**Java Basic Practice**

## 1. Bài tập java cơ bản

Trong phần này, bạn phải nắm được các kiến thức về:

* Các mệnh đề if-else, switch-case.
* Các vòng lặp for, while, do-while.
* Các từ khóa break và continue trong java.
* Các toán tử trong java.
* Mảng (array) trong java.
* File I/O trong java.
* Xử lý ngoại lệ trong java.

### Bài tập Java: xử lý số

**Danh sách bài tập**:

1. Viết chương trình giải phương trình bậc 2: ax2 + bx + c = 0.
2. Viết chương trình tính giai thừa của n.
3. Viết chương trình chuyển đổi một số tự nhiên ở hệ số 10 thành một số ở hệ cơ số B (1 <= B <= 32) bất kỳ.
4. Dãy số Fibonacci được định nghĩa như sau: F0 = 0, F1 = 1, F2 = 1, Fn = F(n-1) + F(n-2) với n >= 2. Ví dụ: 0, 1, 1, 2, 3, 5, 8, … Hãy viết chương trình tìm số Fibonacci thứ n.
5. Viết chương trình tìm ước số chung lớn nhất (UCLN) và bội số chung nhỏ nhất (BCNN) của hai số a và b.
6. Viết chương trình liệt kê tất cả các số nguyên tố nhỏ hơn n.
7. Viết chương trình liệt kê n số nguyên tố đầu tiên.
8. Viết chương trình liệt kê tất cả các số nguyên tố có 5 chữ số.

### ****Bài tập java: xử lý chuỗi****

1. Nhập một mảng số thực a0, a1, a2, …, an-1. Không dùng thêm mảng số thực nào khác (có thể dùng thêm mảng số nguyên), hãy in ra màn hình mảng trên theo thứ tự tăng dần. .
2. Viết chương trình nhập vào mảng A có n phần tử, các phần tử là số nguyên lớn hơn 0 và nhỏ hơn 100. Thực hiện các chức năng sau:   
   a) Tìm phần tử lớn thứ nhất và lớn thứ 2 trong mảng với các chỉ số của chúng (chỉ số đầu tiên tìm được).   
   b) Sắp xếp mảng theo thứ tự tăng dần.   
   c) Nhập số nguyên x và chèn x vào mảng A sao cho vẫn đảm bảo tính tăng dần cho mảng A.

### Dãy số Fibonacci được định nghĩa như sau: F0 = 0, F1 = 1, F2 = 1, Fn = F(n-1) + F(n-2) với n >= 2. Ví dụ: 0, 1, 1, 2, 3, 5, 8,

**Danh sách bài tập**:

### ****Bài tập java: xử lý mảng****

**Danh sách bài tập**:

## 2. Bài tập java nâng cao

Trong phần này, bạn phải nắm được các kiến thức về:

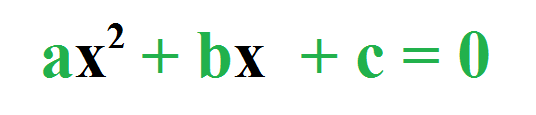
* Lớp và đối tượng trong java.
* Access modifier trong java
* Các tính chất của lập trình hướng đối tượng (OOP).
* Các khái niệm Java OOPs.
* Collection trong java.
* Xử lý ngoại lệ trong java.

### Bài 1: Làm quen với lập trình hướng đối tượng

**Đề bài**: Viết chương trình quản lý sinh viên. Mỗi đối tượng sinh viên có các thuộc tính sau: id, name, age, address và gpa (điểm trung bình). Yêu cầu: tạo ra một menu với các chức năng sau:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
1. Add student.  
2. Edit student by id.  
3. Delete student by id.  
4. Sort student by gpa.  
5. Sort student by name.  
6. Show student.  
0. Exit.  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**Bài 1: Giải phương trình bậc 2 trong Java**



Kiến thức cơ bản:

Sử dụng Scanner để nhập giá trị từ bàn phím

**private static Scanner scanner = new Scanner(System.in);**

Kiến thức toán học về phương trình bậc 2.

Kết quả mong muốn:

Nhập hệ số bậc 2, a = 2

Nhập hệ số bậc 1, b = 1

Nhập hằng số tự do, c = -1

Phương trình có 2 nghiệm là: x1 = 0.5 và x2 = -1.0

**Bài 2: Tính giai thừa trong Java (không sử dụng đệ quy)**

**n!**

Kiến thức cơ bản:

Sử dụng Scanner để nhập giá trị từ bàn phím

**private static Scanner scanner = new Scanner(System.in);**

Kiến thức toán học về giai thừa: n! = 1\*2\*…\*(n-1)\*n

Kết quả mong muốn:

Giai thừa của 5 là: 120

Giai thừa của 0 là: 1

Giai thừa của 10 là: 3628800

**Bài 3: Chuyển đổi hệ cơ số 10 sang hệ cơ số B (2<= B <= 16)**

Kiến thức cơ bản:

Sử dụng Scanner để nhập giá trị từ bàn phím

**private static Scanner scanner = new Scanner(System.in);**

Kiến thức toán học về hệ cơ số:

Hệ cơ số 2: là hệ nhị phân, số được biểu diễn bởi 2 ký tự 0 và 1

Hệ cơ số 16: là hệ cơ số 16, số được biểu diễn bởi các ký tự 0 ~ 9, A, B, C, D, E, F

Kết quả mong muốn:

So 15 trong he co so 2 = 1111

So 15 trong he co so 16 = F

**Bài 4: Viết chương trình tìm số Fibonacci thứ n**

Kiến thức cơ bản:

Sử dụng Scanner để nhập giá trị từ bàn phím

**private static Scanner scanner = new Scanner(System.in);**

Dãy số Fibonacci