

LAB 1: LÀM QUEN VỚI LẬP TRÌNH JAVA



MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Sử dụng được NetBean/Eclipse IDE để tạo 1 dự án
- ✓ Tạo một lớp chứa phương thức main
- ✓ Đọc dữ liệu từ bàn phím
- ✓ Xuất dữ liệu ra màn hình
- ✓ Sử dụng các hàm toán học từ Math.*

BÀI 1 (2 ĐIỂM)

Viết chương trình cho phép nhập họ và tên sinh viên, điểm trung bình từ bàn phím sau đó xuất ra màn hình với định dạng: <<họ và tên>> <<điểm>> điểm.

HƯỚNG DẪN:

- ✓ Chạy NetBean/Eclipse
- ✓ Tạo một dự án
- ✓ Tạo lớp chứa phương thức `public static void main(String[] args){...}`

```
public class Lab1Bai1{
    public static void main(String[] args){
    }
}
```
- ✓ Viết mã cho `main()`: Sử dụng `Scanner` để đọc dữ liệu từ bàn phím

```
Scanner scanner = new Scanner(System.in);
System.out.print("Họ và tên: ");
String hoTen = scanner.nextLine();
System.out.print("Điểm TB: ");
double diemTB = scanner.nextDouble();
```
- ✓ Sử dụng `System.out.print()`, `System.out.printf()` và `System.out.println()` để xuất ra màn hình

```
System.out.printf("%s %f điểm", hoTen, diemTB);
```

- ✓ Chạy ứng dụng

BÀI 2 (2 ĐIỂM)

Viết chương trình nhập từ bàn phím 2 cạnh của hình chữ nhật. Tính và xuất chu vi, diện tích và cạnh nhỏ của hình chữ nhật.

HƯỚNG DẪN:

- ✓ Chu vi = $(dai + rong) * 2$
- ✓ Diện tích = $dai * rong$
- ✓ Cạnh nhỏ nhất = $Math.min(dai, rong)$

BÀI 3 (2 ĐIỂM)

Viết chương trình nhập từ bàn phím cạnh của một khối lập phương. Tính và xuất thể tích của khối chữ nhật

HƯỚNG DẪN:

- ✓ Thể tích lập phương = $canh * canh * canh$
- ✓ Hoặc $Math.pow(canh, 3)$

BÀI 4 (2 ĐIỂM)

Viết chương trình nhập các hệ số của phương trình bậc 2. Tính delta và xuất căn delta ra màn hình.

HƯỚNG DẪN:

- ✓ $\Delta = Math.pow(b, 2) - 4 * a * c$
- ✓ Sử dụng $Math.sqrt(delta)$ để tính căn delta

BÀI 5 (2 ĐIỂM):

Giảng viên cho thêm

LAB 2: KIỂU, TOÁN TỬ, LỆNH IF VÀ SWITCH

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Sử dụng thành thạo các loại toán tử
- ✓ Biết cách xây dựng các loại biểu thức
- ✓ Sử dụng thành thạo lệnh rẽ nhánh (if)
- ✓ Sử dụng thành thạo lệnh lựa chọn (switch)

BÀI 1 (2 ĐIỂM)

Viết chương trình cho phép giải phương trình bậc nhất trong đó các hệ số a và b nhập từ bàn phím

HƯỚNG DẪN

- ✓ Sử dụng Scanner để đọc dữ liệu từ bàn phím
- ✓ Sử dụng lệnh if để xét giá trị của a và b

```
if(a == 0){  
    if(b == 0){  
        thông báo vô số nghiệm  
    }  
    else{  
        thông báo vô nghiệm  
    }  
}  
else{  
    tính và xuất nghiệm x = -b/a;  
}
```

BÀI 2 (2 ĐIỂM)

Viết chương trình cho phép giải phương trình bậc hai trong đó các hệ số a, b và c nhập từ bàn phím

HƯỚNG DẪN

- ✓ Nhập a, b và c
- ✓ Xét a
 - Nếu $a == 0 \Rightarrow$ giải phương trình bậc nhất
 - Ngược lại ($a \neq 0$)
 - Tính delta
 - Biện luận theo delta
 - $\Delta < 0$: vô nghiệm
 - $\Delta = 0$: nghiệm kép $x = -b/(2*a)$
 - $\Delta > 0$: 2 nghiệm phân biệt
 - $X1 = (-b + \sqrt{\Delta})/(2*a)$
 - $X2 = (-b - \sqrt{\Delta})/(2*a)$

BÀI 3 (2 ĐIỂM)

Viết chương trình nhập vào số điện sử dụng của tháng và tính tiền điện theo phương pháp lũy tiến

- ✓ Nếu số điện sử dụng từ 0 đến 50 thì giá mỗi số điện là 1000
- ✓ Nếu số điện sử dụng trên 50 thì giá mỗi số điện vượt hạn mức là 1200

HƯỚNG DẪN

- ✓ Nếu số điện sử dụng < 50 : $tien = soDien * 1000$
- ✓ Ngược lại: $tien = 50 * 1000 + (soDien - 50) * 1200$

BÀI 4 (2 ĐIỂM)

Viết chương trình tổ chức 1 menu gồm 3 chức năng để gọi 3 bài trên và một chức năng để thoát khỏi ứng dụng.

HƯỚNG DẪN

- ✓ Tạo phương thức menu() xuất ra màn hình thực đơn như sau

+-----+

1. Giải phương trình bậc nhất

2. Giải phương trình bậc 2

3. Tính tiền điện

4. Kết thúc

+-----+

Chọn chức năng: _

- ✓ Viết mã vào phương thức menu để nhận vào 1 số từ 1 đến 4.
- ✓ Sử dụng lệnh switch để gọi đến các hàm chức năng tương ứng.
 - giaiPTB1()
Chứa mã của bài 1
 - giaiPTB2()
Chứa mã của bài 2
 - tinhTienDien()
Chứa mã của bài 3

BÀI 5 (2 ĐIỂM)

Giảng viên cho thêm

LAB 3: MẢNG VÀ LỆNH LẶP

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Sử dụng thành thạo các lệnh lặp và ngắt vòng lặp
- ✓ Sử dụng mảng để nắm giữ nhiều phần tử cùng kiểu dữ liệu

BÀI 1 (2 ĐIỂM)

Viết chương trình nhập một số nguyên từ bàn phím và cho biết 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

- ✓ Cho một vòng lặp chạy từ 2 cho đến số nhập vào -1. Nếu có một số nhập vào chia hết cho biến chạy thì số đó không phải là số nguyên tố.

```
boolean ok = true;
for(int i=2; i < N-1; i++){ // N là số nhập từ bàn phím
    if(N % i == 0){
        ok = false;
        break;
    }
    i++;
}
```

- ✓ Kiểm tra biến ok bạn sẽ biết N có phải là số nguyên tố hay không

BÀI 2 (2 ĐIỂM)

Viết chương trình xuất ra màn hình bảng cửu chương

HƯỚNG DẪN

- ✓ Để xuất 1 bảng nhân x bạn cần xây dựng đoạn mã

```
int x = 8;
for(int i=1; i<=10;i++){
```

```
System.out.printf("%d x %d = %d", x, i, x*i)
System.out.println()
}
```

- ✓ Vậy để xuất 9 bảng nhân, bạn sử dụng 2 vòng lặp for lồng nhau
 - Vòng for ngoài cho biến chạy i từ 1 đến 9
 - Vòng for trong cho biến chạy j từ 1 đến 10
- ✓ Cứ mỗi lần lặp của vòng lặp bên trong bạn xuất
 - System.out.printf("%d x %d = %d", i, j, i*j)
 - System.out.println()

BÀI 3: (2 ĐIỂM)

Viết chương trình nhập mảng số nguyên từ bàn phím.

- ✓ Sắp xếp và xuất mảng vừa nhập ra màn hình.
- ✓ Xuất phần tử có giá trị nhỏ nhất ra màn hình
- ✓ Tính và xuất ra màn hình trung bình cộng các phần tử chia hết cho 3

HƯỚNG DẪN

- ✓ Sử dụng Arrays.sort(mang) để sắp xếp sau đó sử dụng vòng lặp duyệt từng phần tử và xuất ra màn hình
- ✓ Cho số đầu tiên của mảng là số nhỏ nhất sau đó so sánh số nhỏ nhất với các số còn lại nếu số được so sánh nhỏ hơn thì lấy số đó làm số nhỏ nhất
Min = Math.min(min, a[i])
- ✓ Duyệt mảng và 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 3Lấy tổng chia cho số đếm bạn sẽ có kết quả

BÀI 4 (2 ĐIỂM)

Viết chương trình nhập 2 mảng họ tên và điểm của sinh viên.

- ✓ Xuất 2 mảng đã nhập, mỗi sinh viên có thêm học lực

- Yếu: điểm < 5
 - Trung bình: $5 \leq \text{điểm} < 6.5$
 - Khá: $6.5 \leq \text{điểm} < 7.5$
 - Giỏi: $7.5 \leq \text{điểm} < 9$
 - Xuất sắc: điểm ≥ 9
- ✓ Sắp xếp danh sách sinh viên đã nhập tăng dần theo điểm

HƯỚNG DẪN

- ✓ Sử dụng lệnh if để xét học lực sau đó xuất thông tin từng sinh viên
 - Họ tên:
 - Điểm:
 - Học lực:
- ✓ Bài này bạn không thể sử dụng `Arrays.sort()` để sắp xếp được mà phải sử dụng đến thuật toán tùy biến (tham khảo slide bài giảng)

BÀI 5 (2 ĐIỂM)

Giảng viên cho thêm

LAB 4: LỚP VÀ ĐỐI TƯỢNG

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Xây dựng một lớp hoàn chỉnh
- ✓ Biết cách sử dụng các đặc tả truy xuất cho các thành viên bên trong một lớp

BÀI 1 (2 ĐIỂM)

Tạo lớp SanPham gồm 3 thuộc tính là tên, giá và giảm giá. Lớp cũng gồm 2 phương thức là tính thuế nhập khẩu (10% giá sản phẩm) và xuất thông tin ra màn hình. Thông tin xuất ra màn hình gồm:

- ✓ Tên sản phẩm:
- ✓ Đơn giá:
- ✓ Giảm giá:
- ✓ Thuế nhập khẩu:

HƯỚNG DẪN:

- ✓ Khai báo lớp theo mô hình sau

SanPham
+tenSp: String +donGia: double +giamGia: double
+getThueNhapKhau(): double +xuat(): void +nhap():void

BÀI 2 (2 ĐIỂM)

Viết chương trình tạo ra 2 sản phẩm thông tin được nhập từ bàn phím sau đó gọi phương thức xuất để xuất thông tin 2 đối tượng sản phẩm đã tạo

HƯỚNG DẪN:

- ✓ Tạo lớp chứa phương thức main()
- ✓ Trong phương thức main() tạo 2 đối tượng sp1 và sp2 từ lớp sản phẩm
- ✓ Gọi phương thức nhap() của 2 đối tượng sp1 và sp2 để nhập dữ liệu từ bàn phím
- ✓ Gọi phương thức xuất() của 2 đối tượng sp1 và sp2 để xuất thông tin của mỗi đối tượng ra màn hình

BÀI 3 (2 ĐIỂM)

Nâng cấp lớp SanPham bằng cách bổ sung public cho các đặt tả truy xuất cho phương thức xuất() và private cho getThueThuNhap(). Đồng thời bổ sung 2 hàm tạo, hàm tạo thứ nhất gồm 3 tham số là tên, giá và giảm giá, hàm tạo thứ 2 gồm 2 tham số là tên và giá (ngầm hiểu không giảm giá).

Viết chương trình tạo 2 sản phẩm có giảm giá và không giảm giá sau đó xuất thông tin 2 sản phẩm ra màn hình

HƯỚNG DẪN:

- ✓ Bổ sung 2 hàm tạo

```
public SanPham(String tenSp, double donGia, double giamGia){
    this.tenSp = tenSp;
    ...
}
public SanPham(String tenSp, double donGia){
    this(tenSp, donGia, 0)
}
```
- ✓ Sử dụng hàm tạo để tạo sản phẩm

```
SanPham sp1 = new SanPham(tenSp, donGia)
```

BÀI 4 (2 ĐIỂM)

Nâng cấp lớp SanPham bằng cách khai báo các trường dữ liệu với đặc tả truy xuất là private để hạn chế truy xuất trực tiếp đến các trường này sau đó bổ sung các phương thức getter và setter để đọc ghi dữ liệu các trường.

HƯỚNG DẪN:

- ✓ Cứ mỗi trường dữ liệu được khai private bạn cần định nghĩa một cặp phương thức getter/setter để cho phép đọc ghi dữ liệu thông qua các phương thức này. Với cách làm này bạn dễ dàng nâng cấp lớp để bảo vệ hoặc thực hiện các tính toán cần thiết trước khi thực hiện các thao tác đọc ghi dữ liệu

```
public String getTenSp(){  
    return this.tenSp;  
}  
public void setTenSp(String tenSp){  
    this.tenSp = tenSp;  
}
```

BÀI 5 (2 ĐIỂM)

Giảng viên cho thêm

LAB 5: ARRAYLIST

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Sử dụng ArrayList để duy trì tập hợp các phần tử
- ✓ Sử dụng thư viện tiện ích xử lý tập hợp Collections

BÀI 1 (2 ĐIỂM)

Nhập danh sách số thực với số lượng tùy ý từ bàn phím sau đó xuất danh sách vừa nhập và tổng của nó.

HƯỚNG DẪN:

- ✓ Sử dụng ArrayList<Double> để duy trì danh sách số thực được nhập từ bàn phím
- ✓ Sử dụng vòng lặp while để nhập số lượng tùy ý

```
while(true){  
    Double x = scanner.nextDouble();  
    list.add(x);  
  
    System.out.print("Nhập thêm (Y/N)? ");  
    if(scanner.nextLine().equals("N")){  
        break;  
    }  
}
```
- ✓ Sử dụng vòng lặp for-each để duyệt và xuất các phần tử của list ra màn hình

BÀI 2 (3 ĐIỂM)

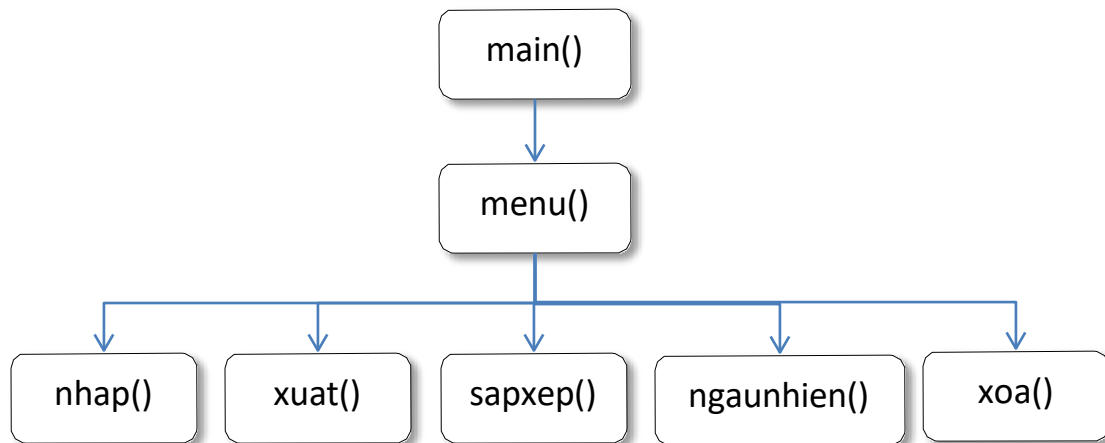
Viết chương trình thực hiện các chức năng sau

1. Nhập danh sách họ và tên
2. Xuất danh sách vừa nhập
3. Xuất danh sách ngẫu nhiên

4. Sắp xếp giảm dần và xuất danh sách
5. Tìm và xóa họ tên nhập từ bàn phím
6. Kết thúc

HƯỚNG DẪN

- ✓ Tổ chức ứng dụng theo sơ đồ sau



- ✓ Nhập và xuất tương tự bài 1
- ✓ Sử dụng hàm Collections.shuffle(list) để hoán đổi ngẫu nhiên các phần tử trong list
- ✓ Sử dụng Collections.sort(list) để sắp xếp tăng dần sau đó sử dụng Collections.reverse(list) để đảo các phần tử trong ArrayList
- ✓ Duyệt list và sử dụng list.remove() để xóa phần tử, dùng break để ngắt vòng lặp sau khi xóa

BÀI 3 (3 ĐIỂM)

Xây dựng ứng dụng quản lý sản phẩm (thông tin mỗi sản phẩm gồm tên và giá) theo menu sau

1. Nhập danh sách sản phẩm từ bàn phím
2. Sắp xếp giảm dần theo giá và xuất ra màn hình
3. Tìm và xóa sản phẩm theo tên nhập từ bàn phím
4. Xuất giá trung bình của các sản phẩm

HƯỚNG DẪN

- ✓ Tổ chức ứng dụng tương tự bài 2
- ✓ Sử dụng Collections.sort(list, comparator) để sắp xếp danh sách sản phẩm với tiêu chí sắp xếp được định nghĩa như sau

```
Comparator<SanPham> comp = new Comparator<SanPham>() {  
    @Override  
    public int compare(SanPham o1, SanPham o2) {  
        return o1.donGia.compareTo(o2.donGia);  
    }  
};
```

BÀI 4 (2 ĐIỂM)

Giảng viên cho thêm

LAB 6: CHUỖI VÀ BIỂU THỨC CHÍNH QUI

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Hiểu sâu sắc về chuỗi
- ✓ Sử dụng thành thạo các phương thức thao tác chuỗi
- ✓ Biết cách xây dựng biểu thức chính qui
- ✓ Sử dụng phương thức `matches()` để kiểm tra chuỗi có so khớp (đối sánh) với biểu thức chính qui hay không.

BÀI 1 (2 ĐIỂM)

Nhập họ và tên từ bàn phím. Xuất tên, họ và tên đệm ra màn hình trong đó tên và họ xuất IN HOA.

HƯỚNG DẪN:

- ✓ Sử dụng các phương thức xử lý chuỗi: `substring()`, `indexOf()`, `lastIndexOf()` và `toUpperCase()` để thực hiện bài này
 - Họ được lấy từ đầu đến ký tự trắng đầu tiên
 - Tên được lấy từ sau ký tự trắng cuối cùng đến cuối chuỗi
 - Tên đệm được lấy sau ký tự trắng đầu tiên và trước ký tự trắng cuối cùng

BÀI 2 (3 ĐIỂM)

Nhập vào mảng 5 sản phẩm (mỗi sản phẩm gồm tên, giá và hãng). Xuất thông tin các sản phẩm có hãng là Nokia.

HƯỚNG DẪN:

- ✓ Xây dựng lớp `SanPham` có mô hình như bài

SanPham
+tenSp: String
+donGia: double

+hang: String
+xuat(): void
+nhap():void

- ✓ Tạo lớp chứa phương thức main() và sử dụng vòng lặp để nhập 5 sản phẩm vào ArrayList<SanPham>
- ✓ Duyệt ArrayList<SanPham> và so sánh trường hãng với “Nokia” sau đó gọi phương thức để xuất thông tin sản phẩm

BÀI 3 (3 ĐIỂM)

Nhập mảng 5 sinh viên (mỗi sinh viên gồm họ tên, email, số điện thoại, chứng minh nhân dân). Kiểm tra và thông báo lỗi nếu nhập không đúng định dạng email, số điện thoại và CMND.

HƯỚNG DẪN:

- ✓ Thực hiện tương tự như bài 3 để nhập danh sách sinh viên
- ✓ Kiểm soát dữ liệu nhập vào bằng cách sử dụng biểu thức chính qui (tham khảo slide bài giảng) để kiểm tra và thông báo lỗi
 - Email
 - Số điện thoại
 - CMND

BÀI 4 (2 ĐIỂM)

Giảng viên cho thêm

LAB 7: KẾ THỪA

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Xây dựng ứng dụng với nhiều lớp được tổ chức theo sự phân cấp thừa kế trong Java
- ✓ Sử dụng lại những gì đã có ở một lớp khác
- ✓ Ghi đè để hiệu chỉnh lại nội dung của phương thức ở lớp con
- ✓ Ứng dụng lớp trừu tượng và phương thức trừu tượng

BÀI 1 (2 ĐIỂM)

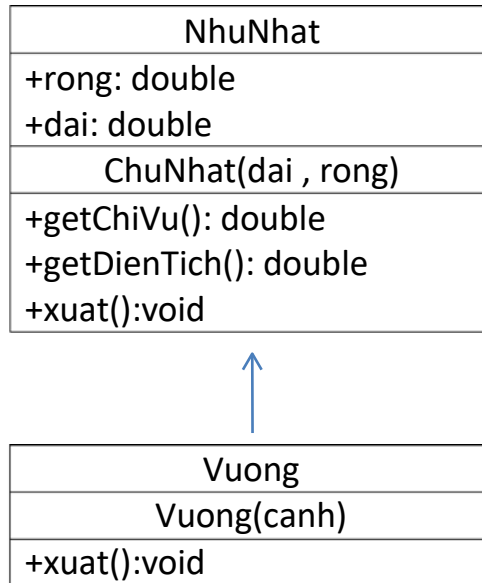
Xây dựng lớp ChuNhat gồm 2 thuộc tính là rong và dai và các phương thức getChuVi() và getDienTich() để tính chu vi và diện tích. Phương thức xuất() sẽ xuất ra màn hình chiều rộng, chiều dài, diện tích và chu vi.

Xây dựng lớp Vuong kế thừa từ lớp ChuNhat và ghi đè phương thức xuất() để xuất thông tin cạnh, diện tích và chu vi.

Viết chương trình nhập 2 hình chữ nhật và một hình vuông sau đó xuất ra màn hình.

HƯỚNG DẪN

- ✓ Xây dựng các lớp theo kiến trúc phân cấp kế thừa như sau



- ✓ Hàm tạo của lớp **Vuong** gọi lại hàm tạo của lớp **ChuNhat** và truyền cạnh của hình vuông cho 2 tham số chiều dài và chiều rộng
- ✓ Ghi đè phương thức xuất để xuất thông tin của hình vuông
- ✓ Tạo lớp chứa phương thức `main()` và nhập chiều dài, chiều rộng của hình chữ nhật và cạnh của hình vuông. Sau đó sử dụng các lớp **ChuNhat** và **Vuong** để tạo các đối tượng và gọi phương thức `xuat()` để xem thông tin:
`ChuNhat cn = new ChuNhat(dai, rong)`
`ChuNhat vu = new Vuong(canh)`
`cn.xuat();`
`vu.xuat();`

BÀI 2 (2 ĐIỂM)

Tạo lớp **SinhVienPoly** gồm 2 thuộc tính họ tên và ngành cùng với phương thức trừu tượng là `getDiem()`. Thêm phương thức `getHocLuc()` để xếp loại học lực. Lớp cũng bao gồm một phương thức `xuat()` để xuất họ tên, ngành, điểm và học lực ra màn hình.

HƯỚNG DẪN

- ✓ Xây dựng lớp có mô hình như sau

SinhVienPoly
+hoTen: String
+nganh: String
SinhVienPoly (hoTen , nganh)
+getDiem(): double
+getHocLuc(): String
+xuat():void

- ✓ Vì chưa biết sinh viên này học những môn nào nên chưa tính được điểm vì vậy phương thức getDiem() phải là phương thức trừu tượng
- ✓ Chú ý lớp SinhVienPoly phải là lớp trừu tượng vì có phương thức getDiem() là phương thức trừu tượng
- ✓ Phương thức getHocLuc() được viết bình thường vẫn sử dụng phương thức getDiem() để lấy điểm của sinh viên mặc dù hiện tại vẫn chưa biết điểm được tính thế nào. Học lực được tính như sau
 - Yếu: điểm < 5
 - Trung bình: 5 <= điểm < 6.5
 - Khá: 6.5 <= điểm < 7.5
 - Giỏi: 7.5 <= điểm < 9
 - Xuất sắc: điểm >= 9

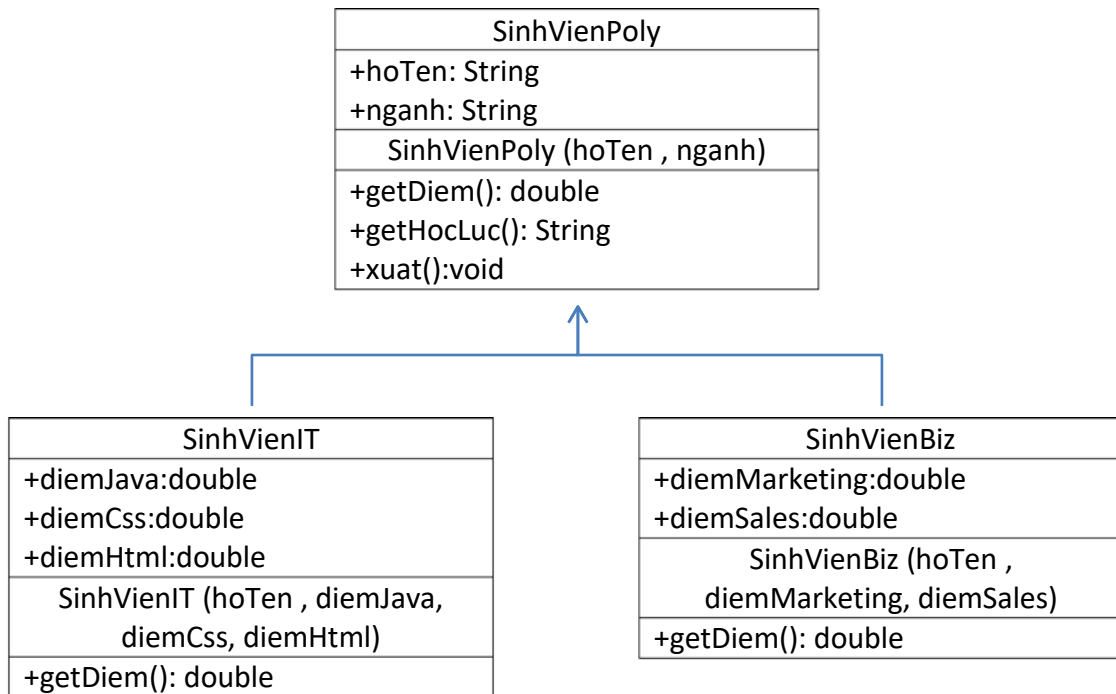
BÀI 3 (2 ĐIỂM)

Tạo lớp SinhVienIT và SinhVienBiz kế thừa từ lớp SinhVienPoly.

- ✓ SinhVienIT gồm các thuộc tính điểm java, html, css. Ghi đè phương thức getDiem() để tính điểm cho sinh viên IT theo công thức $(2 * \text{java} + \text{html} + \text{css}) / 4$
- ✓ SinhVienBiz gồm các thuộc tính điểm marketing, sales. Ghi đè phương thức getDiem() để tính điểm cho sinh viên Biz theo công thức $(2 * \text{marketing} + \text{sales}) / 3$

HƯỚNG DẪN

- ✓ Tổ chức các lớp theo kiến trúc phân cấp kế thừa như sau



- ✓ Ghi đè phương thức `getDiem()` trên 2 lớp **SinhVienIT** và **SinhVienBiz** theo yêu cầu của đề để tính điểm cho các sinh viên của các ngành.

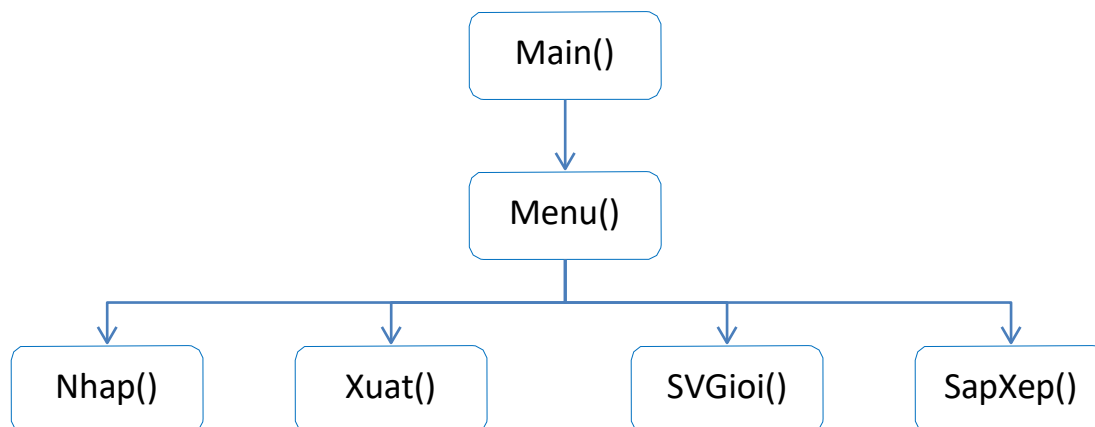
BÀI 4 (2 ĐIỂM)

Viết chương trình quản lý sinh viên:

1. Nhập danh sách sinh viên
2. Xuất thông tin danh sách sinh viên
3. Xuất danh sách sinh viên có học lực giỏi
4. Sắp xếp danh sách sinh viên theo điểm
5. Kết thúc

HƯỚNG DẪN

- ✓ Tổ chức chương trình

**BÀI 5 (2 ĐIỂM)**

Giảng viên cho thêm

LAB 8: KIẾN THỨC NÂNG CAO VỀ PHƯƠNG THỨC VÀ LỚP

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Sử dụng phương thức tĩnh để xây dựng một thư viện các hàm tiện ích phục vụ cho các thao tác xử lý nhanh sau này.

PHÁT BIỂU CHUNG

Tạo lớp final có tên là XPoly và bổ sung các phương thức tĩnh và viết mã theo yêu cầu của các bài sau đây.

BÀI 1 (2 ĐIỂM)

Thêm một phương thức tĩnh có tên là sum() với tham số biến đổi kiểu double. Viết mã cho phương thức này để thực hiện tính tổng các tham số truyền vào.

HƯỚNG DẪN

- ✓ Thêm phương thức tĩnh với tham số biến đổi
 - `public static final double sum(double...x){...}`

BÀI 2 (2 ĐIỂM)

Bổ sung thêm 2 phương thức tĩnh để tìm số lớn nhất và nhỏ nhất từ tham số biến đổi

- ✓ `XPoly.min(double...x)`: tìm số nhỏ nhất
- ✓ `XPoly.max(double ...x)`: tìm số lớn nhất

BÀI 3 (2 ĐIỂM)

Bổ sung thêm 1 phương thức tĩnh `toUpperFirstChar(String s)` để chuyển đổi các ký tự đầu tiên của mỗi từ sang ký tự hoa. Ví dụ: “nguyen van teo” sẽ đổi thành “Nguyen Van Teo”.

HƯỚNG DẪN

-
- ✓ Tách chuỗi tham số theo ký tự trắng với phương thức `split(" ")` `String[]`
`words = s.split(" ");`
 - ✓ Duyệt các từ trong mảng `words` và lấy ký tự đầu tiên của mỗi từ với phương thức `charAt()` và đổi sang ký tự hoa
`char firstChar = words[i].charAt(0);`
`String upperFirstChar = String.valueOf(firstChar).toUpperCase().charAt(0);`
 - ✓ Thay đổi từ cũ bằng từ đã chuyển đổi sang ký tự hoa
`words[i] = upperFirstChar + words[i].substring(1);`
 - ✓ Thực hiện nối mảng các từ thành chuỗi với phương thức `String.join(words, " ")`
`String ss = String.join(words, " ")`

BÀI 4 (2 ĐIỂM)

Viết chương trình sử dụng các phương thức tiện ích đã xây dựng trong lớp `XPoly` ở trên.

BÀI 5 (2 ĐIỂM)

Giảng viên cho thêm

