

I. Mục tiêu:

- Cấu trúc ứng dụng Java.
- Làm quen với ngôn ngữ lập trình java.
- Biến, kiểu dữ liệu nguyên thủy
- Cấu trúc điều khiển giải quyết một số bài toán đơn giản.

II. Tóm Tắt Lý Thuyết:**1. Kiểu dữ liệu cơ bản**

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

2. Các phép toán:

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

3. Khai báo biến: <kiểu dữ liệu> ten_bien;**4. Khai báo hằng số:** 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: 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 ]  
}
```

Nếu Java 7+ thì switch có thể dùng String

- *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 (=0: bằng, <0: s1<s2, >0: 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];`
- Khai báo và cấp phát: `int a[]={4,6,5,7,8};`
- 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. Bài Tập.

1. Sử dụng Debugger trong Eclipse

- Khởi động Eclipse
- Tạo một project tên Lab01
- Gõ vào cấu trúc chương trình tính dãy fibonacci như sau:

- Đánh dấu các điểm breakpoint ở dòng 12, 13, 14

```
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 }
```

- 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.

2. Viết chương trình như sau:

```
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("Nhap so a: ");
7         a = input.nextInt();
8         System.out.print("Nhap so 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)
3. 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

4. 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.
5. Viết chương trình khai báo mảng số thực, viết hàm nhập / xuất mảng, tìm phần tử lớn nhất trong mảng.


```
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.
6. 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("Ngay hien tai: " + 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*
7. Viết chương trình nhập vào 3 số nguyên, tìm số lớn nhất.
8. Viết chương trình nhập vào tháng (1-12) in ra Quý tương ứng, báo lỗi nếu nhập tháng sai.
9. Viết chương trình khai báo mảng số thực, viết hàm nhập / xuất mảng, tìm phần tử lớn nhất trong mảng.
10. Viết chương trình nhập vào họ tên, in tách họ và tên
Vd : Ho Dinh Kha
→ Ho
→ Kha
11. Viết chương trình nhập ngày tháng năm theo định danh dd/MM/yyyy, so sánh với ngày tháng hiện tại xem ngày đó trước, sau, hay bằng ngày hiện tại.

12. Viết chương trình nhập vào 3 điểm toán, lý, hóa. In ra điểm trung bình và xếp loại của điểm biết :
 - Nếu điểm trung bình <5 thì in ra màn hình “Rớt!”
 - Nếu điểm trung bình ≥ 5 thì in ra màn hình “Đậu!” và
 - Nếu điểm trung bình ≥ 5 và <7 thì in ra màn hình “Trung Bình!”
 - Nếu điểm trung bình ≥ 7 và <8 thì in ra màn hình “Khá!”
 - Nếu điểm trung bình ≥ 8 và <9 thì in ra màn hình “Giỏi!”
 - Nếu điểm trung bình ≥ 9 thì in ra màn hình “Xuất Sắc!”
13. Viết chương trình nhập vào một số nguyên kiểm tra xem số nhập vào có phải là bội số của 3 và 5 hay không
14. Viết chương trình nhập vào một số nguyên dương và xuất ra màn hình là số này có phải là số chính phương không
15. Viết một chương trình :
 - Nhập vào tên của người dùng và xuất ra màn hình dòng chữ « Hello ! <Tên Người Dùng>
16. Viết một chương trình nhập vào một chuỗi và xuất ra cái chuỗi đảo ngược của chuỗi đó
17. Viết một chương trình nhập vào một chuỗi và xuất ra màn hình số ký tự chữ cái, số khoảng trắng, số ký tự số
18. Viết một chương trình đưa vào một arraylist String và đầu ra là arraylist đảo ngược của arraylist đưa vào (nhập done để kết thúc nhập arraylist)
19. Viết một chương trình đưa vào 1 arraylist interger và đầu ra là arraylist được sắp xếp theo thứ tự tăng dần (nhập 0 để kết thúc nhập arraylist)
20. Viết một chương trình Java so sánh 2 arraylist nếu giống nhau thì in ra màn hình “true” còn ngược lại in ra “false” (nhập 0 để kết thúc)