**LAB 1. LẬP TRÌNH CƠ BẢN**

1. **MỤC TIÊU**

Trang bị cho sinh viên kỹ năng lập trình cơ bản trong java: xử lý nhập dữ liệu dùng lớp Scanner, cấu trúc if, vòng lặp while…do, for.., khai báo mảng, xử lý trên mảng.

1. **NỘI DUNG**

* Sử dụng NETBEAN để tạo dự án
* Xử lý nhập, xuất dữ liệu
* Lệnh rẽ nhánh (if), lệnh lựa chọn (switch).
* Lệnh lặp, sử dụng mảng để thao tác trên các phần tử cùng kiểu

1. **YÊU CẦU PHẦN CỨNG, PHẦN MỀM**

- Máy tính cài HĐH windows, RAM tối thiểu 256MB.

- NETBEAN IDE 8.0, JDK 1.8

**D.KẾT QUẢ SAU KHI HOÀN THÀNH**

- Sinh viên thành thạo các câu lệnh nhập, xuất, 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**

- **Nhập dữ liệu**: Sử dụng Lớp Scanner

Khai báo đối tượng Scanner nhap= new Scanner (System.in) sau đó tiến hành nhập dữ liệu dựa trên đối tượng nhap để nhập số thực, số nguyên..

**Bài 1.** Nhập 1 số nguyên, 1 số double, 1 ký tự, 1 chuỗi từ bàn phím và hiển thị giá trị các biến vừa nhập ra màn hình.

**Hướng dẫn:**

|  |
| --- |
| import java.util.Scanner;  public class Bai1 {  public static void main(String[] args) {  int a;  float f;  double d;  long l;  char ch;  String str;  Scanner sc = new Scanner(System.in);  System.out.println("Nhap mot so nguyen:");  a = sc.nextInt();  System.out.println("Nhap mot so thuc:");  f = sc.nextFloat();  System.out.println("Nhap mot so double:");  d = sc.nextDouble();  System.out.println("Nhap mot so long:");  l = sc.nextLong();  System.out.println("Nhap mot ki tu:");  sc.nextLine(); // ki tu ket thuc nhap se bi mat  ch = sc.nextLine().charAt(0);  System.out.println("Nhap mot chuoi:");  str = sc.nextLine();  System.out.println("Du lieu vua nhap:");  System.out.println("a="+a);  System.out.println("l="+l);  System.out.println("f="+f);  System.out.println("ch="+ch);  System.out.println("str="+str);  System.out.println("d="+d);  System.out.println("\nIn tren cung 1 dong:");  System.out.println("a="+a+", f="+f+", l="+l+", d="+d+", ch="+ch+", str="+str);  }  } |

**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 2.** Tìm max, min của 2 (hoặc 3) số được nhập vào từ bàn phím

**Hướng dẫn**

|  |
| --- |
| public class Bai2 {  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 3.** Giải phương trình bậc 2ax^2 + bx +c = 0.

**Hướng dẫn**

|  |
| --- |
| public class PTBH {  public static void main(String[] args) {  int a, b, c, delta;  Scanner sc = new Scanner(System.in);  do{  System.out.println("Nhap he so a:");  a = sc.nextInt();  if(a==0)  System.out.println("Nhap lai a!");  }while(a==0);  System.out.println("Nhap he so b:");  b = sc.nextInt();  System.out.println("Nhap he so c:");  c = sc.nextInt();  delta = b\*b-4\*a\*c;  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.println("PT co 2 nghiem phan biet:");  System.out.printf("\tx1=%.2f \n",x1);  System.out.printf("\tx2=%.2f \n",x2);  }  }  } |

**Bài 4.** Viết chương trình tính tiền điện nhập vào số điện sử dụng của tháng, tính tiền theo phương pháp lũy tiến

Nếu số điện sử dụng từ 0 đến 50 giá mỗi số điện là 5000

Trên 50 số, mỗi số điện vượt mức là 1200

**Hướng dẫn:**

Tương tự bài 3, sử dụng cấu trúc if kiểm tra

Nếu số điện <50: Tiền điện =số điện \*5000

Ngược lại: Tiền điện = 50\*5000+(số điện – 50)\*1200.

**Cấu trúc switch**

|  |
| --- |
| switch (controlling\_expr){  case value1: //do task1  break;  case value2: //do task 2  break;  ...  case valueN: //do taskN  break;  default: //do other task |

**Bài 5.** Viết chương trình java tạo và thực thi theo menu sau:

1. Nhập vào số nguyên dương n
2. Tính tổng: 1 + 1/2 + 1/3 + ... + 1/n
3. Tính tổng: 1 + 1/2! + 1/3! + ... + 1/n!
4. Thoát.

**Hướng dẫn:**

|  |
| --- |
| public class Bai5 {  int n;  public int getN() {  return n;  }  void nhap(){  boolean check = false;  Scanner sc = new Scanner(System.in);  do {  try {  System.out.println("Nhap vao n:");  n = Integer.parseInt(sc.nextLine());  check = true;  } catch (java.lang.NumberFormatException e) {  System.out.println("hay nhap dung kieu du lieu" + e.toString());  check = false;  }  } while (check == false);  }  float tinhTong1(int a){  float tong1=0;  for (int i = 1; i <= a; i++) {  tong1+=1.0/i;  }  return tong1;  }  float tinhTong2(int a){  float tong2=0;  int giaithua=1;  for (int i = 1; i <= a; i++) {  giaithua\*=i;  tong2+=1.0/giaithua;  }  return tong2;  }  void menu(){  System.out.println("1 Nhap vao so nguyen duong n");  System.out.println("2 Tinh tong :1+....+1/n");  System.out.println("3 Tinh tong :1+....+1/n!");  }  public static void main(String[] args) {  Bai5 bai5=new Bai5();  int n=0;  Scanner sc = new Scanner(System.in);  do {  boolean check = false;  do {  bai5.menu();  try {  System.out.println("Nhap vao lua chon :");  n = Integer.parseInt(sc.nextLine());  check = true;  } catch (java.lang.NumberFormatException e) {  System.out.println("hay nhap dung kieu du lieu" + e.toString());  check = false;  }  } while (check == false);  switch (n) {  case 1: {  bai5.nhap();  break;  }  case 2: {  if(bai5.getN()==0){  System.out.println("ban chua nhap n!hoac ban de n =0");  }  else  System.out.println("Tong 1+...1/"+n+" la:"+bai5.tinhTong1(bai5.getN()));  break;  }  case 3: {  if(bai5.getN()==0){  System.out.println("ban chua nhap n!hoac ban de n =0");  }  else  System.out.println("Tong 1+...1/"+n+"! la:"+bai5.tinhTong2(bai5.getN()));  break;  }  case 0: {  break;  }  default: {  System.out.println("Khong co lua chon cua ban ");  break;  }  }  } while (n != 0);  }  } |

**Bài 6.** Viết chương trình java cho phép tạo và thực thi 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. Thoát

**Hướng dẫn:**

**Hàm 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;  } |

**Hàm 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+" ");  }  } |

**Hàm 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;  } |

**Hàm 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 +" ");  }  }  } |

**Hàm tính tổng**

|  |
| --- |
| int tong(int a){  int tong1=0;  for (int i = 0; i < a; i++) {  tong1+=i;  }  return tong1;  } |

**Hàm phân tích thành thừa số**

|  |
| --- |
| void tichthuaso(int a){  int i=2;  int dem=0;  System.out.println("ich 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;  }  }  } |

Cuối cùng viết hàm main, nhập số nguyên dùng Scanner. Dùng switch case để tạo menu tương tự bài 5

**while, do…while, for**

|  |
| --- |
| Dạng 1  while(boolean\_expr){  //do something  }  Dạng 2  do {  //do something  } while(boolean\_expr); |

**Bài 7.** Tìm UCLN, BCNN của 2 số được nhập vào từ bàn phím.

**Hướng dẫn:**

|  |
| --- |
| public class Bai7 {  int a;  int b;  public int getA() {  return a;  }  public int getB() {  return b;  }  void nhap() {  boolean check = false;  Scanner sc = new Scanner(System.in);  do {  try {  System.out.println("Nhap vao a:");  a = Integer.parseInt(sc.nextLine());  check = true;  } catch (java.lang.NumberFormatException e) {  System.out.println("hay nhap dung kieu du lieu" + e.toString());  check = false;  }  } while (check == false);  do {  try {  System.out.println("Nhap vao b:");  b = Integer.parseInt(sc.nextLine());  check = true;  } catch (java.lang.NumberFormatException e) {  System.out.println("hay nhap dung kieu du lieu" + e.toString());  check = false;  }  } while (check == false);  }  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) {  Bai7 bai7=new Bai7();  bai7.nhap();  System.out.println("Uoc chung lon nhat :" + bai7.USCLN(bai7.getA(), bai7.getB()));  System.out.println("Boi so chung nho nhat :" + bai7.BSCNN(bai7.getA(), bai7.getB()));  }  } |

**Bài 8.** Nhập một số nguyên n

a) Kiểm tra 1 số có là số nguyên tố hay không?

b) In ra các số nguyên tố từ 1 đến n

**Hướng dẫn**: Tương tự, sử dụng hàm kiểm tra nguyên tố trong bài 6.

Viết hàm nhập sử dụng do…while như sau

|  |
| --- |
| void nhap() {  boolean check = false;  Scanner sc = new Scanner(System.in);  do {  try {  System.out.println("Nhap vao n:");  n = Integer.parseInt(sc.nextLine());  check = true;  } catch (java.lang.NumberFormatException e) {  System.out.println("hay nhap dung kieu du lieu" + e.toString());  check = false;  }  } while (check == false);  } |

Viết hàm main() gọi hàm nhập và hàm kiểm tra nguyên tố.

**Xử lý trên mảng**

1. Mảng là tập hợp hữu hạn các phần tử cùng kiểu
2. Số lượng phần tử xác định khi khai báo, không đổi
3. Cú pháp DataType[] array = new DataType[size];

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

**Một số phương thức xử lý trên mảng**

* Phương thức length(): Lấy số phần tử của mảng
* Truy cập vào phần tử của mảng 1 chiều: array[index], index: chỉ số, bắt đầu từ 0
* Truy cập vào phần tử của mảng nhiều chiều array[index1][index2]...[indexN]

**Bài 9**. Nhập một dãy n số nguyên, tìm xem trong dãy đã nhập có phần tử nào trùng với phần tử nhập từ bàn phím. Nếu có: in ra vị trí, ngược lại in ra thông báo không tìm thấy.

**Hướng dẫn:**

|  |
| --- |
| import java.util.Scanner;  public class Search {  public static void main (String[] args) {  int size = 0;  Scanner inputData = new Scanner(System.in);  System.out.print(“Enter the size of the array:”);  size = inputData.nexInt;  int[] intArr = new int[size];  }  System.out.println(“Enter value for array:);  for(int i = 0; i < size; i++){  System.out.print(“intArr[“ + i + “] = ”);  intArr[i] = inputData.nextInt();  }  int key = 0;  System.out.println(“Enter value for array:);  key = inputData.nextInt();  boolean found = false;  for(int i = 0; i < size; i++){  if (intArr[i] == key){  System.out.println(“Found ” + key +“at ” + i);  found = true;  break;  }  }  if(!found)  System.out.println (“Could not find “ + key);  }  } |

**Bài 10.** Nhập một mảng nguyên từ bàn phím

1. Sắp xếp và in lại dãy đã sắp ra màn hình.
2. In ra màn hình phần tử có giá trị nhỏ nhất
3. 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 (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 11**. 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 12.** Nhập mảng có n phần tử các số nguyên.

1. In ra các phần tử là số nguyên tố của mảng.
2. In ra các phần tử là số hoản hảo của mảng.
3. In ra các phần tử là số chẵn.

**Hướng dẫn:**

* Khai báo mảng n phần tử nguyên. Sau đó sử dụng các hàm kiểm tra nguyên tố, hoàn hảo, chẵn lẻ.

**Hàm kiểm tra nguyên tố:** tương tự các bài trước.

**Hàm kiểm tra hoàn hảo:** tương tự các bài trước.

**Hàm kiểm tra chẵn/lẻ**

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

**Bài 13.** 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 {  //Hàm chèn một phần tử vào mảng  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 14.** Dãy số Fibonacci được định nghĩa như sau: F0 =1, F1 = 1; Fn = Fn-1 + Fn-2 với n>=2. Nhập một số tự nhiên n.

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

**Hướng dẫn:**

Viết hàm ktnt (int n): tương tự các bài trên

Viết hàm trả về số fibonacci thứ n

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

Trong hàm main(), nhập số nguyên n, gọi hàm 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 15.**

Viết chương trình nhập vào mảng A có n phần tử, các phần tử là những số nguyên lớn hơn 0 và nhỏ hơn 100 được nhập vào từ bàn phím. Thực hiện các chức năng sau:

1. Tìm phần tử lớn nhất và lớn thứ 2 trong mảng cùng chỉ số của các số đó.
2. Sắp xếp mảng theo thứ tự giảm dần.
3. Nhập một số nguyên x và chèn x vào mảng A sao cho vẫn đảm bảo tính sắp xếp giảm dần.

**Hướng dẫn:**

|  |
| --- |
| public class Bai15 {  int[] selectionSort(int[] b) {  for (int i = 0; i < b.length - 1; i++) {  for (int j = i + 1; j < b.length; j++) {  if (b[i] > b[j]) {  int tem = b[i];  b[i] = b[j];  b[j] = tem;  }  }  }  return b;  }  void printArr(int[] a) {  for (int i = 0; i < a.length; i++) {  System.out.print(a[i] + " ");  }  }  int[] chen(int[] a, int x) {  int[] b = new int[a.length + 1];  for (int i = 0; i < a.length; i++) {  b[i] = a[i];  }  for (int i = 0; i < a.length; i++) {  b[i] = a[i];  if (a[i] >= x) {  for (int j = a.length; j > i; j--) {  b[j] = a[j - 1]; //Dịch các phần tử sang phải 1 vị trí  }  b[i] = x; //Thêm x vào vị trí vt  break;  }  }  return b;  }  public static void main(String[] args) {  int n = 0;  Scanner sc = new Scanner(System.in);  System.out.println("Nhap vao n:");  n = Integer.parseInt(sc.nextLine());  int[] a = new int[n];  System.out.println("Nhap vao gia tri cua mang ");  for (int i = 0; i < n; i++) {  System.out.println("a[" + (i + 1) + "]");  a[i] = Integer.parseInt(sc.nextLine());  }  int max1 = a[0];  int max2 = Integer.MIN\_VALUE;  int vt1 = 0;  int vt2 = 0;  for (int i = 0; i < n; i++) {  if (max1 < a[i]) {  max2 = max1;  max1 = a[i];  vt2 = vt1;  vt1 = i;  } else if (a[i] > max2 && a[i] < max1) {  max2 = a[i];  vt2 = i;  }  }  System.out.println("so lon thu 1 la :" + max1 + " vi tri :" + (vt1 + 1));  System.out.println("so lon thu 2 la :" + max2 + " vi tri :" + (vt2 + 1));  System.out.println("mang sau khi sap xep :");  Bai15 bai15 = new Bai15();  bai15.printArr(bai15.selectionSort(a));  System.out.println("mang sau khi chen la :");  bai15.printArr(bai15.chen(a, x));  }  } |

**Bài 16.** Viết chương trình thực hiện chuẩn hoá một xâu ký tự nhập từ bàn phím (loại bỏ các dấu cách thừa, chuyển ký tự đầu mỗi từ thành chữ hoa, các ký tự khác thành chữ thường)

**Hướng dẫn:**

|  |
| --- |
| public class Bai16 {  public String convert(String str){  String tmp="";  tmp+=String.valueOf(str.charAt(0)).toUpperCase();  tmp+=str.substring(1); //tmp+=str.substring(1, str.length());  return tmp;  }  public static void main(String[] args) {  String str;  Bai16 bai16=new Bai16();  System.out.println("Nhap vao chuoi ki tu muon chuan hoa");  Scanner sc=new Scanner(System.in);  str=sc.nextLine();  StringTokenizer tok=new StringTokenizer(str, " ");  while(tok.hasMoreTokens()){  System.out.print(bai16.convert(tok.nextToken())+" ");  }  }  } |

**Bài 17.**  Nhập và in ra mảng hai chiều

|  |
| --- |
| import java.util.Scanner;  public class MangHaiChieu  {  public static void main(String args[])  {  int row, col, i, j;  int arr[][] = new int[10][10];  Scanner scan = new Scanner(System.in);  System.out.print("Enter Number of Row for Array (max 10) : ");  row = scan.nextInt();  System.out.print("Enter Number of Column for Array (max 10) : ");  col = scan.nextInt();  System.out.print("Enter " +(row\*col)+ " Array Elements : ");  for(i=0; i<row; i++)  {  for(j=0; j<col; j++)  {  arr[i][j] = scan.nextInt();  }  }  System.out.print("The Array is :\n");  for(i=0; i<row; i++)  {  for(j=0; j<col; j++)  {  System.out.print(arr[i][j]+ " ");  }  System.out.println();  }  }  } |

**Bài 18**. Viết chương trình in mảng hai chiều dạng bảng sau:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 |

**Hướng dẫn:**

|  |
| --- |
| public class TwoDArray  {  public static void main (String[] args)  {  int[][] table = new int[5][10];  for (int row=0; row < table.length; row++)  for (int col=0; col < table[row].length; col++)  table[row][col] = row \* 10 + col;  for (int row=0; row < table.length; row++)  {  for (int col=0; col < table[row].length; col++)  System.out.print (table[row][col] + "\t");  System.out.println();  }  }  } |

**Bài 19.** Viết chương trình Java để tìm điểm trung bình của hai học sinh trong ba bài báo.

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 2chiều có 2 hàng và 3 cột. Các hàng sẽ đại diện cho học sinh và các cột sẽ giữ 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  // tổng của sinh viên 1  sum[0] = 0;  // tổng của sinh viên 2  sum[1] = 0;  // trung bình mảng  float[] avg = new float[2];  for (int r = 0; r < 2; r++) {  for (int c = 0; c < 3; c++) {  sum[r] += score[r][c];  }  }  // trung bình của sinh viên 1  avg[0] = (float)sum[0] / 3;  // trung bình của sinh viên 2  avg[1] = (float)sum[1] / 3;  System.out.println("Average score of 1st student = " + avg[0]);  System.out.println("Average score of 2nd student = " + avg[1]);  }  } |