

LẬP TRÌNH JAVA

Bùi Trọng Tùng

Bộ môn Truyền thông và Mạng máy tính

Viện CNTT – TT, Đại học BKHN

1

Về môn học này

- Mã học phần: IT3650
- Tên học phần: Lập trình Java
- Khối lượng: 3(0-0-6-6) ~ 90 tiết thực hành
- Học phần theo chương trình hợp tác đào tạo giữa trường Đại học BKHN và Tập đoàn Samsung
- Đánh giá:
 - Điểm quá trình (0.3): Hoàn thành các bài tập tuần
 - Điểm thi cuối kỳ (0.7): Hoàn thành project + 20% Điểm quá trình
- Website: <http://soict.hust.edu.vn/~tungbt/it3650>
- Tài liệu tham khảo:
<http://docs.oracle.com/javase/tutorial/index.html>

2

Học kỳ 20141

- Kế hoạch (dự kiến):
 - Tuần 2-9: Lý thuyết + Bài tập trên lớp + Bài tập tuần
 - Tuần 12-17: Thực hiện Project
 - Tuần 18: Trình bày Project trên lớp
- Thời gian đăng ký project: 01-08/12/2014
- Quy định chung xem trên website môn học

3

Giảng viên

Bùi Trọng Tùng,

Email: tungbt@soict.hust.edu.vn

Địa chỉ: Phòng 801 – Nhà B1 – Đại học BKHN

Website: <http://soict.hust.edu.vn/~tungbt>

4

BÀI 1.

GIỚI THIỆU VỀ NGÔN NGỮ LẬP TRÌNH JAVA

Giới thiệu chung về Java
Sử dụng trình biên dịch Java và máy ảo Java
Các phần tử cơ bản trong Java
Toán tử và biểu thức
Một số lệnh vào ra cơ bản

5

1. GIỚI THIỆU CHUNG VỀ JAVA

6

Lịch sử ngôn ngữ lập trình

- Hợp ngữ
 - MOV, ADD, JMP, CMP...
 - Ngôn ngữ lập trình tuần tự
 - Gần với ngôn ngữ máy
 - Chương trình cồng kềnh, khó sửa lỗi, bảo trì
- Ngôn ngữ lập trình hướng cấu trúc/thủ tục:
 - Pascal, C
 - Ngôn ngữ bậc cao, gần với ngôn ngữ tự nhiên
 - Dễ viết
 - Chương trình gồm các chương trình con (hàm, thủ tục)
 - Dữ liệu và xử lý của hàm, thủ tục tách rời nhau
 - Chương trình = Cấu trúc dữ liệu + Giải thuật

7

Ngôn ngữ lập trình hướng đối tượng

- Chương trình gồm các đối tượng và tương tác giữa chúng

Chương trình = Đối tượng + Thông điệp
- Mô hình hóa đối tượng thực thành các đối tượng phần mềm:
 - Dùng ngôn ngữ lập trình để mô tả đối tượng thực

Máy ATM – Đối tượng thực*	Mô hình hóa – Lớp (class) ATM
Thuộc tính: Ngân hàng, Vị trí	Thuộc tính: bank, location
Hành động: Kiểm tra mã PIN, Rút tiền, Chuyển khoản	Hành động: userAuthenticated(), withdraw(), tranfer()

(*) Trong ngữ cảnh này đề cập đến đối tượng máy ATM nói chung. Một đối tượng máy ATM cụ thể nào đó phải có thuộc tính cụ thể

8

Ngôn ngữ lập trình Java

- Java là một ngôn ngữ lập trình HĐT được phát triển bởi Sun Microsystems.
- Java là một ngôn ngữ lập trình khá trẻ
 - Ban đầu được sử dụng để xây dựng ứng dụng điều khiển các bộ xử lý bên trong các thiết bị điện tử dân dụng như máy điện thoại cầm tay, lò vi sóng...
 - Bắt đầu được sử dụng từ năm 1995 (với tên gọi là "Oak")



**Green Team and James Gosling
(the leader)**

9

Ngôn ngữ lập trình Java

- Đơn giản
 - Loại bỏ con trỏ
 - Không có goto, file header
 - Loại bỏ struct và union
- Hướng đối tượng
 - Java được thiết kế xoay quanh mô hình hướng đối tượng.
- Mạnh
 - Chặt chẽ → Loại bỏ các kiểu dữ liệu dễ gây lỗi
- Độc lập phần cứng
 - Viết một lần, chạy nhiều nơi (chỉ khác nhau trình thông dịch – máy ảo Java)

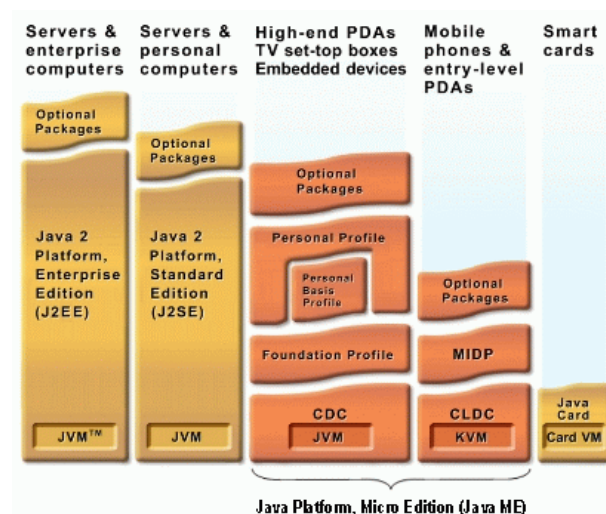
10

Ngôn ngữ lập trình Java

- Ngày nay, nhắc đến Java, không còn nhắc đến như một ngôn ngữ mà còn là một công nghệ, một nền tảng phát triển.
 - Java có một cộng đồng phát triển mạnh mẽ
 - Một tập hợp các thư viện với số lượng lớn (từ Sun và các nguồn khác)
- Java là ngôn ngữ vừa biên dịch vừa thông dịch
 - Biên dịch: Mã nguồn được biên dịch bằng công cụ JAVAC để chuyển thành dạng ByteCode
 - Thông dịch: Bytecode thực thi trên từng loại máy cụ thể nhờ chương trình thông dịch (nằm trong máy ảo Java)
 - Nhằm mục đích viết một lần, chạy nhiều nơi

11

Ngôn ngữ lập trình Java



12

Ngôn ngữ lập trình Java

- J2SE (Java 2 Platform Standard Edition)
 - Cung cấp các thành phần cốt lõi để xây dựng ứng dụng desktop-based
 - JRE: Java Runtime Environment: môi trường thực thi để chạy các ứng dụng Java
- J2EE (Java 2 Platform Enterprise Edition)
 - Xây dựng các ứng dụng hướng dịch vụ (service-oriented)
 - Web service
 - Ứng dụng doanh nghiệp
- J2ME (Java 2 Platform Mobile Edition): xây dựng ứng dụng di động

13

Cài đặt trình biên dịch và môi trường thực thi

- Download J2SE tại <http://www.oracle.com/technetwork/java/javase/download/index.html>
- Cài đặt
- Thiết lập biến môi trường

This PC → Properties → Advanced system settings
→ Environments Variables:

 - Tạo biến JAVA_HOME trong mục System Variables và đặt giá trị là thư mục cài đặt Java
 - Tạo (hoặc thêm) biến PATH: giá trị %JAVA_HOME%\bin;.;
 - Tạo (hoặc thêm) biến PATH: %JAVA_HOME%\bin;.;
 - %JAVA_HOME%\include

14

Kiểm tra Java đã được cài đặt đúng cách

- Trên cửa sổ Command Line lần lượt gõ 2 lệnh:
> javac -version
> Java -version

Nếu hai lệnh được thực hiện thành công, thông tin phiên bản Java sẽ xuất hiện → Java được cài đặt đúng

15

Chương trình Java đầu tiên

```
// The first Java program

public class HelloWorld {
    //Phương thức main, được gọi đầu tiên khi chạy bất cứ
    ứng dụng Java nào
    public static void main (String[] args) {
        System.out.println("Hello Java!");
    } //Kết thúc phương thức main
} //Kết thúc lớp HelloWorld
```

16

Khai báo lớp

```
// The first Java program
```

Khai báo lớp có tên là HelloWorld

```
public class HelloWorld {
```

```
//Phương thức main, được gọi đầu tiên khi chạy bất cứ ứng dụng Java nào
```

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

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

```
    } //Kết thúc phương thức main
```

```
} //Kết thúc lớp HelloWorld
```

Khai báo một phương thức của lớp HelloWorld

- Tạm thời, chúng ta chưa phân tích kỹ ý nghĩa của các khai báo này. Hãy để dành nó cho các bài sau.

17

Câu lệnh

```
// The first Java program
```

```
public class HelloWorld {
```

```
//Phương thức main, được gọi đầu tiên khi chạy bất cứ ứng dụng Java nào
```

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

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

```
    } //Kết thúc phương thức main
```

```
} //Kết thúc lớp HelloWorld
```

- Thực hiện các thao tác của lớp, xử lý của phương thức
- **Kết thúc bằng dấu ;**
- Một câu lệnh Có thể viết trên 1 hoặc nhiều dòng
- Ký hiệu để bao khối lệnh { }

18

Chú thích

```
// The first Java program
```

```
public class HelloWorld {
    /*Phương thức main, được gọi đầu tiên khi chạy bất cứ
    ứng dụng Java nào*/
    public static void main (String[] args) {
        System.out.println("Hello Java!");
    } //Kết thúc phương thức main
} //Kết thúc lớp HelloWorld
```

- Giải thích mã nguồn, các câu lệnh, các bước xử lý phức tạp trong chương trình:

// chú thích trên một dòng

/ chú thích trên một đoạn*/*

- Không có ý nghĩa thực thi

19

Dịch chương trình

- Sử dụng chương trình Notepad gõ lại đoạn lệnh trên
 - Bắt đầu tập thói quen lùi dòng cho các khối lệnh bao nhau
- Lưu file mã nguồn có tên Test.java vào thư mục D:\Java\20141\W1
- Dịch: mở cửa sổ Command Line và gõ lệnh sau để dịch
 >javac D:\Java\20141\W1\Test.java
- Lỗi dịch: *"class HelloWorld is public, should be declared in a file named HelloWorld.java"*
 - Nguyên nhân: Tên file mã nguồn khác tên lớp đã khai báo
 - Sửa: đặt lại tên file
- Dịch thành công: dấu nhắc lệnh > xuất hiện trở lại. File mã nguồn được dịch thành file Byte Code có cùng tên và đuôi .class, cùng thư mục với file mã nguồn

20

Chạy chương trình

- Thực hiện lệnh:

> java D:\Java\20141\W1\Test

Chú ý: Có thể sẽ gặp thông báo lỗi "Could not find or load main class..."

Khắc phục: chuyển vào thư mục chứa file .class

>D:

>cd .\Java\20141\W1

>java Test

- Kết quả thực hiện

```
D:\Java\bin>java HelloWorld
Hello World!
```

System.out.println(): Hiển thị thông báo trên cửa sổ Console

21

Sửa lại phương thức main

```
public static void main (String[] args) {
    myName = "Tung";
    System.out.println("Hello Java! I am " + myName);
} //Kết thúc phương thức main
} //Kết thúc lớp HelloWorld
```

- Lỗi dịch:

```
HelloWorld.java:6: error: cannot find symbol
    myName = "Tung";
    ^
symbol:   variable myName
location: class HelloWorld
HelloWorld.java:7: error: cannot find symbol
    System.out.println("Hello World! I am " + myName);
                                           ^
symbol:   variable myName
location: class HelloWorld
2 errors
```

- Nguyên nhân: chưa khai báo *myName*
- Chúng ta sẽ xem xét các sửa trong phần sau.

22

2. CÁC PHẦN TỬ CƠ BẢN CỦA JAVA

23

Từ khóa và định danh

- Là những từ được Java quy định ý nghĩa và cách sử dụng
- Định danh: xâu ký tự, xác định duy nhất một phần tử trong chương trình
- Quy định với định danh:
 - Không đặt trùng với từ khóa
 - Không bắt đầu bằng chữ số
 - Ký tự được phép sử dụng: chữ cái, chữ số, \$, _
 - Phân biệt chữ hoa, chữ thường

24

Quy tắc “con lạc đà”

- Biến số: bắt đầu bằng chữ thường, viết hoa chữ cái đầu tiên các từ còn lại
- Hằng số: Toàn bộ bằng chữ hoa
- Lớp: viết hoa chữ cái đầu tiên các từ
- Thuộc tính, phương thức: bắt đầu bằng chữ thường, viết hoa chữ cái đầu tiên các từ còn lại
- Gói: sử dụng chữ thường

Biến số	myName, numberOfStudent
Hằng số	MAX_LINE, USER_PARAMETER
Lớp	HelloWorld, Student
Thuộc tính	studentID, mark
Phương thức	setValue(), getValue(), sortByName()
Gói	sis.subject, sis.student

25

Các kiểu dữ liệu nguyên thủy

Kiểu dữ liệu	Kích thước	Giá trị mặc định	Giá trị nhỏ nhất	Giá trị lớn nhất
byte	8	0	-128	127
short	16	0	-32768	32767
int	32	0	-2147483648	2147483647
long	64	0L	-2^{63}	$(2^{63}) - 1$
float	32	0.0f		
double	64	0.0d		
boolean	Không xác định	false	NA	NA
char	16	\u0000	NA	NA

26

Toán tử số học

Toán tử	Ý nghĩa	Kiểu dữ liệu của toán hạng	Ví dụ
-	Phép đổi dấu	Số thực hoặc số nguyên	int a, b; -12; -a; -25.6;
+	Phép toán cộng	Số thực hoặc số nguyên	float x, y; 5 + 8; a + x; 3.6 + 2.9;
-	Phép toán trừ	Số thực hoặc số nguyên	3 - 1.6; a - 5;
*	Phép toán nhân	Số thực hoặc số nguyên	a * b; b * y; 2.6 * 1.7;
/	Phép toán chia	Số thực hoặc số nguyên	10.0/3.0; (bằng 3.33...) 10/3.0; (bằng 3.33...) 10.0/3; (bằng 3.33...)
/	Phép chia lấy phần nguyên	Giữa 2 số nguyên	10/3; (bằng 3)
%	Phép chia lấy phần dư	Giữa 2 số nguyên	10%3; (bằng 1)

Toán tử nhị phân

Toán tử	Ý nghĩa	Kiểu dữ liệu của toán hạng	Ví dụ
&	Phép VÀ nhị phân	2 số nhị phân	0 & 0 (có giá trị 0) 0 & 1 (có giá trị 0) 1 & 0 (có giá trị 0) 1 & 1 (có giá trị 1) 101 & 110 (có giá trị 100)
	Phép HOẶC nhị phân	2 số nhị phân	0 0 (có giá trị 0) 0 1 (có giá trị 0) 1 0 (có giá trị 0) 1 1 (có giá trị 1) 101 110 (có giá trị 111)

Toán tử nhị phân

^	Phép HOẶC CÓ LOẠI 2 số nhị phân TRỪ nhị phân	$0 \wedge 0$	(có giá trị 0)
		$0 \wedge 1$	(có giá trị 1)
		$1 \wedge 0$	(có giá trị 1)
		$1 \wedge 1$	(có giá trị 0)
		$101 \wedge 110$	(có giá trị 011)
<<	Phép DỊCH TRÁI nhị phân	$a \ll n$	(có giá trị $a \cdot 2^n$)
		$101 \ll 2$	(có giá trị 10100)
>>	Phép DỊCH PHẢI nhị phân	$a \gg n$	(có giá trị $a/2^n$)
		$101 \gg 2$	(có giá trị 1)
~	Phép ĐẢO BIT nhị phân (lấy Bù 1)	~ 0	(có giá trị 1)
		~ 1	(có giá trị 0)
		~ 110	(có giá trị 001)

29

Toán tử quan hệ

Toán tử	Ý nghĩa	Ví dụ
>	So sánh lớn hơn giữa 2 số nguyên hoặc thực.	$2 > 3$ (có giá trị 0) $6 > 4$ (có giá trị 1) $a > b$
>=	So sánh lớn hơn hoặc bằng giữa 2 số nguyên hoặc thực.	$6 \geq 4$ (có giá trị 1) $x \geq a$
<	So sánh nhỏ hơn giữa 2 số nguyên hoặc thực.	$5 < 3$ (có giá trị 0),
<=	So sánh nhỏ hơn hoặc bằng giữa 2 số nguyên hoặc thực.	$5 \leq 5$ (có giá trị 1) $2 \leq 9$ (có giá trị 1)
==	So sánh bằng nhau giữa 2 số nguyên hoặc thực.	$3 == 4$ (có giá trị 0) $a == b$
!=	So sánh không bằng (so sánh khác) giữa 2 số nguyên hoặc thực.	$5 != 6$ (có giá trị 1) $6 != 6$ (có giá trị 0)

30

Toán tử logic

Toán tử	Ý nghĩa	Kiểu dữ liệu của toán hạng	Ví dụ
&&	Phép VÀ LOGIC. Biểu thức VÀ LOGIC bằng 1 khi và chỉ khi cả 2 toán hạng đều bằng 1	Hai biểu thức logic	3<5 && 4<6 (có giá trị 1) 2<1 && 2<3 (có giá trị 0) a > b && c < d
	Phép HOẶC LOGIC. Biểu thức HOẶC LOGIC bằng 0 khi và chỉ khi cả 2 toán hạng bằng 0.	Hai biểu thức logic	6 0 (có giá trị 1) 3<2 3<3 (có giá trị 0) x >= a x == 0
!	Phép PHỦ ĐỊNH LOGIC một ngôi. Biểu thức PHỦ ĐỊNH LOGIC có giá trị bằng 1 nếu toán hạng bằng 0 và có giá trị bằng 0 nếu toán hạng bằng 1	Biểu thức logic	!3 (có giá trị 0) !(2>5) (có giá trị 1)

31

Các toán tử khác

- Toán tử rút gọn: +=, -=, *=, /= ...
- Toán tử tăng 1 đơn vị: ++
- Toán tử giảm 1 đơn vị: --
- Toán tử điều kiện: ?:

(boolean_expression)?true_expression:false_expression

- Nếu *boolean_expression* đúng, tính giá trị *true-expression*
- Nếu *boolean_expression* sai, tính giá trị *false-expression*

32

Hằng số

- Phần tử trong chương trình không thể thay đổi giá trị
- Cú pháp:


```
final DataType CONSTANT_NAME = Literal;
```
- Trong đó:
 - **final**: từ khóa
 - **DataType**: Kiểu dữ liệu
 - **CONSTANT_NAME**: Tên hằng. Tuân thủ quy tắc định danh
 - **Literals**: Giá trị hằng

33

Giá trị hằng (Literals)

- Boolean: `true`, `false`
- Số nguyên:
 - Hệ cơ số 8: Bắt đầu bằng chữ số 0
 - ❖ Ví dụ: $012 = 001010_{(2)} = 8 + 2 = 10_{(10)}$
 - Hệ cơ số 16: Bắt đầu bằng 0x
 - ❖ Ví dụ: $0x2A = 00101010 = 2 \times 16 + 10 = 42$
 - Kiểu dữ liệu `long`: Kết thúc bằng ký tự `L` hoặc `l`
 - ❖ Ví dụ: `10L`

34

Giá trị hằng

- Số thực:
 - Mặc định có kiểu `double`
 - Kiểu `float`: Kết thúc bằng ký tự `F` hoặc `f`
 - Dạng dấu phẩy động: Ký tự `e` (hoặc `E`) kèm theo số mũ
 - ❖ Ví dụ: `1.2E7`
- Ký tự: Đặt giữa dấu nháy đơn. Ví dụ: `'a'`
- Xâu ký tự: Đặt giữa dấu nháy kép
 - ❖ Ví dụ: `"SoICT-HUST"`

35

Biến số

- Là phần tử trong chương trình có thể thay đổi giá trị
- Cú pháp:


```
DataType varName1, varName2, ..., varNameN;
```

 hoặc


```
DataType varName1 = Literal1, ..., varNameN = LiteralN;
```
- Trong đó:
 - `varName` là tên biến, đặt theo quy tắc định danh
 - `Literal` có thể là một biến khác đã được khai báo trước
- Trước khi sử dụng trong biểu thức, biến phải được khởi tạo giá trị

36

Toán tử gán

- Cú pháp:

```
variable = expression;
```

- Biến `variable` và biểu thức `expression` nên có cùng kiểu dữ liệu
- Trong trường hợp hai vế có kiểu dữ liệu khác nhau:
 - Vế trái có kiểu dữ liệu “rộng” hơn: ép kiểu tự động
 - Ngược lại: không hợp lệ. Nếu vẫn muốn thực hiện phép gán, cần ép kiểu
- Trong có các giá trị khác kiểu, tất cả các giá trị được ép tự động thành kiểu rộng nhất

37

Toán tử gán (Ví dụ)

```
long a = 1.2; //không hợp lệ
long b = (long) 1.2; //hợp lệ
int m = b/2; //không hợp lệ
char ch = 'a'; //hợp lệ
int n = ch; //hợp lệ
short k = ch; //không hợp lệ
short p = (short) ch; //hợp lệ
float x = 1.2; //không hợp lệ
float y = 1.2f; //hợp lệ
```

38

Hãy thử lại

```
// The second Java program

public class NewHelloWorld {
    //Phương thức main, được gọi đầu tiên khi chạy bất cứ
    ứng dụng Java nào
    public static void main (String[] args) {
        String myName = "Tung";
        System.out.println("Hello Java! I am " + myName);
    } //Kết thúc phương thức main
} //Kết thúc lớp HelloWorld
```

39

3. CÁC PHƯƠNG THỨC VÀO RA CƠ BẢN

40

Hiển thị dữ liệu

- Phương thức `System.out.println()`: Hiển thị dữ liệu và xuống dòng
- Phương thức `System.out.print()`: Hiển thị dữ liệu
- Phương thức `System.out.printf()`: Hiển thị dữ liệu có định dạng
- Phương thức `System.out.format()`: Hiển thị dữ liệu có định dạng
- Có thể dùng toán tử `+` để nối các dữ liệu khi hiển thị

41

Định dạng dữ liệu khi hiển thị

- Dạng Boolean: `%b`
- Dạng ký tự: `%c`
- Dạng số nguyên: `%d`
- Dạng số thực: `%f`
- Dạng chuỗi ký tự: `%s`

42

Nhập dữ liệu từ bàn phím

- Khá phức tạp vì Java coi dữ liệu nhận được từ bàn phím là luồng vào
- Thực hiện
 - Đọc dữ liệu vào bộ đệm:


```
BufferedReader br = new BufferedReader(
    new InputStreamReader(System.in));
```
 - Chuyển dữ liệu từ bộ đệm vào xâu


```
String inValue = br.readLine( );
```
 - Chuyển dữ liệu từ xâu thành giá trị : sử dụng các lớp
 - ❖ Giá trị kiểu int: `Integer.parseInt(inValue)`
 - ❖ Giá trị kiểu long: `Long.parseLong(inValue)`
 - ❖ Giá trị kiểu float: `Float.parseFloat(inValue)`
 - ❖ Giá trị kiểu double: `Double.parseDouble(inValue)`

43

Ví dụ

```
/** The Addition class calculates the sum of two numbers */
import java.io.*;
public class Addition {
    /** The main method begins execution of Java application
    * @param args: input parameter
    */
    public static void main (String[] args) throws
        IOException{
        String inputData;
        BufferedReader br = new BufferedReader(new
            InputStreamReader(System.in));
        System.out.print("Enter the 1st number:");
        inputData = br.readLine();
        int number1 = Integer.parseInt(inputData);
```

44

Ví dụ (tiếp)

```

        System.out.print("Enter the 2nd number:");
        inputData = br.readLine();
        int number2 = Integer.parseInt(inputData);
        int sum = number1 + number2;
        System.out.println("The sum of two numbers: " + sum);
    }
}

```

45

Ví dụ - Giải thích

- Chú thích tạo tài liệu Javadoc:
`/** Comment something */`
- Sử dụng các gói thư viện được Java định nghĩa sẵn
`import somepackage`
 - `java.io`: Thư viện xuất nhập dữ liệu
- Bỏ qua các ngoại lệ (lỗi) trong khi thực thi chương trình:
`throws someException`
 - `IOException`: ngoại lệ xuất hiện khi xuất nhập dữ liệu

46

Nhập dữ liệu – Lớp Scanner

- Được cung cấp bởi thư viện java.util
- Quét luồng dữ liệu người dùng nhập từ bàn phím và phân tách các giá trị có kiểu dữ liệu nguyên thủy hoặc xâu.
- Rất hữu dụng

```
import java.util.Scanner  
...  
Scanner inputData = new Scanner(System.in);  
System.out.print("Enter the 1st number:");  
int number1 = inputData.nextInt();  
System.out.print("Enter the 2nd number:");  
int number2 = inputData.nextInt();  
  
int sum = number1 + number2;  
System.out.println("The sum of two numbers: " + sum);
```

47