

TÀI LIỆU THỰC TẬP LẬP TRÌNH MẠNG: LAB 01

DANH MỤC THUẬT NGỮ TIẾNG ANH

Từ	Nghĩa của từ
abstract	Trừu tượng
break	Dừng vòng lặp
catch	Từ khóa đầu của một khối bắt ngoại lệ
continue	Bỏ qua phần cuối vòng lặp, tiếp tục sang bước tiếp theo
default	Giá trị mặc định của phương thức switch()
extends	Kế thừa
final	Một hằng số, phương thức hay một lớp không được ghi đè
finally	Một phần của khối xử lý ngoại lệ try luôn được thực hiện
implements	Thực hiện giao diện
import	Khai báo một gói thư viện
instanceof	Kiểm tra một đối tượng là một thể hiện của lớp
interface	Giao diện
new	Tạo một đối tượng mới của lớp
null	Tham chiếu rỗng
package	Gói
private	Tiền tố chỉ được truy cập bởi phương thức của lớp
protected	Tiền tố được truy cập bởi phương thức của lớp, lớp con của và các lớp khác trong cùng một gói
public	Tiền tố có thể được truy cập bởi phương thức của tất cả các lớp
return	Trả về của một phương thức
super	Gọi phương thức của lớp cha
synchronized	Đồng bộ
this	Tham chiếu đến đối tượng hiện tại

DANH MỤC CHỮ VIẾT TẮT

Chữ viết tắt	Ý nghĩa
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
IP	Internet Protocol
URL	Uniform Resource Locator
CSDL	Cơ Sở Dữ Liệu
JDBC	Java Database Connectivity
CNTT	Công Nghệ Thông Tin
HĐH	Hệ Điều Hành
MVC	Model-View-Control
DNS	Domain Name System
API	Application Programming Interface
FTP	File Transfer Protocol
JDK	Java Development Kit
GB	GigaByte
UCLN	Ước Chung Lớn Nhất
BCNN	Bội Chung Nhỏ Nhất
RAM	Random Access Memory
RMI	Remote Method Invocation
JVM	Java Virtual Machine
NIC	Network Interface Card
ĐH KTKT CN	Đại học Kinh tế Kỹ thuật Công nghiệp

LỜI NÓI ĐẦU

Ngày nay do nhu cầu thực tế và do sự phát triển mạnh mẽ của nhiều công nghệ tích hợp, dẫn đến các chương trình ứng dụng hầu hết đều có khả năng thực hiện trên môi trường mạng. Ngôn ngữ JAVA là ngôn ngữ phù hợp để viết các ứng dụng mạng. So với lập trình thông thường, lập trình mạng đòi hỏi người lập trình hiểu biết và có kỹ năng tốt để viết các chương trình giao tiếp và trao đổi dữ liệu giữa các máy tính với nhau.

Để hỗ trợ sinh viên chuyên ngành CNTT trong nhà trường tiếp cận với kỹ thuật lập trình mới này, tiếp theo cuốn tài liệu học tập lý thuyết “**Công nghệ JAVA**”, chúng tôi xây dựng cuốn “**Bài tập lập trình mạng**”, nhằm cung cấp cho sinh viên những kiến thức và kỹ thuật cơ bản nhất để phát triển các chương trình ứng dụng mạng, thông qua các dạng bài tập từ cơ bản đến nâng cao qua các chủ đề: lập trình cơ bản, lập trình hướng đối tượng, lập trình CSDL JDBC, lập trình mạng dùng socket, lập trình phân tán với RMI. Sinh viên sẽ thực hiện các bài thực hành này trên phòng máy nhà trường.

Nội dung cuốn tài liệu bao gồm 12 bài lab chia thành các chủ đề khác nhau. Trong mỗi chủ đề chúng tôi đưa ra tóm tắt lý thuyết, bài tập mẫu, sau đó là bài tập tương tự, và bài tập tổng hợp. Kết quả qua những bài lab, sinh viên được rèn và thành thạo các kỹ năng lập trình hướng đối tượng, lập trình CSDL, lập trình với giao thức truyền thông có sẵn và khả năng tích hợp trong các ứng dụng khác nhau, nhất là các giao thức truyền thông thời gian thực, từ đó sinh viên có thể viết được các phần mềm quản lý theo mô hình MVC, xây dựng được các ứng dụng mạng, các ứng dụng tích hợp và triệu gọi lẫn nhau trên mạng Intranet (mạng cục bộ), mạng Internet (mạng toàn cầu), các hệ thống xử lý truy xuất dữ liệu phân tán hoàn chỉnh. Nội dung biên soạn phù hợp với chuẩn đầu ra của ngành CNTT và ngành mạng máy tính và truyền thông dữ liệu về kỹ năng và kiến thức. Sau khi học xong học phần này sinh viên có thể viết phần mềm quản lý, truyền thông.

Chúng tôi xin chân thành cảm ơn Thầy Nguyễn Hoàng Chiến, phó chủ nhiệm khoa, phụ trách khoa CNTT trường ĐH KTKT CN cùng với các đồng nghiệp đã đóng góp ý kiến cho cuốn tài liệu này. Vì tài liệu được biên soạn lần đầu, chúng tôi đã cố gắng hoàn chỉnh, song không tránh khỏi thiếu sót. Rất mong nhận được sự góp ý của bạn đọc để tài liệu học tập được hoàn thiện hơn.

Xin trân trọng cảm ơn!

Nhóm tác giả

LAB 1. LẬP TRÌNH CƠ BẢN [1, 2]

A. MỤC TIÊU

Trang bị cho sinh viên kỹ năng lập trình cơ bản trong java: cấu trúc **if**, vòng lặp **while...do**, **for..**, khai báo mảng, xử lý trên mảng.

B. NỘI DUNG

- Lệnh rẽ nhánh (**if**), lệnh lựa chọn (**switch**).
- Lệnh lặp, sử dụng mảng để thao tác trên các phần tử cùng kiểu.

C. YÊU CẦU PHẦN CỨNG, PHẦN MỀM

- Máy tính cài HĐH windows, RAM tối thiểu 1 GB.
- Phần mềm NETBEAN IDE 8.0, JDK 1.8.

D. KẾT QUẢ SAU KHI HOÀN THÀNH

Sinh viên thành thạo các câu lệnh : rẽ nhánh, lựa chọn, lệnh lặp, thao tác trên mảng, áp dụng giải các bài tập từ đơn giản đến phức tạp.

E. HƯỚNG DẪN CHI TIẾT

1. Cấu trúc rẽ nhánh **if...else**

```
if (boolean_expr){ //Nếu boolean_expression là true
    //do task1
}
else { //Nếu boolean_expression là false
    //do task2
}
```

Bài 1. Tìm max, min của 2 số nguyên nhập vào từ bàn phím

Hướng dẫn:

```
public class Bai1 {
    public static void main(String[] args) {
        int a,b,max,min;
        Scanner sc = new Scanner(System.in); //Phím tắt CTRL + Space
        System.out.println("Nhap so nguyen a:");
        a = sc.nextInt();
        System.out.println("Nhap so nguyen b:");
        b = sc.nextInt();
        max = (a>b)?a:b;
        min = (a>b)?b:a;
    }
}
```

```

        System.out.println("Max la:"+max);
        System.out.println("Min la:"+min);
    }
}

```

Bài 2. Giải phương trình bậc 2 $ax^2 + bx + c = 0$

Hướng dẫn:

Sử dụng Scanner nhập 3 số nguyên a, b, c. Tính $\Delta = b^2 - 4ac$.

Sử dụng **if** kiểm tra từng trường hợp của Δ .

```

if(delta<0)
    System.out.println("PT vo nghiem");
else if(delta==0){
    float x = (float)-b/(2*a);
    System.out.printf("PT co nghiem kep x1=x2=%.2f",x);
}else{
    float x1 = (float) (-b-Math.sqrt(delta))/(2*a);
    float x2 = (float) (-b+Math.sqrt(delta))/(2*a);
    System.out.printf("PT co hai nghiem x1=%.2f, x2=%.2f",x1,x2);
}

```

2. Cấu trúc switch

```

switch (controlling_expr){
case value1: //do task1
break;
case value2: //do task2
break;
...
case valueN: //do taskN
break;
default:

```

Bài 3. Viết chương trình java cho phép tạo và thực hiện theo menu sau:

1. Nhập vào một số nguyên dương n.
2. Tính tổng các số từ 1 đến n
3. Kiểm tra n có là số nguyên tố
4. Kiểm tra n có là số hoàn hảo.
5. Hiển thị số n thành tích các thừa số nguyên tố.

6. Thoát

(Hiển thị 1 số nguyên dương thành tích các thừa số nguyên tố: $n = 24$ in ra: $n = 2^3 \times 3$)

Hướng dẫn:

Phương thức nhập: Dùng Scanner

Phương thức kiểm tra hoàn hảo

```
boolean sohh(int a){
    int tong=0;
    for (int i = 1; i < a; i++) {
        if(a%i==0)
            tong+=i;
    }
    if(tong==a)
        return true;
    else
        return false;
}
```

Phương thức hiển thị số hoàn hảo

```
public void hienThisHH(int a){
    for (int i = 0; i < a; i++) {
        if(sohh(i)==true)
            System.out.print(i+" ");
    }
}
```

Phương thức kiểm tra nguyên tố

```
boolean songt(int a){
    if (a < 2)
        return false;
    for (int i = 2; i <= Math.sqrt(a); i ++){
        if (a%i==0)
        {
            return false;
        }
    }
}
```

```

    }
    return true;
}

```

Phương thức hiển thị số nguyên tố

```

public void hienThisNT(int a){
    for (int i = 1; i < a; i++) {
        if(songt(i)==true){
            System.out.print(i + " ");
        }
    }
}

```

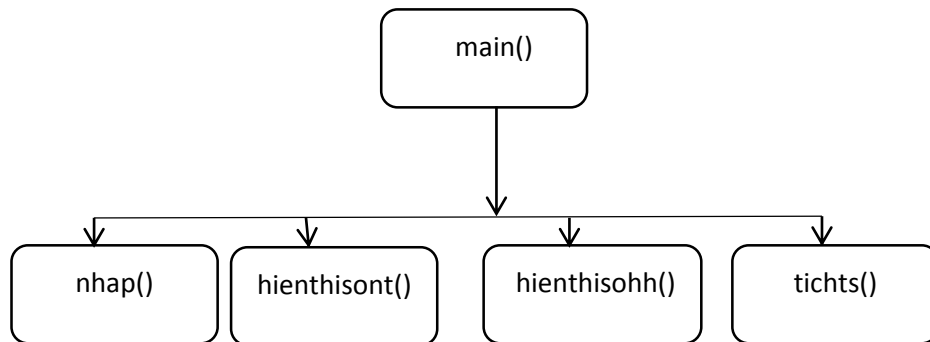
Phương thức phân tích thành thừa số

```

void tichthuaso(int a){
    int i=2;
    int dem=0;
    System.out.println("Tich thua so");
    while(a!=1){
        if(a%i==0){
            a/=i;
            dem++;
        }
        else {
            if(dem==0 || dem==1)
                System.out.print(i+"*");
            else
                System.out.print(i+"*"+dem+"*");
            i++;
            dem=0;
        }
    }
}

```

Viết phương thức main dùng **switch case** để tạo menu. Tổ chức chương trình theo sơ đồ sau:



3. Vòng lặp while, do...while, for

Dạng 1

```
while(boolean_expr){
//do something
}
```

Dạng 2

```
do {
//do something
} while(boolean_expr);
```

Bài 4. Tìm UCLN, BCNN của 2 số được nhập vào từ bàn phím.

Hướng dẫn:

```
public class Bai4{
    int a;
    int b;
    public int getA() {
        return a;
    }
    public int getB() {
        return b;
    }
    void nhap() {
        Nội dung phương thức nhập.
    }
    int USCLN(int a, int b) {
        while (a != b) {
            if (a > b) {
```



```

        a = a - b;
    } else {
        b = b - a;
    }
}
return a;
}
int BSCNN(int a,int b){
    return a*b/USCLN(a, b);
}
public static void main(String[] args) {
    Bai4 bai=new Bai4();
    bai.nhap();
    System.out.println("UCLN:" + bai.USCLN(bai.getA(), bai.getB()));
    System.out.println("BCNN :" + bai.BSCNN(bai.getA(), bai.getB()));
}
}

```

4. Mảng

Khai báo: `DataType[] array = new DataType[size];`

`DataType[] array = {value1, value2,..., valueN};`

Bài 5. Nhập dãy n phần tử. Xóa các phần tử có giá trị bằng x nhập từ bàn phím.

Hướng dẫn:

```

import java.util.Scanner;
public class Delete{
    public static void main(String[] args){
        int n, x, flag = 1, loc = 0;
        Scanner s = new Scanner(System.in);
        System.out.print("Enter no. of elements you want in array:");
        n = s.nextInt();
        int a[] = new int[n];
        System.out.println("Enter all the elements:");
        for (int i = 0; i < n; i++){
            a[i] = s.nextInt();
        }
    }
}

```

```

        System.out.print("Enter the element you want to delete:");
        x = s.nextInt();
        for (int i = 0; i < n; i++){
            if(a[i] == x){
                flag =1;
                loc = i;
                break;
            }
            else{
                flag = 0;
            }
        }
        if(flag == 1){
            for(int i = loc+1; i < n; i++) {
                a[i-1] = a[i];
            }
            System.out.print("After Deleting:");
            for (int i = 0; i < n-2; i++){
                System.out.print(a[i]+",");
            }
            System.out.print(a[n-2]);
        }
        else{
            System.out.println("Element not found");
        }
    }
}

```

Bài 6. Nhập một mảng nguyên từ bàn phím

- Sắp xếp và in lại dãy đã sắp ra màn hình.
- In ra màn hình phần tử có giá trị nhỏ nhất
- Tính trung bình cộng các phần tử chia hết cho 3.

Hướng dẫn:

- Sử dụng `Array.sort` (mang) để sắp xếp, sau đó dùng vòng lặp **for** duyệt từng phần tử in ra màn hình.

- Cho số nhỏ nhất là số đầu tiên, sau đó so sánh số nhỏ nhất đó với các số còn lại nếu số đó được so sánh nhỏ hơn thì lấy số đó làm số nhỏ nhất.
Min=Math.min(Min,a[i])
- Duyệt mảng kiểm tra từng phần tử, nếu phần tử thứ i chia hết cho 3:
if(A[i] %3==0) thì thực hiện cộng vào tổng, tăng số đếm các số chia hết cho 3.
- Lấy tổng chia cho số đếm được kết quả

Bài 7. Nhập mảng có n phần tử các số nguyên.

- In ra các phần tử là số nguyên tố của mảng.
- In ra các phần tử là số hoàn hảo của mảng.
- In ra các phần tử là số chẵn.

Hướng dẫn:

Khai báo mảng n phần tử nguyên. Sau đó sử dụng các phương thức sau:

Phương thức kiểm tra hoàn hảo: tương tự các bài trước.

Phương thức kiểm tra chẵn/lẻ

```
public static boolean kiemTraChan(int n){
    if(n % 2 == 0){
        return true;
    }
    else{
        return false;
    }
}
```

Bài 8. Nhập mảng có n phần tử các số nguyên, sắp xếp mảng tăng dần bằng thuật toán chèn và hiển thị mảng đã sắp xếp ra màn hình.

Hướng dẫn:

```
import java.util.Arrays;
public class InsertionSort {
    void InsertionSort(int[] nums){
        for(int i = 1; i < nums.length; i++){
            int value = nums[i];
            int j = i - 1;
            while(j >= 0 && nums[j] > value){
                nums[j + 1] = nums[j];
                j = j - 1;
            }
        }
    }
}
```

```

        nums[j + 1] = value;
    }
}

public static void main(String args[]){
    InsertionSort ob = new InsertionSort();
    int nums[] = {7, -5, 3, 2, 1, 0, 45};
    System.out.println("Original Array:");
    System.out.println(Arrays.toString(nums));
    ob.InsertionSort(nums);
    System.out.println("Sorted Array");
    System.out.println(Arrays.toString(nums));
}
}

```

Bài 9.

Dãy số Fibonacci được định nghĩa như sau: $F_0 = 1$; $F_1 = 1$; $F_n = F_{n-1} + F_{n-2}$ với $n \geq 2$. Nhập một số tự nhiên n .

- Hãy viết chương trình tìm số Fibonacci thứ n .
- Hãy liệt kê các số Fibonacci nhỏ hơn n là số nguyên tố.

Hướng dẫn:

Viết phương thức `ktnt (int n)`: tương tự các bài trên

Viết phương thức trả về số fibonacci thứ n

```

int fibonacci(int n) {
    if (n < 2) {
        return n;
    }
    return fibonacci(n - 1) + fibonacci(n - 2);
}

```

Trong phương thức `main()`, nhập số nguyên n , gọi phương thức `fibonacci(int n)` để tìm số Fibonacci thứ n .

Sử dụng vòng lặp để in ra các số Fibonacci nhỏ hơn n và là số nguyên tố

```

for (int i = 0; i < n; i++){
    if (ktnt(i) == true) {
        for (int j = 0; j < n; j++) {
            if (i == fibonacci(j)) {

```

```

        System.out.print(i + " ");
    }
}
}
}

```

Bài 10. Viết chương trình tìm điểm trung bình của hai học sinh trong ba môn học.

Cho điểm của học sinh thứ nhất là 60, 55 và 70, điểm của học sinh thứ hai là 80, 60 và 41

Hướng dẫn: Lưu trữ điểm của hai học sinh trong một mảng 2 chiều có 2 hàng và 3 cột. Các hàng sẽ đại diện cho học sinh và các cột lưu điểm của học sinh.

```

class Array2DExample {
    public static void main(String args[]) {
        int[][] score = {
            {60, 55, 70},
            {80, 60, 41}
        };

        int[] sum = new int[2];    // tổng mảng
        sum[0] = 0;    // tổng của sinh viên 1
        sum[1] = 0;    // tổng của sinh viên 2

        float[] avg = new float[2]; // trung bình mảng
        for (int r = 0; r < 2; r++) {
            for (int c = 0; c < 3; c++) {
                sum[r] += score[r][c];
            }
        }

        avg[0] = (float)sum[0] / 3; // trung bình của sinh viên 1
        avg[1] = (float)sum[1] / 3; // trung bình của sinh viên 2
        System.out.println("Average score of 1st student = " + avg[0]);
        System.out.println("Average score of 2nd student = " + avg[1]);
    }
}

```

TÀI LIỆU THAM KHẢO

- [1]. Cay S. Horstmann, *Core Java Volum I - Fundamentals, Tenth Edition*, NewYork : Prentice Hall, 2016.
- [2]. Cay S. Horstmann. *Core Java Volum II - Advanced Features, Tenth Edition*, New York : Prentice Hall, 2017.
- [3].Eng.haneen Ei-masry, *Java database connection*, Islamic University of Gaza Faculty of Engineering Department of Computer Engineering ECOM 4113: DataBase Lab, 2014.
- [4]. Angelos Stavrou, *Advanced Network Programming Lab using Java*, Network Security, ISA 656, Angelos Stavrou.
- [5]. Marenglen Biba, Ph.D, *Manual for Lab practices, Remote Method Invocation Three Tier Application with a Database Server*, Department of Comsputer Science, University of New York.
- [6].Elliotte Rusty Harold, *Java Network Programming, Fourth Edition*, O'Reilly Media, 2013.
- [7]. Đoàn Văn Ban, *Lập trình hướng đối tượng với JAVA*, Nhà xuất bản Khoa học và Kỹ thuật, 2005.
- [8]. ThS. Dương Thành Phết, *Bài tập thực hành Chuyên đề 1 CNPM- Java*, Khoa CNTT- Trường ĐH Công nghệ TP.HCM.
- [9]. <https://www.oracle.com/technetwork/java/socket-140484.html#>
- [10]. https://personales.unican.es/corcuerp/java/Labs/LAB_22.htm
- [11]. <http://www.nrcmec.org/pdf/Manuals/CSE/student/2-2%20java16-17.pdf>
- [12]. <http://cse.mait.ac.in/pdf/LAB%20MANUAL/JAVA.pdf>
- [13]. https://www.academia.edu/35283541/Bài_tập_môn_lập_trình_hướng_đối_tượng