

**I. MỤC TIÊU:**

- Làm quen với ngôn ngữ lập trình java.
- Biến, kiểu dữ liệu.
- Cấu trúc điều khiển trong java để giải quyết một số bài toán đơn giản.

**II. TÓM TẮT LÝ THUYẾT:****1. Cấu trúc chương trình java:**

```
[ package <tên_gói>; ]

[ import <tên_gói>; ]
(có thể import nhiều gói)

public class <tên_lớp> {
    public static void main(String[] args) {
        System.out.println("Hello, java!");
    }
}
```

**2. Kiểu dữ liệu cơ bản**

- Số nguyên: byte, short, int, long
- Số thực: float, double
- Kiểu ký tự: char
- Kiểu luận lý: boolean

**3. Các phép toán:**

- Toán tử số học: + - \* / %
- Toán tử quan hệ: == != > >= < <=
- Phép toán luận lý: && || !
- Toán tử điều kiện: ?:

**4. Khai báo biến, hằng số:**

- <kiểu dữ liệu> ten\_bien;
- final <kiểu dữ liệu> TEN\_HANG;

**5. Chuyển đổi kiểu dữ liệu:**

- Chuỗi sang số: Integer.parseInt(), Float.parseFloat(), Long.parseLong...
- Số sang chuỗi: String.valueOf(), n + ""

**6. Nhập / Xuất:**

- import thư viện java.util.\*
- Sử dụng đối tượng Scanner: nextLine(), nextInt(), nextLong(), nextDouble()... để nhập dữ liệu.
- Xuất kết quả: System.out.println("Hello");

**7. Cấu trúc điều khiển:***a. Cấu trúc rẽ nhánh:***- Cấu trúc if:**

```
if ( <biểu thức điều kiện> )  
    Khối lệnh  
[ else  
    Khối lệnh ]
```

**- Cấu trúc switch:**

```
switch( integer ) {  
    case 1: lệnh  
        ...  
        break;  
    case 2: lệnh  
        ...  
        break;  
    ...  
    [default: lệnh ]  
}
```

**Đối với Java 7+**

```
switch( string ) {  
    case "string1": lệnh  
        ...  
        break;  
    case "string2": lệnh  
        ...  
        break;  
    ...  
    [default: lệnh ]  
}
```

*b. Cấu trúc lặp:*

```
while( điều kiện lặp )  
    Khối lệnh
```

```
do {  
    Lệnh;  
} while (điều kiện lặp );
```

```
for (khởi tạo; điều kiện lặp; tăng biến đếm)  
    Khối lệnh
```

```
for (int item : items)  
    Khối lệnh
```

**8. Kiểu chuỗi:**

- Khai báo: `String str = "";`
- Phép nối chuỗi: `+`
- Một số hàm thường dùng

Tên hàm	Ý nghĩa
substring	Trích chuỗi
replaceAll	Thay thế chuỗi theo mẫu (partten)
charAt	Lấy ký tự trong chuỗi
equals	So sánh 2 chuỗi có bằng nhau không
compareTo	So sánh 2 chuỗi ( <b>=0</b> : bằng, <b>&lt;0</b> : $s1 < s2$ , <b>&gt;0</b> : $s1 > s2$ )
length	Lấy chiều dài chuỗi
toLowerCase	viết thường chuỗi.
toUpperCase	VIẾT HOA CHUỖI.
trim	Cắt bỏ các khoảng trắng dư thừa ở đầu và cuối chuỗi.
indexOf	Tìm vị trí xuất hiện đầu tiên của chuỗi
lastIndexOf	Tìm vị trí xuất hiện sau cùng của chuỗi

**9. Kiểu Date/Time:**

- Sử dụng thư viện java.util: Date
- Sử dụng thư viện java.text: SimpleDateFormat

Một số thao tác trên đối tượng Date

*Chuyển Date sang String:*

```
SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");  
  
String date = sdf.format(new Date());  
  
System.out.println(date); // 22/01/2018
```

*Chuyển String sang Date:*

```
SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");  
  
String dateInString = "22/01/2018";  
  
Date date = sdf.parse(dateInString);  
  
System.out.println(date); // Tue Jan 23 00:00:00 ICT 2018
```

*Lấy ngày giờ hiện hành:*

```
SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");  
  
Date date = new Date();  
  
System.out.println(dateFormat.format(date)); // 2018/01/22 16:16:39
```

*So sánh ngày tháng:*

Dùng phương thức **Date.compareTo** để so sánh 2 đối tượng Date

```
date1.compareTo(date2)
```

- Trả về giá trị 0 nếu date1 bằng date2
- Trả về giá trị > 0 nếu date1 sau ngày date2
- Trả về giá trị < 0 nếu date1 trước ngày date2

**10. Kiểu mảng:***a. Mảng 1 chiều:*

- Khai báo: `int a[];`
- Cấp phát vùng nhớ: `a = new int[10];`
- a. Lấy số phần tử của mảng: **`a.length`**

*b. Mảng 2 chiều:*

- Khai báo: `int[][] a;`
- Cấp phát vùng nhớ: `a = new int[5][5];`

**11. Xử lý lỗi (Exception)**

```
try {  
    lệnh có thể gây lỗi  
} catch (Exception e) {  
    xử lý lỗi  
} [ finally {  
    giải phóng tài nguyên  
} ]
```

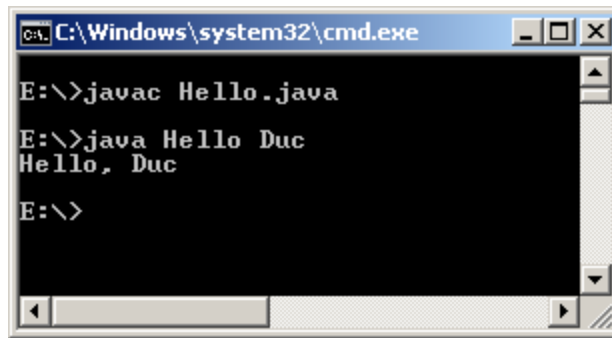
**III. NỘI DUNG THỰC HÀNH:****0. Lab1-0:** Tạo và biên dịch chương trình java – Debug chương trình (*bài đọc thêm, làm ở nhà*)**1) Tạo và biên dịch chương trình java***a. Dùng command line:*

- Mở notepad
- Nhập vào đoạn mã sau:

```
1 class Hello {  
2     public static void main(String[] a) {  
3         System.out.println("Hello, " + a[0]);  
4     }  
5 }
```

**Hình 1**

- Lưu ở đĩa E: đặt tên **Hello.java**
- Dùng lệnh `javac` để biên dịch thành mã bytecode
- Dùng lệnh `java` để chạy chương trình



```
C:\Windows\system32\cmd.exe
E:\>javac Hello.java
E:\>java Hello Duc
Hello, Duc
E:\>
```

*b. Dùng Netbean:*

- Mở Netbean.
- Tạo mới project đặt tên MSSV\_Ten
- Tạo lớp mới đặt tên là Hello
- Nhập code như hình 1
- Để chạy chương trình kích phải chuột vào file Hello.java, chọn Run File, hoặc nhấn Shift + F6
- Export dự án trên, lưu thành file MSSV\_Ten.zip
- Xóa dự án MSSV\_Ten trong Netbean (đánh dấu xóa hoàn toàn)
- Import lại dự án từ file MSSV\_Ten.zip đã lưu.

**2) Debug chương trình:**

- Mở Netbean.
- Import project **Lab1.zip** vào Netbean
- Mở file **Lab1\_0.java**

```
1 public class Lab1_0 {
2     public static void main(String[] args) {
3         int n = 10;
4         Fibonacci(n);
5     }
6     public static void Fibonacci(int n) {
7         int f1 = 1;
8         int f2 = 1;
9         System.out.print(f1 + "\t" + f2);
10        int f;
11        for(int i=3; i<=n; i++) {
12            f = f1 + f2;
13            f1 = f2;
14            f2 = f;
15            System.out.print("\t" + f);
16        }
17    }
18 }
```

- Đánh dấu các điểm breakpoint ở dòng 12, 13, 14
- Nhấn Ctrl + Shift + F5 để chạy chương trình ở chế độ debug
- Nhấn F5 để chạy từng lệnh được đánh dấu.
- Trong cửa sổ debug, chọn tag Variables, xem kết quả thay đổi sau khi chạy từng bước.

1. Mở file **Lab1\_1.java**: chương trình nhập vào 2 số nguyên, in ra kết quả so sánh 2 số vừa nhập.

```
1  import java.util.*;
2  public class Lab1_1 {
3      public static void main(String[] args) {
4          Scanner input = new Scanner(System.in);
5          int a, b;
6          System.out.print("Nhập số a: ");
7          a = input.nextInt();
8          System.out.print("Nhập số b: ");
9          b = input.nextInt();
10         if (a == b) {
11             System.out.println(a + " = " + b);
12         } else {
13             System.out.println(a + " <> " + b);
14         }
15     }
16 }
```

**Yêu cầu:**

- Chạy chương trình, nhập lần lượt các giá trị: (3, 3) ( 2, 5 )
- Quan sát kết quả.
- Hãy chỉnh sửa code để chương trình so sánh được các trường hợp khác (lớn hơn, nhỏ hơn)

**Bài tập 1:**

- Viết chương trình nhập vào 3 số nguyên, tìm số lớn nhất.



2. Mở file **Lab1\_2.java**: chương trình nhập vào 2 số thực và phép toán, in ra kết quả tương ứng của phép toán nhập vào.

```
1 import java.util.*;
2 public class Lab1_2 {
3     public static void main(String[] args) {
4         Scanner input = new Scanner(System.in);
5         float x, y, ketqua = 0;
6         String c;
7         System.out.print("x = ");
8         x = input.nextFloat();
9         System.out.print("y = ");
10        y = input.nextFloat();
11        input.nextLine(); //Xóa ký tự xuống dòng
12        System.out.print("Phep toan: ");
13        c = input.nextLine();
14        switch (c) {
15            case "+":
16                ketqua = x + y;
17                break;
18            case "-":
19                ketqua = x - y;
20                break;
21        }
22        System.out.println(x + " " + c + " " + y + " = " + ketqua);
23    }
24 }
```

### **Yêu cầu:**

- Chạy chương trình, nhập lần lượt các giá trị: ( 2, 3, + ), ( 5, 5, - ), ( 7, 6, a ). Quan sát kết quả.
- Chỉnh sửa lại chương trình trên như sau:
  - Cho nhập thêm phép toán: \*, /
  - Báo lỗi khi nhập sai phép toán ( +, -, \*, / ) (dùng **default**)
  - Báo lỗi khi có phép chia 0

### **Bài tập 2:**

- Viết chương trình nhập vào tháng (1-12) in ra Quý tương ứng, báo lỗi nếu nhập tháng sai.

3. Mở file **Lab1\_3.java**: chương trình nhập vào một mảng các số nguyên, xuất ra mảng vừa nhập và tổng các phần tử.

```
1 import java.util.*;
2 public class Lab1_3 {
3     public static void main(String[] args) {
4         Scanner input = new Scanner(System.in);
5         int a[], n;
6         System.out.print("Nhap so phan tu: ");
7         n = input.nextInt();
8         a = new int[n];
9         for(int i=0; i<a.length; i++) {
10             System.out.print("a[" + i + "]=");
11             a[i] = input.nextInt();
12         }
13         System.out.print("\nMang vua nhap: ");
14         XuatMang(a);
15         System.out.print("\n\nTong cac phan tu: " + Tong(a));
16     }
17     public static void XuatMang(int a[]) {
18         for(int i=0; i<a.length; i++)
19             System.out.print("\t" + a[i]);
20     }
21     public static int Tong(int a[]) {
22         int tong = 0;
23         for(int i=0; i<a.length; i++)
24             tong += a[i];
25         return tong;
26     }
27 }
```

**Yêu cầu:**

- Chạy chương trình, nhập mảng, quan sát kết quả
- Trong chương trình chính, viết thêm hàm **TongLe** tính tổng các phần tử có giá trị lẻ.

**Bài tập 3:**

- Viết chương trình khai báo mảng số thực, viết hàm nhập / xuất mảng, tìm phần tử lớn nhất trong mảng.

**4. Mở file Lab1\_4.java:** chương trình minh họa chuẩn hóa chuỗi

```
1 public class Lab1_4 {
2     public static void main(String[] args) {
3         String s = "  dai hoc  cong  nghe sai gon  ";
4         System.out.println("Chieu dai: " + s.length());
5         s = s.trim();
6         System.out.println("Bỏ khoảng trắng trước và sau chuỗi: '" + s + "'");
7         s = s.replaceAll(" +", " ");
8         System.out.println("Bỏ khoảng trắng dư thừa: '" + s + "'");
9     }
10 }
```

**Yêu cầu:**

- Chạy chương trình. Quan sát kết quả.
- Hãy viết thêm code để viết hoa đầu mỗi từ.

**Bài tập 4:**

- Viết chương trình nhập vào họ tên, in tách họ và tên  
Vd : Ho Dinh Kha  
→ Ho Dinh  
→ Kha

**5. Mở file Lab1\_5.java:** chương trình in ra ngày tháng năm hiện hành theo định dạng dd/MM/yyyy.

```
1 import java.text.SimpleDateFormat;
2 import java.util.*;
3 public class Lab1_6 {
4     public static void main(String[] args) {
5         SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
6         Date date;
7         date = new Date(); //Lay ngay gio hien hanh
8         System.out.println("Ngày hiện tại: " + sdf.format(date));
9     }
10 }
```

**Yêu cầu:**

- Chạy chương trình, quan sát kết quả.
- Trong chương trình chính, tạo thêm đối tượng ngày, in theo định dạng yyyy-MM-dd

**Bài tập 5 :**

- Viết chương trình nhập ngày tháng năm theo định dạng dd/MM/yyyy, so sánh với ngày tháng hiện tại xem ngày đó trước, sau, hay bằng ngày hiện tại.

**BÀI TẬP LÀM THÊM:**

- Viết chương trình nhập vào giờ làm việc, lương theo giờ của một nhân viên. Tính lương của nhân viên đó theo công thức sau :
  - Lương được tính theo giờ khi số giờ  $< 40 \rightarrow$  lương giờ  $\times$  số giờ
  - Nếu số giờ  $\geq 40$  thì được tính theo lương tuần + số giờ dư ra
  - Lương tuần  $= 40 \times$  lương giờ  $\times 1.5$
- Viết chương trình nhập vào chuỗi nhị phân  $\rightarrow$  Xuất ra giá trị thập phân tương ứng.  
Ví dụ: "10001"  $\rightarrow$  17
- Viết chương trình nhập vào một biểu thức có 2 toán hạng, tính ra giá trị của biểu thức đó.  
Ví dụ : "5 + 3"  $\rightarrow$  8
- Mở rộng bài tập 3, cho phép nhập biểu thức có nhiều toán hạng. (chỉ cần làm phép +, -)  
Ví dụ : "5 + 4 - 5"  $\rightarrow$  4
- Viết chương trình nhập vào một chuỗi. Viết hàm in ra số lượng các chữ số có trong chuỗi.  
Ví dụ : "abc1232445"
  - 1  $\rightarrow$  1
  - 2  $\rightarrow$  2
  - 3  $\rightarrow$  1
  - 4  $\rightarrow$  2
  - 5  $\rightarrow$  1
- Cho mảng số nguyên có n phần tử. Viết hàm thực hiện các công việc sau:
  - Nhập mảng
  - Xuất mảng
  - Sắp xếp mảng tăng dần
  - Vẽ dấu \* tương ứng với giá trị của phần tử

Ví dụ: 1 4 2 3 2

```
*
* * * *
* *
* * *
* *
```
- Viết chương trình nhập vào chuỗi s. Đảo ngược các từ trong chuỗi s  
Ví dụ: s = "khoa cong nghe thong tin"  $\rightarrow$  "tin thong nghe cong khoa"
- Viết chương trình nhập vào chuỗi s. Sau đó nhập chuỗi con s1, và vị trí k cần chèn trong chuỗi s. Viết hàm chèn chuỗi s1 vào chuỗi s tại vị trí k