

BÀI THỰC HÀNH SỐ 1

Lập trình Java căn bản

I. Mục tiêu.

Hiểu và nắm vững các đặc điểm cú pháp, câu lệnh, lập trình tổng quát trong lập trình Java.

II. Thời gian

- Hướng dẫn chung và thực hành tại lớp: 5 tiết
- Bài tập về nhà: 10 tiết

III. Hướng dẫn chung

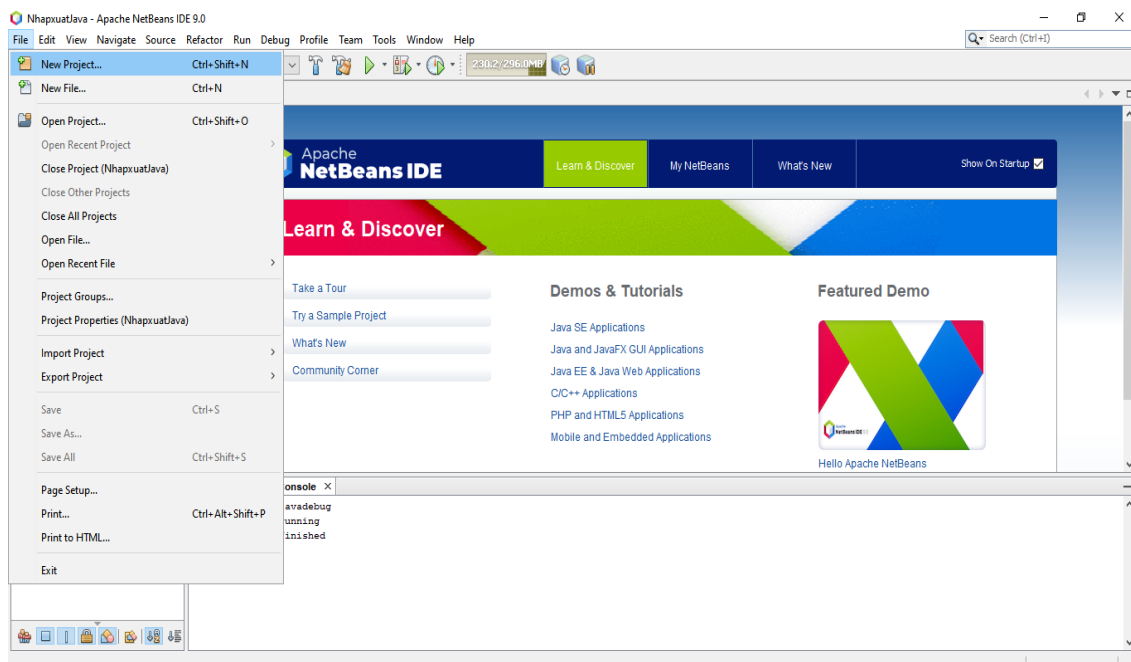
Câu 1: Tạo project mới trong Netbeans

Hướng dẫn:

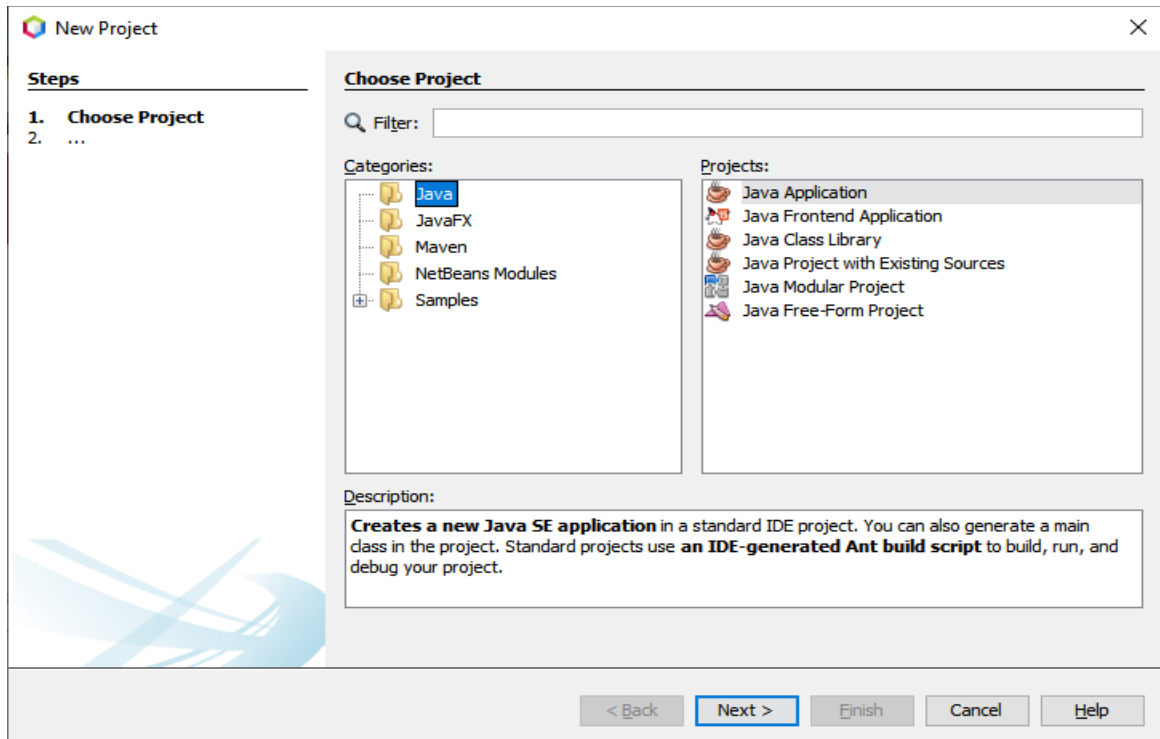
- **Bước 1:** Download Netbeans phiên bản Java SE hoặc All tại:

<https://netbeans.org/downloads/>

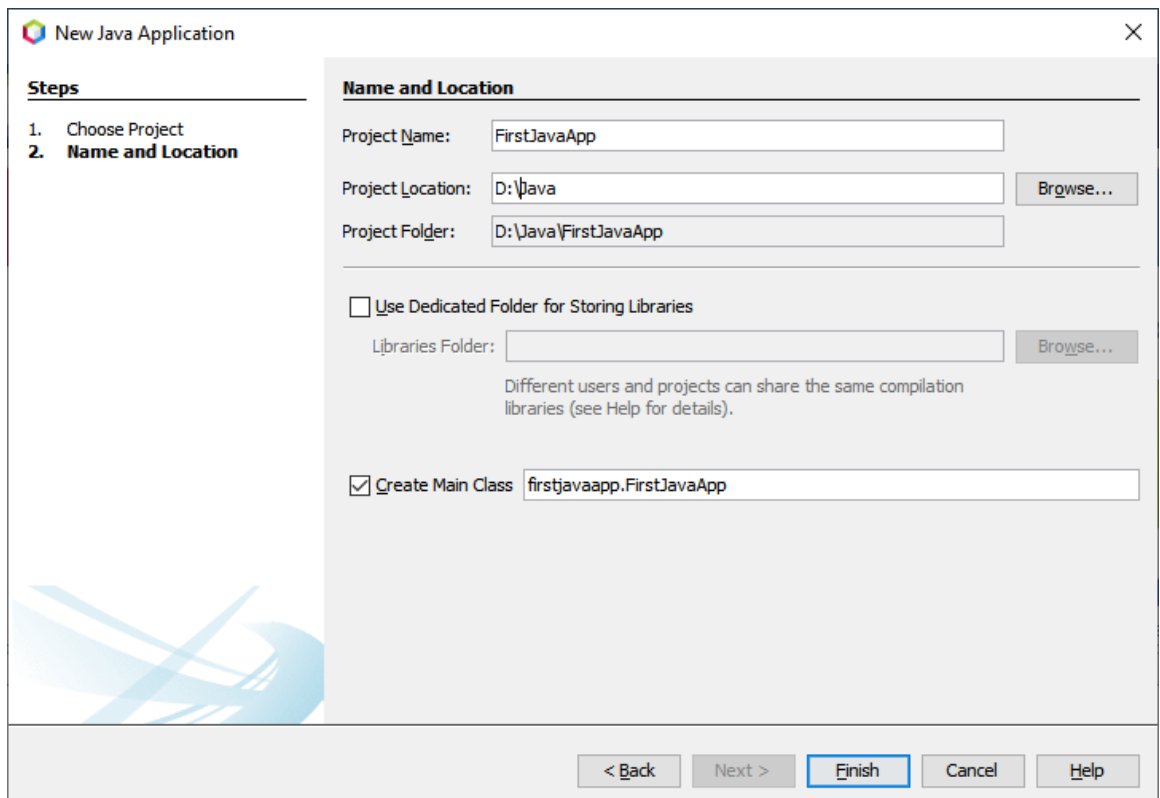
- **Bước 2:** Cài đặt Netbeans
- **Bước 3:** Mở Netbeans, chọn **File-> New Project...**



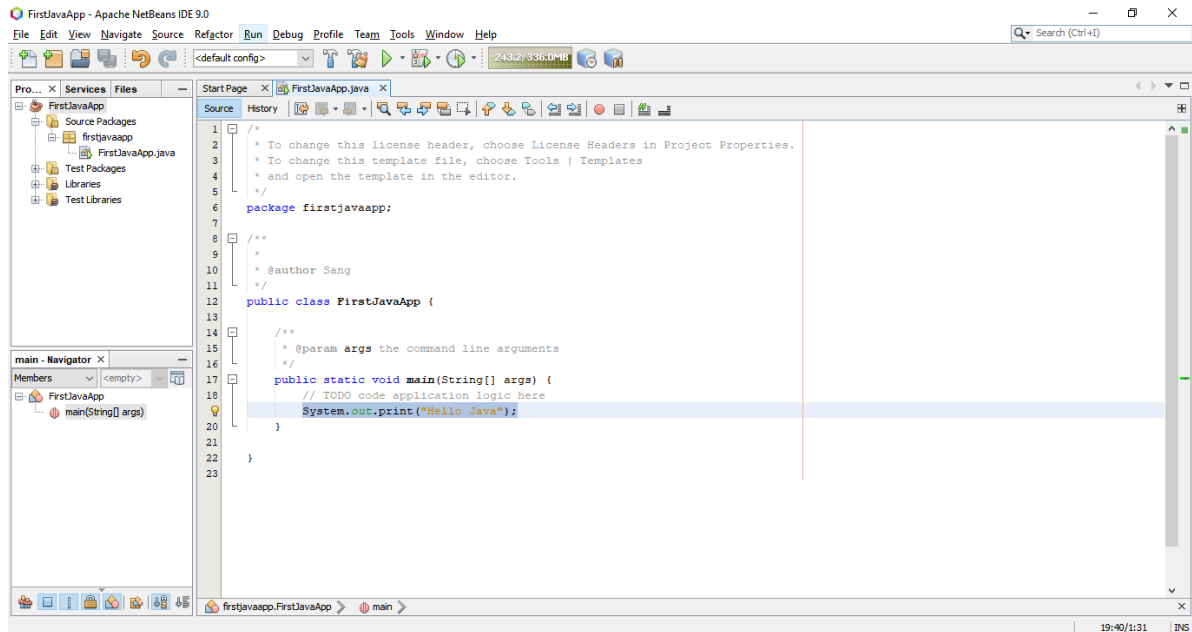
- **Bước 4:** Xuất hiện cửa sổ **New Project**, chọn **Projects: Java Application-> Next**



- **Bước 5:** Điền tên **Project Name** và chọn đường dẫn lưu ứng dụng trong cửa sổ **New Java Application**. Chọn **check** ở **Create Main Class** để tạo **class main** cho ứng dụng. Sau đó chọn **Finish**. Lưu ý: mặc định tên Main Class sẽ trùng tên với Project Name.



○ **Bước 6:** Thêm đoạn code **System.out.print("Hello Java");** vào trong hàm main. Sau đó trên thanh menu của NetBeans chọn Run-> Run Projects hoặc nhấn F6 để thực thi ứng dụng.



Câu 2: Nhập ký tự, số từ bàn phím.

Hướng dẫn: Có 3 cách sử dụng

○ **Cách 1:** sử dụng lớp **BufferedReader**

- BufferedReader được dùng để đọc dữ liệu từ bàn phím hay từ file. Có thể đọc được một chuỗi, một mảng hoặc một ký tự. Dùng để đọc dữ liệu từ bàn phím thì tham số khởi tạo đối tượng BufferedReader là InputStreamReader, hoặc dùng FileReader để đọc dữ liệu từ file.

VD: `BufferedReader nhap = new BufferedReader(new InputStreamReader(System.in));`

- Một số phương thức để đọc dữ liệu của lớp BufferedReader:

- **read():** đọc một ký tự
- **readLine():** đọc một dòng text

- Ví dụ minh họa:

```
// Tạo đối tượng BufferedReader
BufferedReader nhap = new BufferedReader(new InputStreamReader(System.in));
String ten = "";
System.out.print("Nhập tên: ");
```

```
//Nhập từ bàn phím dùng BufferedReader
try {
    ten = nhap.readLine();
} catch (IOException ex) {
    System.out.println("loi");
}
//Hiển thị tên
System.out.println("Hello " + ten + "!");
```

- Lưu ý: các phương thức để nhập dữ liệu của Bufferreader phải được đặt trong khối lệnh try... catch với lớp xử lý ngoại lệ **IOException**.

○ **Cách 2:** Sử dụng lớp **Scanner** (hay dùng)

- Scanner phân loại được dữ liệu mà người dùng nhập vào theo kiểu int hay float, double...

- Một số phương thức để đọc dữ liệu của lớp Scanner

- **next()** //dùng cho **String**
- **nextInt()**
- **nextFloat()**
- **nextBoolean()**
- **nextByte()**
- **nextLine()**

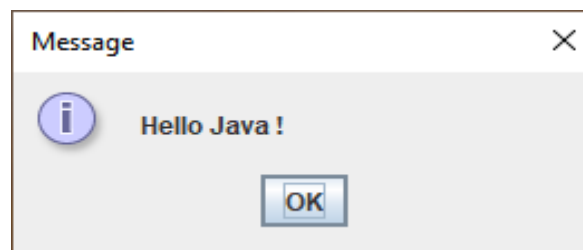
- Ví dụ minh họa:

```
// Tạo một đối tượng Scanner
Scanner scanIn=new Scanner(System.in);
String ten = ""; // Tên
int tuoi; // Tuổi
System.out.println("Nhập tên và tuổi: ");
ten = scanIn.nextLine();
tuoi = scanIn.nextInt();
System.out.println("Tên " + ten + "!");
System.out.println("Tuổi " + tuoi + "!");
```

○ **Cách 3: sử dụng JOptionPane**

- **JOptionPane** là một lớp thừa kế từ lớp **JComponent**
- Khi chạy sẽ xuất hiện một dialog box cho phép nhập dữ liệu.
- Một số phương thức của JOptionPane:
 - **showConfirmDialog()**: hiển thị một câu hỏi lựa chọn giống như yes/no/cancel.
 - **showInputDialog()**: hiển thị box nhập.
 - **showMessageDialog()**: báo cho người dùng một sự kiện vừa xảy ra.
- Ví dụ minh họa:

```
String ten = "";  
ten = JOptionPane.showInputDialog("Please enter your name");  
String msg = "Hello " + ten + " !";  
JOptionPane.showMessageDialog(null, msg);  
System.out.println("Xin chào : " +msg);
```



Câu 3: Viết chương trình cho phép nhập họ tên sinh viên và địa chỉ từ bàn phím, sau đó xuất ra màn hình họ tên và địa chỉ vừa mới nhập vào.

Hướng dẫn:

- **Bước 1:** Tạo một project mới như đã hướng dẫn ở câu 1.

- **Bước 2:** Viết mã cho hàm `main()`; sử dụng một trong các cách nhập dữ liệu từ bàn phím như đã hướng dẫn ở câu 2.
- **Bước 3:** ví dụ: dùng `Scanner` để nhập dữ liệu từ bàn phím và xuất họ tên địa chỉ ra màn hình bằng cách dùng **`System.out.println()`** hoặc **`System.out.printf()`**.

```
Scanner scanIn=new Scanner(System.in);
String hoTen;
String diaChi;
System.out.println("Nhập họ và tên: ");
hoTen = scanIn.nextLine();
System.out.println("Nhập địa chỉ: ");
diaChi = scanIn.nextLine();
System.out.println("Họ và tên sinh viên: " + hoTen);
System.out.println("Địa chỉ của sinh viên: ", diaChi);
```

IV. Thực hành tại lớp

Phần 1: Bài tập lập trình về: kiểu dữ liệu, toán tử, lệnh `if` và `switch`.

Mục tiêu: Các bài tập trong phần này giúp sinh viên có khả năng:

- Hiểu rõ được các kiểu dữ liệu của Java.
- Sử dụng thành thạo các loại toán tử của Java.
- Sử dụng thành thạo lệnh rẽ nhánh **`If`**, lệnh lựa chọn **`Switch`**.
- Xử lý chuỗi ký tự.

Bài 1: Viết hàm nhập vào bán kính của đường tròn, tính chu vi đường tròn đó.

Hướng dẫn: sử dụng hằng số `Math.PI` để tính chu vi.

Bài 2: Viết hàm Nhập vào 2 số `a`, `b`. Xuất ra màn hình kết quả `a/b` (lấy số 3 lẻ)

Hướng dẫn: sử dụng hàm `printf` để in ra. `System.out.printf("%.3f", x);`

Bài 3: Viết hàm nhập 2 phân số. Tính tổng, hiệu, tích, thương của 2 phân số nói trên.

Bài 4: Viết hàm nhập vào 2 chuỗi `x` và `y`. (Hướng dẫn: sử dụng hàm **`nextLine()`** của lớp `Scanner`). Thực hiện các thao tác sau:

- Cho biết tổng chiều dài của chuỗi `x`. (Hướng dẫn: sử dụng hàm **`length`**)
- Cho biết 3 ký tự đầu tiên của chuỗi `x`. (Hướng dẫn: sử dụng hàm **`substring`**)
- Cho biết 3 ký tự cuối của chuỗi `x`.
- Cho biết ký tự thứ 6 của chuỗi `x`. (Hướng dẫn: sử dụng hàm **`charAt`**)

- Tạo chuỗi mới gồm 3 kí tự đầu tiên của chuỗi x và 3 kí tự cuối của chuỗi y.
- Kiểm tra 2 chuỗi x, y có bằng nhau hay không (phân biệt chữ hoa, thường)? (Hướng dẫn: sử dụng hàm **equals**)
- Kiểm tra 2 chuỗi x, y có bằng nhau hay không (không phân biệt chữ hoa, h.thường)? (Hướng dẫn: sử dụng hàm **equalsIgnoreCase**)
- Cho biết y có xuất hiện trong x hay không? Nếu có, tại vị trí nào? (Hướng dẫn: sử dụng hàm **indexOf**)
- Cho biết tất cả các vị trí xuất hiện của y trong x.

Bài 5: Viết hàm nhập vào số điện sử dụng của tháng và tính tiền điện theo phương pháp lũy tiến.

- Nếu số điện ≤ 50 kw thì giá mỗi số điện là 2000 đ.
- Nếu số điện > 50 kw và ≤ 100 kw thì giá mỗi số điện vượt hạn mức 50 kw là 2500 đ.
- Nếu số điện > 100 kw thì giá mỗi số điện vượt hạn mức 50 kw là 3500 đ.

Bài 6: Viết hàm để tổ chức 1 menu để gọi các hàm đã viết ở bài 1 đến bài 5 như mẫu:

```
#####
```

1. Bài 1
2. Bài 2
3. Bài 3
4. Bài 4
5. Bài 5
6. Thoát

```
#####
```

Chọn chức năng: _

Chọn chức năng từ 1 đến 6 để gọi các hàm đã viết ở bài 1 đến 5

Hướng dẫn:

- Dùng hàm **System.out.println()** để xuất ra màn hình menu như mẫu.
- Dùng hàm **nextInt()** để nhận ký tự số từ 1 đến 6 nhập từ bàn phím tương đương với 6 chức năng trong menu.
- Sử dụng lệnh **Switch** để gọi đến các hàm tương ứng đã viết ở bài 1 đến 5.

Phần 2: Bài tập về mảng và vòng lặp.

Mục tiêu: Các bài tập trong phần này giúp sinh viên có khả năng:

- Sử dụng thành thạo các lệnh về vòng lặp
- Sử dụng thành thạo về mảng.

Bài 1: Viết chương trình nhập vào một số nguyên dương từ bàn phím và kiểm tra xem số đó có phải là số nguyên tố hay không? (Số nguyên tố là số chỉ chia hết cho 1 và chính nó).

Hướng dẫn: sử dụng vòng lặp **for**

```
ok = true;
```

```
for (int i = 2; i < n-1; i++){ //n là số nguyên dương nhập từ bàn phím
```

```
    if (n % i == 0){ // n chia hết cho i
```

```
        ok = false;
```

```
        break;
```

```
    }
```

```
}
```

Kiểm tra, nếu sau khi hết vòng lặp mà `ok = true` thì `n` là số nguyên tố

Bài 2: Viết chương trình xuất ra màn hình “Bảng tính Pythagoras”

×	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

Hướng dẫn: Sử dụng 2 vòng lặp **for** lồng nhau.

Bài 3: Nhập mảng một chiều có n phần tử, các phần tử có kiểu dữ liệu số nguyên, thực hiện các thao tác sau:

- Xuất các phần tử trong mảng một chiều. (Hướng dẫn: sử dụng hàm **Arrays.toString**)
- Tìm phần tử lớn nhất và nhỏ nhất trong mảng. (Hướng dẫn: Dùng hàm **Math.max** và **Math.min**)
- Nhập vào số x, kiểm tra x có nằm trong mảng hay không? (Hướng dẫn: sử dụng hàm **Arrays.binarySearch**)
- Đếm số phần tử có giá trị bằng x trong mảng (x nhập từ bàn phím).
- Sắp xếp mảng một chiều tăng dần. (Hướng dẫn: sử dụng hàm **Arrays.sort**)

Bài 4: Nhập vào mảng hai chiều NxN số nguyên, thực hiện các thao tác sau:

- Xuất mảng 2 chiều ra màn hình.
- Tìm giá trị nhỏ nhất của mảng 2 chiều.
- Sắp xếp các phần tử trên từng dòng tăng dần.
- Sắp xếp các phần tử trên đường chéo chính tăng dần.

V. Bài tập về nhà

Phần 1: Nhóm bài tập lập trình về: kiểu dữ liệu, toán tử, lệnh if và switch

Bài 1: Tìm ước số chung lớn nhất của hai số nguyên nhập từ bàn phím.

Bài 2: Tìm bội số chung nhỏ nhất của 2 số nguyên nhập từ bàn phím.

Bài 3: Nhập tử và mẫu một phân số, đơn giản phân số và xuất phân số sau khi đơn giản.

Bài 4: Nhập vào số nguyên dương n, liệt kê tất cả các ước số của n.

Phần 2: Nhóm bài tập về mảng và vòng lặp.

Bài 1: Nhập vào mảng số nguyên A(kích thước n)

- Tạo mảng số nguyên B (kích thước m) với các giá trị ngẫu nhiên. (Hướng dẫn: sử dụng hàm **Math.random**)
- Xuất toàn bộ các phần tử của B ra màn hình.
- Tạo mảng C từ mảng A (copy A qua C) (Hướng dẫn: sử dụng hàm **copyOf**)
- Thay thế phần tử thứ 1->3 của mảng C bằng 3 phần tử cuối của mảng B (Hướng dẫn: sử dụng hàm **System.arraycopy**)
- Sắp xếp mảng C tăng dần và xuất ra màn hình.

Bài 2. Viết chương trình nhập vào mảng hai chiều A có n dòng, m cột, 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:

- Tìm phần tử lớn nhất của mảng A cùng chỉ số của số đó.
- Tìm và in ra các phần tử là số nguyên tố của mảng A (các phần tử không nguyên tố thì thay bằng số 0).
- Sắp xếp tất cả các cột của mảng A theo thứ tự tăng dần và in kết quả ra màn hình