'	line separator	the platform-specific line separator

Chuyển đổi Numeric- String

- ♦ Chuyển chuỗi số nguyên->int.

`Integer.parseInt(strExpression)`

`Integer.parseInt("6723") = 6723`

- ♦ Chuyển chuỗi số thực->float.

`Float.parseFloat(strExpression)`

`Float.parseFloat("34.56") = 34.56`

- ♦ Chuyển chuỗi số thực->double.

`Double.parseDouble(strExpression)`

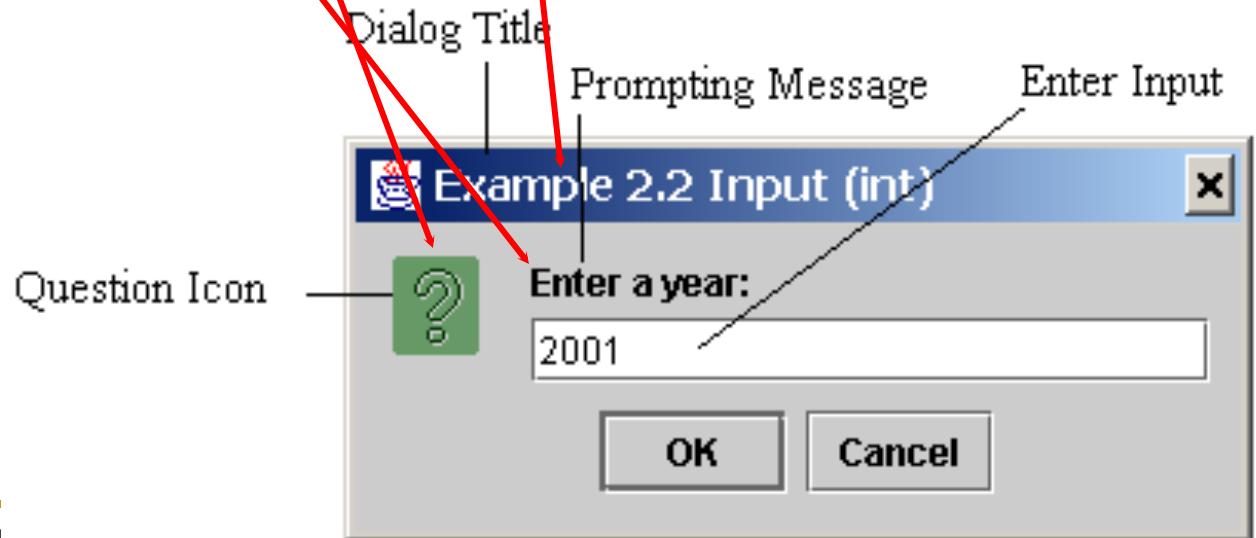
`Double.parseDouble("34.56") = 34.56`

Sử dụng Dialog Boxes Input/Output

- Sử dụng (graphical user interface).
- class JOptionPane
 - package javax.swing.
 - Các phương thức:
 - Input: `showInputDialog`
 - Output: `showMessageDialog`.
- Syntax:
`str = JOptionPane.showInputDialog(strExpression)`
- Chương trình kết thúc với `System.exit(0);`

Input/JOptionPane

```
String string = JOptionPane.showInputDialog(  
    null, "Prompting Message", "Dialog Title",  
    JOptionPane.QUESTION_MESSAGE);
```



Output/JOptionPane

The output of the statement

```
JOptionPane.showMessageDialog(null, "Hello World!", "Greetings",  
JOptionPane.INFORMATION_MESSAGE);
```

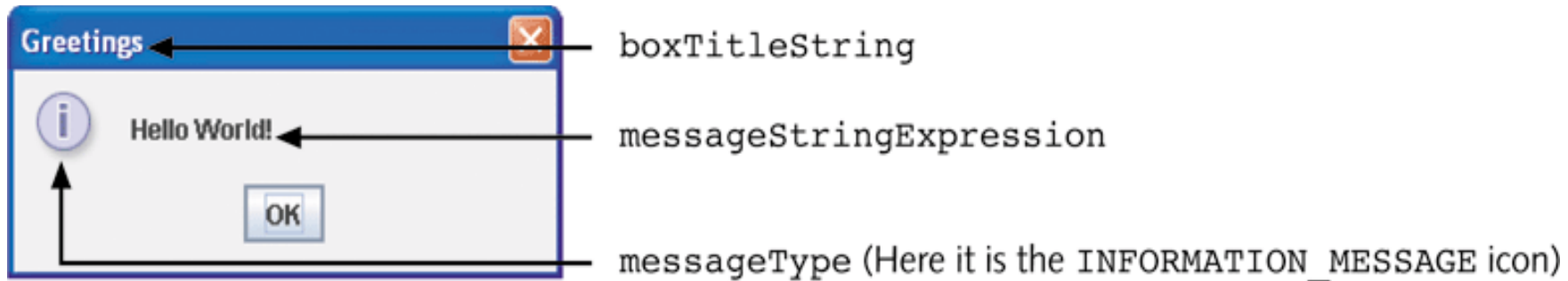


Figure 3-9 Message dialog box showing its various components

Định dạng Output sử dụng String.format

Ví dụ:

```
double x = 15.674;  
double y = 235.73;  
double z = 9525.9864;  
int num = 83;  
String str;
```

Expression	Value
<code>String.format("%.2f", x)</code>	<code>"15.67"</code>
<code>String.format("%.3f", y)</code>	<code>"235.730"</code>
<code>String.format("%.2f", z)</code>	<code>"9525.99"</code>
<code>String.format("%7s", "Hello")</code>	<code>" Hello"</code>
<code>String.format("%5d%7.2f", num, x)</code>	<code>" 83 15.67"</code>
<code>String.format("The value of num = %5d", num)</code>	<code>"The value of num = 83"</code>
<code>str = String.format("%.2f", z)</code>	<code>str = "9525.99"</code>

Tổng kết

- Cấu trúc chương trình JAVA
- Định danh, từ khóa ,t toán tử.
- Cấu trúc điều khiển
- Các kiểu dữ liệu - Primitive và Reference
- Input/output console

DEMO

Bài tập 1

- Tạo chương trình Fibonacci
 - Chạy chương trình
 - Sử dụng công cụ debug
- Viết chương trình tính diện tích hình chữ nhật
- Viết chương trình tìm kết quả các biểu thức sau:
 - $(a \ll 2) + (b \gg 2)$
 - $b > 0$
 - $(a + b * 100) / 10$
 - $a \& b$
 - Giả sử $a = 10$, $b = 5$
- Viết chương trình nhập số có 3 chữ số. Xuất đảo ngược
 - VD $789 \rightarrow 987$

Bài tập 2

- Viết chương trình nhập 3 số nguyên. Xuất giá trị lớn nhất
- Viết chương trình nhập số nguyên n. Xuất tổng các các số.
 - Ví dụ: 234-> 2+3+4=9
- Viết chương trình in ra số Armstrong
 - Armstrong là số có 3 chữ số sau cho : tổng lũy thừa 3 của từng phần tử bằng chính nó
- Viết chương trình khai báo dãy 10 số nguyên. Xuất ra giá trị lớn nhất
- Viết chương trình xuất ra các số nguyên tố nhỏ hơn 30
- Viết chương trình nhập họ tên.
 - Cho biết số kí tự
 - Cho biết kí tự 'a' xuất hiện những vị trí nào