

Mục lục

Chương 1: Lập trình hướng đối tượng và các khái niệm cơ bản.....	3
Bài tập 1: Ví dụ đầu tiên.....	3
Chương 2: Các thành phần cơ sở của java	5
Bài tập 2: Khai báo biến, sử dụng các lớp và hàm có sẵn; chạy và thực thi chương trình	5
Bài tập 3: Khai báo biến, sử dụng các mẫu định dạng số thực ; in dữ liệu	5
Bài tập 4: Khai báo dữ liệu; nhập dữ liệu; Sử dụng lớp Math; in dữ liệu theo định dạng.	5
Bài tập 5: Cấu trúc rẽ nhánh: Bài toán phương trình bậc 1.....	6
Bài tập 6: Cấu trúc rẽ nhánh: Bài toán phương trình bậc 2.....	6
Bài tập 7: Cấu trúc rẽ nhánh: Bài toán tam giác.....	6
Bài tập 8: Cấu trúc rẽ nhánh: Bài toán tính khoảng cách 2 điểm.	6
Bài tập 9: Cấu trúc chọn: Bài toán tìm số ngày trong 1 tháng của năm.	6
Bài tập 10: Cấu trúc chọn: Bài toán bắt phím và tính biểu thức tương ứng.....	7
Bài tập 11: Cấu trúc lặp : Bài toán tách số.....	7
Bài tập 12: Cấu trúc lặp : Bài toán số hoàn hảo.....	7
Bài tập 13: Cấu trúc lặp : Bài toán xử lý chuỗi tìm từ trong câu.	7
Bài tập 14: Cấu trúc lặp +thao tác với xâu ký tự: Bài toán kiểm tra ký tự.....	7
Bài tập 15: Cấu trúc điều khiển +Thao tác với xâu ký tự: Bài toán so sánh hai chuỗi	7
Bài tập 16: Cấu trúc điều khiển +Thao tác với xâu ký tự: Bài toán nối chuỗi.....	7
Bài tập 17: Cấu trúc điều khiển +Thao tác với xâu ký tự: Bài toán tách chuỗi	7
Bài tập 18: Xây dựng hàm bên ngoài và gọi hàm trong hàm main(): bài toán PT bậc 2.	8
Bài tập 19: Xây dựng hàm bên ngoài và gọi hàm trong hàm main(): bài toán số fibonacci.....	8
Chương 3: Chương 3: Lớp và các thành phần.....	8
Bài tập 20: Xây dựng lớp: Bài toán hình chữ nhật.....	8
Bài tập 21: Xây dựng lớp : Bài toán hình tròn	9
Bài tập 22: Xây dựng lớp :Bài toán phương trình bậc 2.	9
Bài tập 23: Xây dựng lớp : Bài toán hồ sơ.	9
Bài tập 24: Xây dựng lớp: Bài toán sinh Viên	10
Bài tập 25: Xây dựng lớp: Bài toán quản lý đăng ký xe.	11
Bài tập 26: Xây dựng lớp: Bài toán Nhân viên và hàm thành phần.....	11
Bài tập 27: Xây dựng lớp: Bài toán TamGiac:.....	12
Bài tập 28: Xây dựng lớp: Bài toán nhân viên sản xuất.....	13
Bài tập 29: Xây dựng lớp: Bài toán Sinh viên	13
Bài tập 30: Xây dựng lớp: Cài đặt quan hệ kết tập	14
Bài tập 31: Quan hệ kết tập: Bài toán Sinh viên – môn học	16
Bài tập 32: Quan hệ kết tập: Bài toán hóa đơn bán hàng.....	17
Bài tập 33: Quan hệ kết tập: Bài toán sinh viên – điểm.....	20

Bài tập 34: Quan hệ kết tập: Bài toán Quản lý sách.....	20
Bài tập 35: Quan hệ kế thừa: Đa hình thông qua kế thừa- bài toán hình phẳng	20
Bài tập 36: Quan hệ kế thừa: Bài toán quản lý giao dịch nhà đất.	21
Bài tập 37: Quan hệ kế thừa: Bài toán quản lý tiền điện.....	22
Bài tập 38: Quan hệ kế thừa: Bài toán quản lý khách sạn.....	22
Bài tập 39: Quan hệ kế thừa: Bài toán Quản lý Kho.....	22
Bài tập 40: Quan hệ kế thừa - kết tập: Bài toán bệnh viện.....	23
Bài tập 41: Quan hệ kế thừa- kết tập: Bài toán quản lý sinh viên	24
Bài tập 42: Quan hệ kế thừa - kết tập: Bài toán Quản lý máy tính	25
Bài tập 43: Quan hệ kế thừa- lớp trừu tượng : Bài toán hình phẳng	25
Bài tập 44: Quan hệ kế thừa- lớp trừu tượng : Bài toán quản lý nhân viên	25
Bài tập 45: Quan hệ kế thừa- lớp trừu tượng: Bài toán thuê bao internet	26
Bài tập 46: Quan hệ đa kế thừa- Interface: Bài toán Máy móc	27
Bài tập 47: Quan hệ đa kế thừa- Interface: Bài toán Sinh viên	29
Chương 4: Lỗi thực thi và quá trình gom rác	31
Bài tập 48: Kiểm soát lỗi thực thi: Bài toán chia cho 0.	31
Bài tập 49: Kiểm soát lỗi thực thi: bài toán ràng buộc dữ liệu.	31
Bài tập 50: Kiểm soát lỗi thực thi: : Bài toán lớp xử lý ngoại lệ 1.....	32
Bài tập 51: Kiểm soát lỗi thực thi: Bài toán viết lớp xử lý ngoại lệ 2.....	32
Bài tập 52: Kiểm soát lỗi thực thi: Tự viết hàm xử lý ngoại lệ.....	32
Bài tập 53: Xử lý ngoại lệ: Bài toán sinh viên kiểm tra hợp lệ dữ liệu.....	32
Chương 5: Các lớp cơ sở và cấu trúc dữ liệu.	34
Bài tập 54: Các thao tác trên tập hợp: Bài toán nhập xuất, sắp xếp tập hợp nguyên thủy	34
Bài tập 55: Các thao tác trên tập hợp: Bài toán tìm, thêm, xóa phần tử lớp Circle.....	34
Bài tập 56: Các thao tác trên tập hợp: Bài toán quản lý của Công ty TrueLove.....	35
Bài tập 57: Các thao tác trên tập hợp: Bài toán quản lý phòng học	36
Bài tập 58: Các thao tác trên tập hợp: Quản lý khách hàng xếp hàng mua vé tại nhà ga.	37
Bài tập 59: Các thao tác trên tập hợp: Quản lý đĩa CD	38
Bài tập 60: Các thao tác trên tập hợp: Quản lý mua vé tàu	39
Chương 6: Các luồng vào ra và thao tác với tệp dữ liệu	40
Bài tập 61: Thao tác với file text.....	40
Bài tập 62: Tuần tự hóa đọc ghi danh sách đối tượng vào file (Object Serializable)	40
Bài tập 63: Bài tập tổng hợp: Quản lý phòng thi	42

Chương 1: Lập trình hướng đối tượng và các khái niệm cơ bản

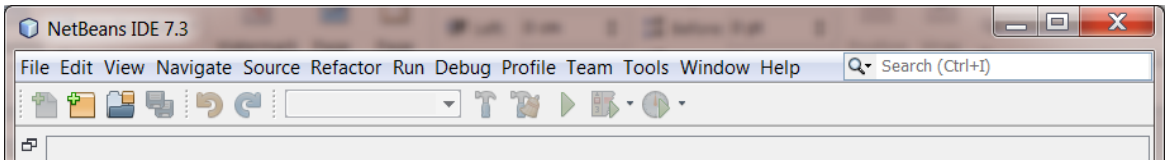
- + Sử dụng JDK để biên dịch và thực thi chương trình.
- + Sử dụng NetBeans để soạn thảo, biên dịch và thực thi chương trình.
- + Khai báo và sử dụng biến, đối tượng.
- + Sử dụng thành thạo các cấu trúc điều khiển.

Bài tập 1: Ví dụ đầu tiên.

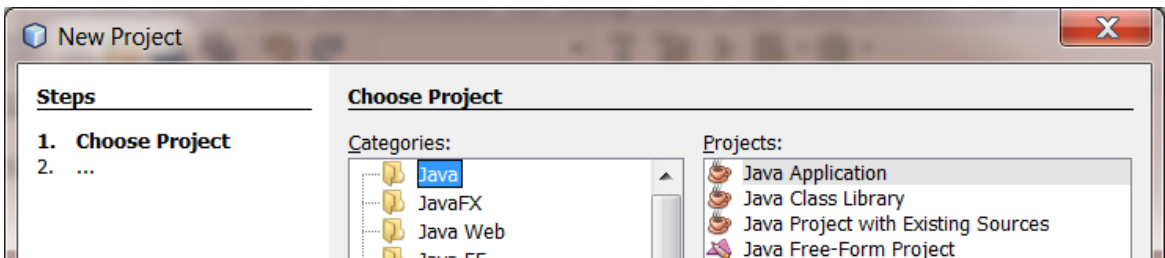
Mục đích: Sử dụng công cụ netBeans

Yêu cầu:

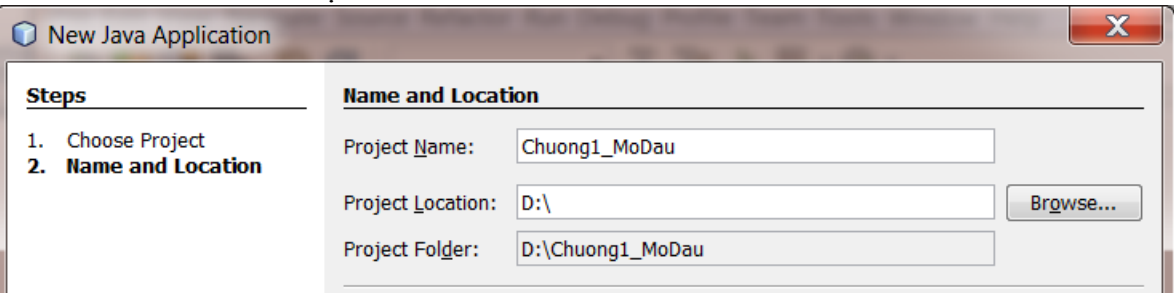
- Khởi động NetBeans: Sau khi download và giải nén giao diện như sau:



- Cách tạo Java Project: Vào menu File /New/ new Project; java->Java Application;



Chọn next. Màn hình hiển thị như sau



Trong đó:

- Project Name: đặt tên prj theo quy tắc đặt tên (độ dài bất kỳ, phân biệt chữ hoa, chữ thường; bao gồm 1-9, a-z, A-Z; _, \$)
- Project location:
 - o chọn thư mục lưu project (Nên lưu vào ổ đĩa E trên phòng máy)
- Project folder:
 - o Mặc định có tên giống với tên prj, có thể đổi tên lại nếu cần.
 - o Chọn Finish kết thúc quá trình tạo mới 1 prj.
- Tạo 1 lớp tên là HelloWorld.
 - o Thực hiện chọn chuột phải vào **Project / New/** hoặc menu **File\new**;
 - o chọn class (xem hình minh họa bên dưới).
 - o nên đặt tên lớp và tên file.java giống nhau.

Chương 2: Các thành phần cơ sở của java

Bài tập 2: Khai báo biến, sử dụng các lớp và hàm có sẵn; chạy và thực thi chương trình

Yêu cầu: Tự động sinh chiều dài, chiều rộng của hình chữ nhật. tính và in ra chu vi diện tích của hình. Yêu cầu kết quả in ra có định dạng.

Gợi ý:

1. Khai báo và import thư viện Random
2. Sử dụng hàm random để sinh số thực, mở rộng sinh các phần tử khác.
3. sử dụng lớp DecimalFormat để định dạng số thực
4. in thông tin ra màn hình sử dụng hàm xuất dữ liệu.

Thực hiện : //tạo file **HCN.java** và triển khai nội dung sau trong hàm main.

```
public class HCN {  
    public static void main(String[] args) {  
        float dai,rong, chuVi, dienTich;  
  
        // cách 1: dai rộng tự sinh các số thực sử dụng lớp có sẵn Random  
        Random random = new Random();  
        dai = random.nextFloat () + 1; //tìm hiểu và giải thích có cần +1 hay không?  
        rong = random.nextFloat () + 1;  
        chuVi=(dai+rong)*2;  
        dienTich=dai*rong;  
  
        //khai báo định dạng và kết xuất kết quả  
        DecimalFormat d=new DecimalFormat("###,###.0#");  
        System.out.println("dai="+dai+" rong="+rong+" chu vi ="  
            + d.format(chuVi)+ "dienTich="+ d.format(dienTich);  
    }  
}
```

Kết quả được xuất ra là gì?

Mở rộng. Sử dụng định dạng in printf in kết quả dạng sau. Mỗi cột có độ dài 15 chỗ:

Chiều dài	Chiều rộng	chu vi	Diện tích
7.8	9.3	34.2	72.5

Bài tập 3: Khai báo biến, sử dụng các mẫu định dạng số thực ; in dữ liệu

Yêu cầu: Viết hàm in số thực theo định dạng theo các phương pháp khác nhau.

//Tạo file **DecimalFormatDemo.java**

```
public class DecimalFormatDemo {  
    //tách tạo hàm dùng chung  
    public static void customFormat(String mau, double giaTri) {  
        DecimalFormat df = new DecimalFormat(mau);  
        String output = df.format(giaTri);  
        System.out.println(giaTri + " theo mẫu " + mau + " là " + output);  
    }  
    public static void main(String[] args) {  
        //gọi hàm đã xây dựng trong hàm main.  
        customFormat("###,###.###", 123456.789);  
        customFormat("###.##", 123456.789);  
        customFormat("000000.000", 123.78);  
        customFormat("$###,###.###", 12345.67);  
        customFormat("###,##0.0#", 000.007);  
    }  
}
```

Bài tập 4: Khai báo dữ liệu; nhập dữ liệu; Sử dụng lớp Math; in dữ liệu theo định dạng.

Yêu cầu: Viết chương trình tính thể tích và diện tích bề mặt của hình cầu với bán kính r nhập vào

($r \geq 0$). Kết quả chỉ cần lấy 4 chữ số thập phân. Công thức tính: Thể tích = $\frac{4}{3}\pi r^3$; Diện tích bề mặt = $4\pi r^2$

Hướng dẫn:

- Tìm các danh từ tương tức các thông tin mô tả hình cầu khai báo các biến, đồng thời dựa vào công thức tính đã cho chọn kiểu dữ liệu phù hợp.
- Dựa vào các động từ yêu cầu thực hiện xây dựng các lệnh phù hợp.
- Lấy số **Pi** trong theo cú pháp: **Math.PI**
- Chọn định dạng số thực theo lớp **DecimalFormat** của hệ thống như đã thực hiện ở các bài trên.

```
DecimalFormat df = new DecimalFormat("0.#");
df.format(x); in x theo định dạng đã định nghĩa như ở trên.
```

- Xây dựng lớp chứa hàm main nhập và in hình cầu.

Bài tập 5: Cấu trúc rẽ nhánh: Bài toán phương trình bậc 1.

Yêu cầu: Viết chương trình giải phương trình một ẩn: $ax + b = 0$,

Hướng dẫn:

- Tạo lớp: PTB1 có chứa hàm main:
- Khai báo các hệ số có kiểu dữ liệu thực hoặc nguyên
- Thực hiện các lệnh nhập : nhập các hệ số a,b
- Thực hiện giải phương trình và in in kết quả tương ứng.

Bài tập 6: Cấu trúc rẽ nhánh: Bài toán phương trình bậc 2.

Yêu cầu: Viết chương trình giải phương trình bậc hai $ax^2 + bx + c = 0$,

Hướng dẫn:

- Tạo lớp: PTB2 có chứa hàm main:
- Khai báo các hệ số có kiểu dữ liệu thực hoặc nguyên
- Thực hiện các lệnh nhập : nhập các hệ số, trong đó yêu cầu nhập a khác không
- Thực hiện giải phương trình và in in kết quả tương ứng

Bài tập 7: Cấu trúc rẽ nhánh: Bài toán tam giác.

Yêu cầu: Viết chương trình cho nhập vào kích thước 3 cạnh a, b, c của một tam giác. Tính và in ra diện tích, chu vi của tam giác. Trong đó 3 số a, b, c có lập thành một tam giác không thì cần các cạnh phải là số dương và tổng 2 số bất kỳ luôn lớn hơn số còn lại.

Hướng dẫn: Diện tích = $\sqrt{s(s-a)(s-b)(s-c)}$, biết s bằng 1/2 chu vi tam giác.

Bài tập 8: Cấu trúc rẽ nhánh: Bài toán tính khoảng cách 2 điểm.

Yêu cầu: Viết chương trình tính khoảng cách giữa 2 điểm có tọa độ (x_1, y_1) và (x_2, y_2) , biết công thức để

tính khoảng cách là: distance = $\sqrt{[x_1 - x_2]^2 + [y_1 - y_2]^2}$

Bài tập 9: Cấu trúc chọn: Bài toán tìm số ngày trong 1 tháng của năm.

Yêu cầu: Viết chương trình nhập vào tháng, năm dương lịch bất kỳ. in ra số ngày của tháng đó. Chú ý tháng 2 là tháng nhuận có 29 ngày nếu năm đó chia hết cho 4 và không phải là năm thế kỷ.

Bài tập 10: Cấu trúc chọn: Bài toán bất phím và tính biểu thức tương ứng.

Yêu cầu: Viết chương trình nhập vào hai số a, b và một phím bất kỳ.

- Nếu nhập “C” hoặc ” C” thì in ra tổng hai số
- Nếu nhập “T” hoặc ”t” thì in ra hiệu hai số
- Nếu nhập “N” hoặc ” N” thì in ra tích hai số
- Nếu nhập “D” hoặc ” d” thì in ra thương hai số
- Nếu nhập phím khác các ký tự trên thì in ra chính a và b.

Gợi ý : Để lấy 1 ký tự có 2 cách

- Cách 1 : nhập vào 1 xâu. Bỏ ký tự trắng 2 đầu ; lấy ký tự đầu tiên trong xâu (Xâu.trim() ; xâu.charAt(vị trí).
- Cách 2 : Dùng lệnh (char) System.in.read() để đọc 1 ký tự bất kỳ. Yêu cầu bắt lỗi lan truyền.

Bài tập 11: Cấu trúc lặp : Bài toán tách số.

Yêu cầu: Viết chương trình nhập vào một số tự nhiên lớn bất kỳ. in số tự nhiên đó ra theo chiều ngược lại, tổng các chữ số tạo thành số tự nhiên đó.

Bài tập 12: Cấu trúc lặp : Bài toán số hoàn hảo

Yêu cầu: Số hoàn hảo là số có tổng các ước số nhỏ hơn nó bằng chính nó. Nhập vào một số nguyên N từ bàn phím và kiểm tra số đó có là hoàn hảo không (6=1+2+3 là số hoàn hảo)

Bài tập 13: Cấu trúc lặp : Bài toán xử lý chuỗi tìm từ trong câu.

Yêu cầu: Viết chương trình cho nhập một câu. Tìm các từ và in in mỗi từ trên một dòng. Biết rằng các từ ngăn cách nhau bằng 1 hoặc một vài ký tự trắng .

Hướng dẫn:

- Xâu.charAt(int) để lấy ra kí tự tại vị trí trong chuỗi
- Xâu.len()- trả lại độ dài xâu
- isSpace(char) của lớp Character để kiểm tra khoảng trắng

Bài tập 14: Cấu trúc lặp +thao tác với xâu ký tự: Bài toán kiểm tra ký tự

Yêu cầu: Viết chương trình cho nhập một chuỗi và in ra các ký tự chữ cái (a-z hoặc A-Z) có trong chuỗi.

Hướng dẫn: dùng hàm *isLetter(char)* của lớp *Character* để kiểm tra chữ cái.

Bài tập 15: Cấu trúc điều khiển +Thao tác với xâu ký tự: Bài toán so sánh hai chuỗi

Yêu cầu: Viết chương trình cho nhập vào hai chuỗi, kiểm tra xem hai chuỗi có bằng nhau không, không phân biệt chữ hoa chữ thường- gợi ý sử dụng phương thức **xâu.equalsIgnoreCase(tên xâu so sánh)**

Bài tập 16: Cấu trúc điều khiển +Thao tác với xâu ký tự: Bài toán nối chuỗi

Yêu cầu: Viết chương trình cho nhập một chuỗi và một số nguyên n. Chương trình in ra một chuỗi mới là chuỗi được ghép từ n lần chuỗi đã nhập. Nếu n<2 thì xuất ra chuỗi gốc. Ví dụ nhập “hi” và 4, kết quả xuất ra “hihihihi”.

Bài tập 17: Cấu trúc điều khiển +Thao tác với xâu ký tự: Bài toán tách chuỗi

Yêu cầu: Viết chương trình cho nhập vào một chuỗi, tách chuỗi này thành các chuỗi con dựa vào khoảng trắng, xuất kết quả thành từng dòng. Ví dụ: nhập s = Xin Chao

Xuất kết quả: Xin

Chao

Hướng dẫn: Áp dụng thuật toán đếm số từ trong xâu.

Bài tập 18: Xây dựng hàm bên ngoài và gọi hàm trong hàm main(): bài toán PT bậc 2.

Yêu cầu: Viết chương trình giải phương trình bậc hai $ax^2 + bx + c = 0$ với a khác không

Hướng dẫn:

- Sử dụng lệnh lặp không do-while để nhập hệ số a, yêu cầu a khác không
- Thực hiện nhập các hệ số b,c. Giải và biên luận phương trình .

Bài tập 19: Xây dựng hàm bên ngoài và gọi hàm trong hàm main(): bài toán số fibonacci.

Yêu cầu: Viết chương trình xuất ra dãy số **Fibonacci**, Yêu cầu phải viết 2 phương thức. Phương thức 1 dùng để trả về số Fibonacci tại vị trí thứ k bất kỳ. Phương thức 2 dùng để xuất dãy số Fibonacci từ 1...n.

Dãy Fibonacci : 1 1 2 3 5 8 13 21 34 ...

- Số Fib thứ k = (số Fib thứ k-1) + (số Fib thứ k-2), 2 số Fib đầu tiên trong dãy luôn luôn là 1.

Chương 3: Chương 3: Lớp và các thành phần.

Bài tập 20: Xây dựng lớp: Bài toán hình chữ nhật.

Yêu cầu cũ: Thực hiện nhập vào chiều dài, chiều rộng hình chữ nhật. Tính và in ra chu vi, diện tích. Yêu cầu in ra có định dạng.

Yêu cầu mới:

Thực hiện: xây dựng lớp hình chữ nhật có các thuộc tính và phương thức phù hợp. In hình chữ nhật có định dạng tiêu đề và dữ liệu hiển thị riêng biệt

Gợi ý:Dựa vào yêu cầu bài toán ta thực hiện

- 2 danh từ mô tả tương ứng 2 thuộc tính cần khai báo: chiều dài, chiều rộng
- 5 hành động tương ứng 5 phương thức sẽ xây dựng: nhập dữ liệu; in dữ liệu; in tiêu đề; tính chu vi, tính diện tích

```
public class HCN {
```

```
    public static void main(String[] args) {
```

```
        float dai,rong, chuVi, dienTich;
```

```
        Random random = new Random();
```

```
        dai = random.nextFloat () + 10;
```

```
        rong = random.nextFloat () + 10;
```

```
        chuVi=(dai+rong)*2;
```

```
        dienTich=dai*rong;
```

```
        //in kết quả
```

```
        DecimalFormat d=new DecimalFormat("###,###.0#");
```

```
        System.out.printf("% 15.1f% 15.1f% 15s% 15s%n",
```

```
            dai,rong,d.format(tinhChuVi()),d.format(tinhDienTich());
```

```
        //in tiêu đề
```

```
        System.out.printf("% 15s% 15s% 15s% 15s%n",
```

```
            "Chiều dài ", "Chiều rộng ", " chu vi", "Diện tích");
```

```
    } }
```

Vậy Bài toán có lời giải như sau :

```
public class HìnhChuNhat {
```

```
    private float dai,rong;
```

```
    public float tinhDienTich(){
```

```
        return dai*rong;
```

```
    }
```

```
    public float tinhChuVi() {
```

```
        return (dai+rong)*2;
```

```
void nhap()
```

```
float tinhCV()
float tinhDT()
```

```
void xuat()
```

```
void inTieuDe()
```



```

    }
    void nhap() {
        Random random = new Random();
        dai = random.nextFloat() + 1;
        rong = random.nextFloat() + 1;
    }
    public static void inTieuDe(){
        System.out.printf("%15s%15s%15s%15s%n",
            "Chiều dài ", "Chiều rộng ", " chu vi", "Diện tích");
    }
    public void xuat() {
        System.out.printf("%15.1f%15.1f%15.1f%15.1f%n",
            dai, rong, tinhChuVi(), tinhDienTich());
    }
    public static void main(String[] args) {
        HìnhChuNhat h=new HìnhChuNhat();
        h.nhap();
        HìnhChuNhat.inTieuDe();
        h.xuat();
    }
}

```

Bài tập 21: Xây dựng lớp : Bài toán hình tròn

Yêu cầu :

- Xây dựng lớp Hình tròn có bán kính R ; các phương thức tính chu vi, diện tích, in tiêu đề ; in dữ liệu hình tròn.
- Xây dựng lớp chứa hàm main thực hiện minh họa nhập và in thông tin hình tròn đã nhập.

Bài tập 22: Xây dựng lớp : Bài toán phương trình bậc 2.

Yêu cầu: Xây dựng lớp phương trình bậc hai : $ax^2 + bx + c = 0$. Thực hiện gọi và minh họa lớp đã xây dựng

Hướng dẫn: Tạo lớp: PTB2 có :

- Các thuộc tính:
 - o Là các hệ số của phương trình
- Các phương thức:
 - o Phương thức nhập : nhập các hệ số, trong đó yêu cầu nhập a khác không
 - o Phương thức giải phương trình
 - o Phương thức in tiêu đề
 - o Phương thức in kết quả
- Tạo lớp PTB2demo thực hiện:
 - o Khai báo minh họa 1 đối tượng phương trình bậc . Yêu cầu đối tượng nhập thông tin các hệ số và in kết quả
 - o Mở rộng bài toán: nhập vào 1 danh sách phương trình bậc 2. In thông tin danh sách sau khi nhập.

Bài tập 23: Xây dựng lớp : Bài toán hồ sơ.

Yêu cầu: Cho các lớp dưới đây, cho biết kết quả chương trình và giải thích. Chú ý cách sử dụng phạm vi các thuộc tính và hàm.

```

// HoSo.java
class HoSo {
    static int soNguoi;
    String hoTen;
    HoSo( String ht ){

```

```

        hoTen = ht;
        soNguoi++;
    }
    static void tongKet(){
        System.out.println( "Ho khai nay co " + soNguoi + " nguoi" );
    }
    void xuatHoTen(){
        System.out.println( hoTen );
    }
}
//NhanSu.java
public class NhanSu
{
    public static void main( String argv[] ){
        HoSo n1 = new HoSo( "Tran Van Lang" );
        HoSo n2 = new HoSo( "Le Thi Binh Minh" );
        HoSo n3 = new HoSo( "Tran Thuy Thuc Trinh" );
        HoSo n4 = new HoSo( "Tran Thuy Anh Quynh" );
        HoSo.tongKet();
        n1.xuatHoTen();
        n2.xuatHoTen();
        n3.xuatHoTen();
        n4.xuatHoTen();
    }
}

```

Bài tập 24: Xây dựng lớp: Bài toán sinh Viên

Viết chương trình theo hướng đối tượng quản lý sinh viên đơn giản. Sinh viên có:

- Các thuộc tính riêng tư (private):
 - o Mã sinh viên – số nguyên
 - o Họ tên: chuỗi ký tự
 - o Điểm LT, điểm TH: số thực
- Phương thức:
 - o hàm tạo mặc định khởi gán giá trị mặc định cho các thuộc tính
 - o Hàm tạo có đối gán đầy đủ thông cho các thuộc tính
 - o Hàm getter/setter các thuộc tính
 - o Phương thức static inTieuDe diễn tả tiêu đề in của đối tượng.
 - o Xây dựng phương thức inDuLieu(). In nội dung dữ liệu tương ứng với tiêu đề.
- Xây dựng hàm main tạo 3 đối tượng sinh viên
 - o Sv1 chứa thông tin của mình bằng hàm tạo đủ thông số, thông tin truyền vào cho hàm tạo nhập vào từ bàn phím
 - o Sv2 thông tin được tạo bằng hàm đủ thông số với các giá trị cố định do người dùng mình họa.
 - o Sv3 tạo bằng hàm tạo mặc định. Nhập các thông tin vào từ bàn phím và gọi hàm set để gán giá trị cho các thuộc tính
 - o In bản danh sách sinh viên gồm 4 cột: mã sinh viên, họ tên, điểm lý thuyết, thực hành, điểm trung bình. Bảng có 3 dòng cho 3 sinh viên
- Mở rộng bài toán:
 - o nhập vào 1 danh sách (mảng) sinh viên. In danh sách sinh viên dạng bảng.
 - o Đưa ra sinh viên có điểm trung bình lớn nhất trong danh sách.

Bài tập 25: Xây dựng lớp: Bài toán quản lý đăng ký xe.

Sở giao thông cần theo dõi việc đăng ký xe của người dân. Dựa vào thông tin trị giá xe và dung tích xylanh của xe, sở giao thông cũng tính mức thuế phải đóng trước bạ khi mua xe như sau:

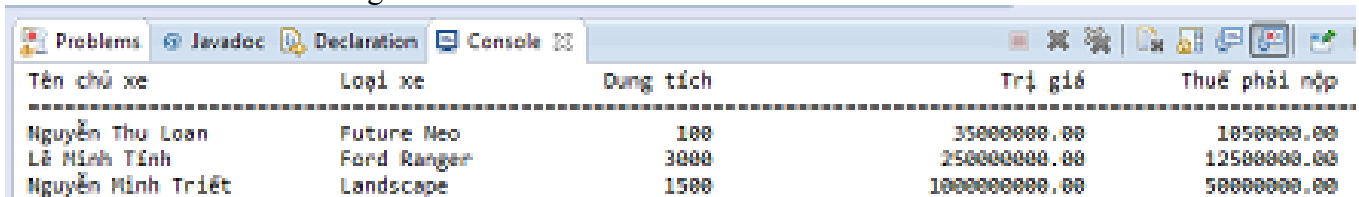
- Dưới 100cc, 1% trị giá xe.
- Từ 100 đến 200cc, 3% trị giá xe.
- Trên 200cc, 5% trị giá xe.

1. Hãy thiết kế và cài đặt class Vehicle với các thuộc tính và phương thức phù hợp. Lớp phải có các hàm tạo và phải bảo đảm tính đóng gói.

2. Xây dựng class chứa hàm main. Hàm main in ra menu lựa chọn các công việc:

- Nhập thông tin và tạo danh sách các
- Xuất bảng kê khai tiền thuế trước bạ của các xe.
- . Thoát.

Mẫu thiết kế xuất của chương trình:



Tên chủ xe	Loại xe	Dung tích	Trị giá	Thuế phải nộp
Nguyễn Thu Loan	Future Neo	188	35000000.00	1850000.00
Lê Minh Tính	Ford Ranger	3000	250000000.00	12500000.00
Nguyễn Minh Triết	Landscape	1500	1000000000.00	50000000.00

Hướng dẫn: Căn cứ vào mẫu kết xuất của chương trình ta tạo lớp Xe có :

- **Danh từ thuộc tính mô tả:**
 - o Tên chủ xe, loại xe, dung tích, trị giá
- Danh từ phát sinh: thuế
- Động từ thực hiện:
 - o tính thuế dựa vào dung tích xi lanh vậy ta viết phương thức tính thuế thuộc lớp xe theo công thức đã cho.
 - o Nhập thông tin xe; in thông tin xe; in có tiêu đề: xây dựng thêm phương thức nhập thông tin xe; phương thức xuất tiêu đề; phương thức in dữ liệu
- Căn cứ yêu cầu hàm main ta thực hiện:
 - o Tạo lớp Xedemo có 1 thuộc tính là danh sách xe. Vì danh sách sẽ đc dùng trong hàm main nên nó có thêm chỉ thị static.
 - o Xây dựng phương thức nhập danh sách xe cũng có chỉ thị static vì thao tác trên ds xe
 - o Xây dựng phương thức inDL cũng static gọi xe in tiêu đề và in ds xe dùng lặp for.
 - o Xây dựng hàm main có cấu trúc lặp while hoặc do-while chọn công việc bằng cách gọi hàm tương ứng thông qua biến chọn.

Bài tập 26: Xây dựng lớp: Bài toán Nhân viên và hàm thành phần.

(a) Xây dựng lớp NhanVien như sau: **NhanVien.java**

Các thuộc tính :

- o mã nhân viên - maNV; số sản phẩm - soSP .

Các phương thức:

- o khởi dựng không đối : NhanVien()
- o khởi dựng có đối: NhanVien (ma: String, sp : int)
- o get/set thuộc tính mã, số sản phẩm : getMaNV() ;setMaNV(ma : String) ; getSoSP() ; setSoSP(sp : int).

- kiểm tra vượt chuẩn: `coVuotChuan()` : boolean
- tổng kết: `getTongKet()` : String
- lấy lương: `getLuong()`: double
- In tiêu đề: `XuatTieuDe()` : static void
- kết xuất đối tượng dạng xâu: `toString()` : String
- Trong các phương thức khởi tạo và các hàm `setSoSP`, khi gán giá trị cho thuộc tính `soSP` thì cần kiểm tra giá trị đó có phải là số dương không, nếu là số dương thì mới gán giá trị cho thuộc tính, ngược lại thì gán bằng 0.
- `coVuotChuan()` : trả về true nếu `soSP > 500`, ngược lại trả về false. hàm này dùng để kiểm tra xem số lượng sản phẩm của nhân viên có vượt quá số lượng chuẩn hay không.
- `getTongKet()` : trả về chữ "Vượt" khi `soSP > 500`, ngược lại để trống (có thể sử dụng hàm `coVuotChuan()` để kiểm tra).
- `getLuong()` : trả về lương của một nhân viên, lương ăn theo sản phẩm với đơn giá cơ bản cho 1 sản phẩm là 20.000, và nếu số sản phẩm của nhân viên vượt chuẩn thì phần vượt chuẩn được tính đơn giá là 30.000.
- `XuatTieuDe()`: xuất tiêu đề gồm: mã nhân viên, số sản phẩm, lương, tổng kết.
- `inDL()` : in dữ liệu tương ứng với tiêu đề để tạo khoảng cách phù hợp

(b) Viết hàm main để kiểm tra lớp NhanVien theo yêu cầu sau:

- Tạo 2 nhân viên với các thuộc tính cho người dùng nhập vào.
- Xuất ra các thông tin của họ, gồm mã, số sản phẩm, lương, tổng kết và cho biết nhân viên có vượt chuẩn không.

(c). Mở rộng bài toán:

- Nhập vào 1 mảng Nhân viên, in mảng sau khi nhập.
- In danh sách nhân viên vượt chuẩn.

Bài tập 27: Xây dựng lớp: Bài toán TamGiac:

Yêu cầu: Xây dựng lớp Tam giác có các mô tả cạnh là **ma,mb,mc** .

- Xây dựng phương thức nhập các giá trị của cạnh vào từ bàn phím. Yêu cầu các cạnh nhập vào phải thỏa mãn là các cạnh của tam giác.
- Tính chu vi, diện tích tam giác.

Vậy ta có lớp tam giác được mô tả như sau:

TamGiac.java

```
- ma : int
- mb : int
- mc : int
+ TamGiac(a: int, b: int, c: int)
+ getCanhA() : int
+ setCanhA(v : int) : void
+ getCanhB() : int
+ setCanhB(v : int) : void
+ getCanhC() : int
+ setCanhC(v : int) : void
+ laTamGiac(int, int, int) : boolean
+ getChuVi() : int
```

+ getDienTich() : double

Hướng dẫn thực hiện:

- Các giá trị lập thành một hình tam giác khi và chỉ khi các cạnh lớn hơn 0 và tổng hai cạnh bất kỳ luôn lớn hơn cạnh còn lại.
- setCanhA, setCanhB, setCanhC : cũng yêu cầu phải kiểm tra giá trị gán có là số dương và lập thành tam giác hay không, nếu không thì không gán (giữ lại giá trị cũ).
- getChuVi(), getDienTich(): tính chu vi và diện tích của tam giác.
- laTamGiac(int, int, int): trả về giá trị true khi ba giá trị a, b, c lập thành một hình tam giác, ngược lại trả về giá trị false. Các hàm hỗ trợ thực hiện yêu cầu bài toán.

Viết hàm main để kiểm tra lớp CHinhTamGiac theo yêu cầu sau:

- Cho nhập vào 3 giá trị số nguyên và khởi tạo hình tam giác có các cạnh ứng với 3 giá trị này, nếu 3 giá trị này không lập thành tam giác thì thông báo, ngược lại thì tính và xuất ra thông tin tam giác, giá trị chu vi và diện tích của nó.
- Mở rộng bài toán: Nhập vào 1 mảng tam giác, in mảng sau khi nhập

Bài tập 28: Xây dựng lớp: Bài toán nhân viên sản xuất.

Yêu cầu:

a. Xây dựng lớp CNhanVien biết các danh từ mô tả nhân viên bao gồm: mHo, mTen, mSoSP lần lượt là các thuộc tính họ, tên và số sản phẩm của nhân viên.

Các hành động:

- khởi tạo CNhanVien(String, String, int), hàm này sẽ khởi tạo họ, tên, số sản phẩm của nhân viên; hàm phải kiểm tra số sản phẩm là số lớn hơn hoặc bằng 0, nếu là số âm thì gán giá trị cho mSoSP bằng 0.
- Các hành động lấy và gán giá trị cho thuộc tính của lớp (các hàm get/set).
- Hành động getLuong() để tính lương cho nhân viên, lương = số sản phẩm * đơn giá, với đơn giá tùy thuộc vào số sản phẩm như bảng bên.
- Hành động soSanhNV(CNhanVien nv2): hàm này trả về giá trị true khi số sản phẩm (mSoSP) lớn hơn số sản phẩm của nv2, ngược lại trả về false

Số sản phẩm	Đơn giá
1 - 199	0.5
200 - 399	0.55
400 - 599	0.6
600 trở lên	0.65

b. Viết hàm main sử dụng lớp CNhanVien theo yêu cầu sau:

- Cho người dùng nhập vào 2 nhân viên, mỗi nhân viên nhập vào họ, tên, số sản phẩm của họ. Hãy tính và xuất ra lương của từng nhân viên.
- So sánh và xuất ra thông báo nhân viên nào có số sản phẩm nhiều hơn và nhiều hơn bao nhiêu. Dùng 2 cách so sánh: dùng soSanhNV() và không dùng soSanhNV().
- Mở rộng : nhập vào 1 danh sách nhân viên và in danh sách nhân viên. Sắp xếp danh sách nhân viên theo chiều tăng dần của lương đã nhận

Bài tập 29: Xây dựng lớp: Bài toán Sinh viên

Yêu cầu:

- Xây dựng lớp SinhVien biết rằng sinh viên được mô tả thông qua : mã sinh viên, họ tên, điểm lý thuyết, điểm thực hành.

Thực hiện các nhiệm vụ: phương thức :

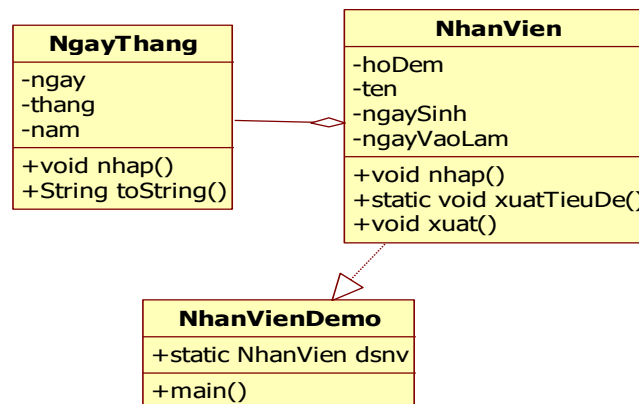
- khởi tạo không đối- khởi gán các giá trị mặc định cho thuộc tính
- khởi có đầy đủ các đối- khởi gán giá trị như truyền vào
- nhập()- nhập thông tin Sinh viên và tính luôn điểm trung bình= (đlt+dth)/2).
- toString()- kết xuất xâu mô tả Sinh Viên

Viết hàm main cho phép thực hiện:

- Nhập 1 danh sách sinh viên
- Xuất thông tin sinh viên và cho biết kết quả học tập của sinh viên (Đậu khi điểm trung bình ≥ 5 , ngược lại là Rớt).
- Sắp xếp danh sách sinh viên theo tên. Tên trùng nhau sắp xếp theo điểm lý thuyết
- Tìm sinh viên có điểm trung bình cao nhất, sinh viên có điểm trung bình thấp nhất

Bài tập 30: Xây dựng lớp: Cài đặt quan hệ kết tập

- Thông tin nhân viên trong công ty được quản lý thông qua họ đệm, tên nhân viên, ngày sinh và ngày vào làm.
- Thông tin về ngày tháng được mô tả thông qua ngày, tháng, năm. Yêu cầu tháng phải nằm trong khoảng từ 1-12 và ngày phải phù hợp với tháng tương ứng.
- Hãy mô tả sơ đồ và cài đặt lớp thực hiện yêu cầu minh họa sử dụng lớp nhập vào 1 nhân viên và in ra nhân viên đó sau khi nhập



DS nhan vien sau khi nhap:			
Ho dem	Ten	Ngay sinh	Ngay vao lam
vu van h	f	7/8/8	6/7/8
uy	uj	8/8/9	7/7/7
98	98	9/9/9	8/8/8

// NgayThang.java

```
public class NgayThang {
    private int thang; // 1-12
    private int ngay; // 1-31 based on month
    private int nam; // any year
    public NgayThang( int thang1 , int ngay1, int nam1 ) {
        if ( thang1 > 0 && thang1 <= 12 ) // tháng hợp lệ
            this.thang = thang1;
        else
        {
            this.thang = 1;
            System.out.println( "Tháng " + thang1 + " không hợp lệ. Đặt lại tháng =1." );
        }
        nam = nam1; // could validate year
    }
}
```

```

        ngay = checkDay( ngay1 ); // kiểm tra ngày hợp lệ
        System.out.println( "in thông tin ngày tháng " + toString() );
    }
    private int checkDay( int ngayKiemTra ) {
        int ngayTrongThang[]={0,31,28,31,30,31,30,31,31,30,31,30,31};
        // kiểm tra nếu ngày tháng hợp lệ
        if (ngayKiemTra > 0 && ngayKiemTra <= ngayTrongThang [ thang ] )
            return ngayKiemTra;
        else
            if (((thang==2) && (ngayKiemTra == 29)) &&
                ((nam % 400 == 0) || ((nam % 4 == 0) && (nam % 100 != 0 ))))
                return ngayKiemTra;
            else {
                System.out.println( "ngày " + ngayKiemTra + " không hợp lệ, đặt lại =1." );
                return 1;
            }
    }
    //end of checkDay()
    //ghi đè phương thức toString để mô tả tháng/ngày/năm
    public String toString() { return thang + "/" + ngay + "/" + nam; }
} // end of class NgayThang

// Tạo mới lớp NhanVien.java
public class NhanVien
{
    private String hoDem;
    private String ten;
    private NgayThang ngaySinh;
    private NgayThang ngayVaoLam;
    public NhanVien () {
        hoDem=""; ten="";
        ngaySinh =new NgayThang(0,0,0);
        ngayVaoLam=new NgayThang(0,0,0);
    }
    public NhanVien(String hoDem, String ten, NgayThang ngaySinh, NgayThang ngayVaoLam) {
        this.hoDem = hoDem; this.ten = ten;
        this.ngaySinh = ngaySinh;
        this.ngayVaoLam = ngayVaoLam;
    }
    //sinh viên tự xây dựng các hàm cần thiết khác để thực hiện được kết quả như yêu cầu kết xuất trên
    public String toString() {
        return ten + ", " + hoDem + " ngày vào lam: " + ngayVaoLam + " sinh ngày: " + ngaySinh;
    }
}

//NhanVienDemo.java
public class NhanVienDemo {
    public static void main(String args[]) {
        NgayThang ngaySinh = new NgayThang ( 7, 24, 1991 );
        NgayThang ngayVaoLam = new NgayThang( 3, 12, 1998 );
        NhanVien nv= new NhanVien ( "Vu thi", "Lan anh", ngaySinh, ngayVaoLam );
        //gọi hàm hiển thị message dialog trong thư viện javax.swing.JOptionPane;
        JOptionPane.showMessageDialog( null, nv.toString(), "minh họa lớp Employee",
        JOptionPane.INFORMATION_MESSAGE );
        System.exit( 0 ); //thoát khỏi chương trình
    }
}

//thử cài đặt lại ngày sai và tháng sai, xem bạn nhận được thông báo gì?.
Mở rộng: Nhập vào 1 danh sách nhân viên từ bàn phím. In danh sách nhân viên sau khi nhập ra màn hình
theo dạng bảng dữ liệu. Xây dựng lớp và hàm thành phần.

```


Bài tập 31: Quan hệ kết tập: Bài toán Sinh viên – môn học

Cho các lớp có mẫu như sau: :

Date
ngay:int thang:int nam:int --- void nhap() void xuat()

Mon
tenMon:string soHocTrinh:int hocKy:string --- void nhap() void xuat()

SinhVien
hoTen:string ngaySinh>Date monHoc:Mon diemThi:float ---- void nhap() void inTieuDe() void xuat()

- Trong lớp **SinhVien**: ngaySinh có kiểu **Date**, **monHoc** là danh sách các môn đã học, mỗi môn học có kiểu **Mon**. Hãy vẽ các liên 1. kết biểu diễn mối quan hệ giữa các lớp trên sao cho mỗi sinh viên học nhiều môn, và mỗi môn có 1 điểm tổng kết cuối cùng.
- Cài đặt sơ đồ lớp trên theo hướng đối tượng.
- Xây dựng lớp chứa hàm main nhập vào 1 sinh viên và các môn sinh viên học. Sắp xếp danh sách môn học theo chiều tăng dần của Điểm thi.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Cho mô tả của hóa đơn bán hàng như sau:

Mã HD: 001		Ngày bán: 12/2/2015	
Khách hàng: Vũ văn A		Địa chỉ: Hà Nội	
Tên hàng	đơn giá	số lượng	Thành tiền
Áo	200	152	352
Tất	10	256	266
....			
Tổng tiền		408	618

- Tìm các cụm danh từ, và các danh từ mô tả cụm danh từ phát hiện các lớp
- Tìm công việc cần thực hiện trên các danh từ để phát hiện các phương thức phù hợp
- Vẽ sơ đồ liên kết các lớp
- Cài đặt sơ đồ theo hướng đối tượng.
- Thực hiện nhập vào 1 danh sách hóa đơn
- In danh sách hóa đơn theo mẫu (không kẻ khung)
- Sắp xếp danh sách hàng hóa trong hóa đơn theo chiều tăng dần tên hàng. Tên hàng trùng nhau sắp tăng dần theo thành tiền

- **Hàng hóa** sở hữu thông tin: tên hàng, số lượng, đơn giá. Thành tiền = số lượng * đơn giá
- **Hóa đơn** bao gồm: mã hóa đơn, ngày lập, khách mua hàng và danh sách hàng hóa.
- **Cách thực hiện:** Xây dựng 2 lớp **khách hàng**, **hàng hóa** trước. sau đó xây dựng lớp **hóa đơn** ngoài các thuộc tính của nó là: mã hóa đơn và ngày bán, nó còn có thuộc tính khách hàng là đối tượng lớp Khách hàng, ds hàng hóa là tập hợp gồm các đối tượng lớp Hàng Hóa.

[illegible]

Gợi ý thực hiện:

```
public class KháchHang {
    private String hoten,diaChi;
    public void nhap()
    {
        Scanner s=new Scanner(System.in);
        System.out.print("nhap ho ten=");
        hoten=s.nextLine();
        System.out.print("nhap dia chi=");
        diaChi=s.nextLine();
    }
    public String toString() {
        return "Khach hang: " + hoten + ",\t diaChi:" + diaChi ;
    }
}

public class Hang {
    private String tenHang;
    private float soLuong,donGia;
    public void nhap() {
        Scanner s=new Scanner(System.in);
        System.out.print("nhap ten hang="); tenHang=s.nextLine();
        System.out.print("nhap so luong="); soLuong=s.nextFloat();
        System.out.print("nhap don gia="); donGia=s.nextFloat();
    }
    public float tinhTongTien(){
        return soLuong*donGia;
    }
    static void inTieuDe(){
        System.out.printf("%10s %10s %10s %15s %n","ten hang", "so luong","don gia","thanh tien");
    }
    public void inDL(){
        System.out.printf("%10s %10.1f %10.1f %15.0f %n",
tenHang,soLuong,donGia,tinhTongTien());
    }
}

public class HoaDon {
    private String soHD,ngayHD;
    private int soMatHang;
    private KháchHang kh= new KháchHang();
    private Hang dsHang[];
    private float tongTienHang=0;
    public void nhap() {
        Scanner s=new Scanner(System.in);
        System.out.print("nhap so HD="); soHD=s.nextLine();
        System.out.print("nhap ngay lap="); ngayHD=s.nextLine();
        System.out.print("nhap khách hàng:"); kh.nhap();
        System.out.println("nhập danh sách hàng:");

        System.out.print("nhap so mat hang="); soMatHang=s.nextInt();

        dsHang=new Hang[soMatHang];//xin cấp phát mảng
        for (int i=0;i<soMatHang;i++)
        {
            System.out.println("nhap mat hang thứ " +i+1);
            dsHang[i]=new Hang();
            dsHang[i].nhap();
            tongTienHang+=dsHang[i].tinhTongTien();
        }
    }
    public void inHD() {
        DecimalFormat df=new DecimalFormat("###,###.0#");
```

```

        //in số hóa đơn, ngay lập
        System.out.println("SốHD:" + soHD + "\t\t ngay lap"+ngayHD);
        System.out.println(kh);
        System.out.println("Danh sách hàng hóa");
        Hang.inTieuDe();
        for (int i=0;i<soMatHang;i++)
            dsHang[i].inDL();
        System.out.println("-----");
        System.out.println("Tổng tiền:"+df.format(tongTienHang));
    } //end of function in hóa đơn
}
/**
 * Sắp xếp danh sách hàng hóa theo 1 tiêu chí bất kỳ
 */
public void sapXep(){
    Comparator<Hang> c=new Comparator<Hang>() {
        /**
         * @param t
         * @param t1
         * @return số dương nếu t>t1
         * số âm nếu t<t1
         * =0 nếu t=t1
         * /so sánh xâu thì dùng phương thức xâu1.compareToxxxx(xâu2)
         * nếu so sánh số: lớp_bao_boc.compare(số 1, số 2)
         */
        @Override
        public int compare(Hang t, Hang t1) {
            //sắp theo tên
            // return t.getTenHang().compareToIgnoreCase(t1.getTenHang());
            //sắp theo số lượng
            return Float.compare(t.getSoLuong(),t1.getSoLuong());
        }
    };
    Arrays.sort(dsHang, c);
    System.out.println("so luong min:");
    Collections.min(Arrays.asList(dsHang),c).inDL();
} //end of function sắp xếp
} //end of classs
//tạo file HoaDonDemo.java và thực hiện cài đặt lớp HoaDonDemo
public class HoaDonDemo {
    public static void main(String[] args) {

        // mô phỏng trong trường hợp nhập nhiều hóa đơn
        HoaDon dshd[];
        int n;
        System.out.print("nhập số lượng hóa đơn n =");
        Scanner s=new Scanner(System.in);
        n=s.nextInt();
        //xin cấp phát cho n hóa đơn
        dshd=new HoaDon[n];
        for(int i=0;i<n;i++){
            System.out.println("nhập hóa đơn thu "+i);
            HoaDon hd=new HoaDon();
            hd.nhap();
            dshd[i]=hd;
            System.out.println("-----");
        }
        for(int i=0;i<n;i++){
            System.out.println("In hóa đơn thu "+i);
            dshd[i].inHD();
        }
    }
}

```

}

Bài tập 33: Quan hệ kết tập: Bài toán sinh viên – điểm

Viết chương trình quản lý điểm của sinh viên. Biết mỗi sinh viên có các thông tin: Mã sinh viên, Tên sinh viên, Lớp học và Môn học. Một sinh viên chỉ thuộc 1 Lớp học và học nhiều môn học.

Thông tin về Lớp học bao gồm: Tên lớp, khoá. Thông tin về môn học bao gồm: Tên môn, số trình, điểm. Yêu cầu chương trình có các chức năng sau:

- Nhập thông tin cho n sinh viên sao cho mỗi sinh viên có đủ thông tin.
- In ra danh sách các SV vừa nhập gồm các thông tin: mã SV, tên SV, Tên lớp, Khoá.
- In phiếu báo điểm cho từng sinh viên theo mẫu:

**Phiếu
báo
điểm**

PHIẾU BÁO ĐIỂM		
Mã sinh viên: <i>SV001.</i>	Tên sinh viên: <i>Nguyễn Hải Hà</i>	
Lớp: <i>Tin 2</i>	Khoá: <i>52 ;</i>	Điểm TB: <i>8.17</i>
Bảng điểm:		
Tên môn	Số trình	Điểm
Cơ sở dữ liệu	4	8
Lập trình HDT	3	7
Hệ điều hành	5	9

**Sinh
viên**

**Môn
học**

$$= \sum(\text{Số trình} * \text{Điểm}) / \sum(\text{Số trình})$$

Bài tập 34: Quan hệ kết tập: Bài toán Quản lý sách

Cho sơ đồ như dưới đây. Viết chương trình chính nhập vào thông tin của k phiếu nhập (chú ý: mỗi phiếu nhập có nhiều sách nhập (n)). In ra thông tin của các phiếu nhập theo mẫu:

PHIẾU NHẬP SÁCH			
Mã phiếu: <i>PH001.</i>	Ngày lập: <i>1/1/2007</i>		
Tổng số lượng nhập về:	<i>300</i>		
Chi tiết:			
Tên sách	Tên tác giả	Tên NXB	Số lượng
Cơ sở dữ liệu	Nguyễn Văn Ba	Thống kê	100
Lập trình HDT	Phạm Văn ắt	Thống kê	150
Hệ điều hành	Đỗ Ngọc Tân	GTVT	50

Bài tập 35: Quan hệ kế thừa: Đa hình thông qua kế thừa- bài toán hình phẳng

- Viết lớp **Shape**, lớp này chỉ có một hàm là **draw()** không có tham số và không trả về giá trị, chỉ xuất ra dòng chữ “Drawing a Shape”.
- Viết lớp **Rectangle** kế thừa từ lớp **Shape**. Thêm vào lớp **Rectangle** hai thuộc tính là chiều dài và chiều rộng (số nguyên). Viết các hàm khởi tạo và các hàm get/set cho lớp này, chú ý giá trị gán cho 2 thuộc tính nằm trong [1, 15].
 - Trong lớp **Rectangle**, viết lại hàm **draw()** của lớp **Shape** để vẽ ra hình chữ nhật bởi các dấu *, với số lượng dấu * mỗi cạnh tùy thuộc vào các thuộc tính chiều dài và chiều rộng. Ví dụ thuộc tính chiều dài và chiều rộng lần lượt là 7 và 3 thì hàm draw() vẽ được hình: vuông bằng dấu *.
- Viết lớp **RightTriangle** kế thừa từ lớp **Shape**. Thêm vào lớp **RightTriangle** một thuộc tính là cạnh vuông (số nguyên). Viết các hàm khởi tạo và các hàm **get/set** cho lớp này, chú ý giá trị gán cho thuộc tính nằm trong [1, 20].

Bài tập 37: Quan hệ kế thừa: Bài toán quản lý tiền điện.

Xây dựng chương trình quản lý danh sách hoá đơn tiền điện của khách hàng. Thông tin bao gồm các loại khách hàng :

Khách hàng Việt Nam: mã khách hàng, họ tên, ngày ra hoá đơn (ngày, tháng, năm), đối tượng khách hàng (sinh hoạt, kinh doanh, sản xuất): số lượng (số KW tiêu thụ), đơn giá, định mức. Thành tiền được tính như sau:

- Nếu số lượng \leq định mức thì: thành tiền = số lượng * đơn giá.
- Ngược lại thì: thành tiền = số lượng * đơn giá * định mức + số lượng KW vượt định mức * Đơn giá * 2.5.

Khách hàng nước ngoài: mã khách hàng, họ tên, ngày ra hoá đơn (ngày, tháng, năm), quốc tịch, số lượng, đơn giá. Thành tiền được tính = số lượng * đơn giá.

Thực hiện các yêu cầu sau:

- Xây dựng các lớp với chức năng thừa kế.
- Nhập xuất danh sách các hóa đơn khách hàng.
- Tính tổng số lượng cho từng loại khách hàng.
- Tính trung bình thành tiền của khách hàng người nước ngoài
- Xuất ra các hoá đơn trong tháng 09 năm 2013 (của cả 2 loại khách hàng).

Bài tập 38: Quan hệ kế thừa: Bài toán quản lý khách sạn.

Một khách sạn X cần quản lý các hóa đơn của khách hàng thuê phòng. Hóa đơn có 2 loại: hóa đơn theo giờ, hóa đơn theo ngày. Thông tin chung của chi tiết hóa đơn là: Mã hóa đơn, ngày hóa đơn (ngày, tháng, năm), Tên khách hàng, mã phòng, đơn giá. Thông tin riêng của từng loại hóa đơn gồm:

- Hóa đơn theo giờ còn có số giờ thuê. Thành tiền = số giờ thuê * đơn giá. Nếu trường hợp số giờ > 24 tiếng và < 30 tiếng thì cũng chỉ tính 24 giờ. Nếu trường hợp số giờ là > 30 tiếng thì không dùng loại hóa đơn theo giờ.
- Hóa đơn theo ngày sẽ có số ngày thuê. Thành tiền = số ngày thuê * đơn giá. Nếu số ngày > 7 thì giảm 20% đơn giá cho những ngày còn lại.

Thực hiện các yêu cầu sau:

- Xây dựng các lớp với chức năng thừa kế.
- Nhập xuất danh sách các hóa đơn thuê phòng.
- Tính tổng số lượng cho từng loại thuê phòng.
- Tính trung bình thành tiền của hóa đơn thuê phòng trong tháng 9/2013.

Bài tập 39: Quan hệ kế thừa: Bài toán Quản lý Kho

Hàng hóa quản lý trong kho của một siêu thị gồm có hàng thực phẩm, hàng sành sứ và hàng điện máy. Mỗi loại hàng đều có:

- Mã hàng (không được sửa, không được để trống),
- Tên hàng (không được rỗng),
- Số lượng tồn (≥ 0),
- Đơn giá (> 0).
- Hàng thực phẩm thì cần quan tâm đến thông tin ngày sản xuất, ngày hết hạn (ngày hết hạn

phải sau hoặc là ngày sản xuất) và nhà cung cấp.

- Hàng điện máy cần biết thời gian bảo hành bao nhiêu tháng (≥ 0), công suất bao nhiêu KW (> 0).
- Hàng sành sứ thì cần biết thông tin về nhà sản xuất và ngày nhập kho.

Ngoài ra, người quản lý cần quan tâm đến số lượng tồn kho và các yếu tố khác của từng loại hàng hóa để đánh giá mức độ bán buôn, tiền VAT từng loại hàng hóa. Biết rằng VAT của hàng điện máy và sành sứ là 10%, VAT của hàng thực phẩm là 5%.

Dựa vào các thông tin trên, **hãy xác định**:

- Các lớp có thể có. Lớp nào là lớp trừu tượng (abstract class), lớp nào là lớp cụ thể
- Các thuộc tính cho từng lớp.
- Các phương thức cho từng lớp (phương thức nào là phương thức trừu tượng (abstract method), danh sách các tham số có thể có cho từng phương thức và kiểu trả về của phương thức).
- Thiết kế mô hình class (xây dựng cây thừa kế, các giao diện nếu có).
- Tạo một project. Thực hiện cài đặt tường minh cho mỗi loại hàng cụ thể trên. Trong đó, để đánh giá mức độ bán buôn thì:
 - Hàng điện máy, nếu số lượng tồn kho < 3 thì được đánh giá là bán được.
 - Hàng thực phẩm, nếu vẫn còn tồn kho và bị hết hạn thì đánh giá là khó bán.
 - Hàng sành sứ, nếu số lượng tồn kho > 50 và thời gian lưu kho > 10 ngày thì đánh giá là bán chậm.
- Các trường hợp còn lại xem như không đánh giá.

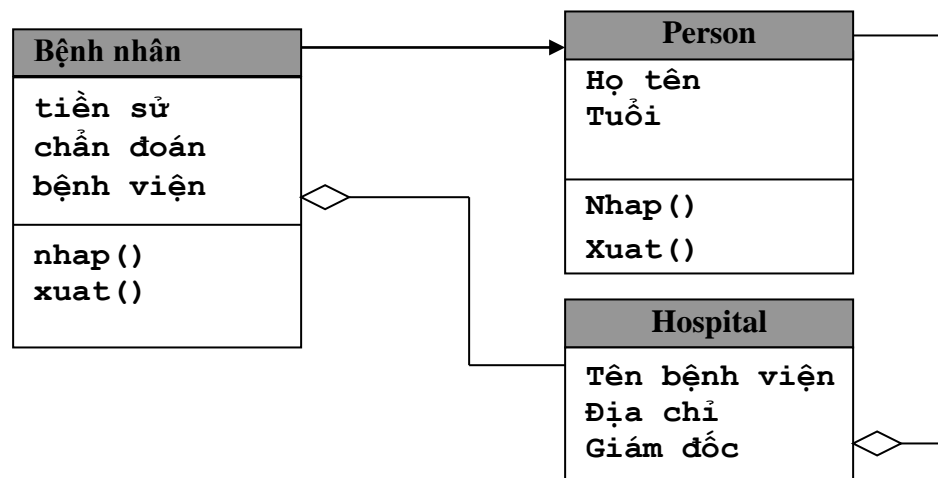
Hãy viết lớp quản lý danh sách hàng hóa.

- Dùng mảng hoặc danh sách để lưu trữ danh sách hàng hóa.
- Viết phương thức thêm một hàng hóa vào danh sách (thêm thành công nếu không trùng mã)
- Viết phương thức in toàn bộ danh sách các hàng hóa theo từng loại.

Tạo lớp cho phần thử nghiệm, với menu lựa chọn theo yêu cầu nhập xuất hàng hóa như thiết kế.

Bài tập 40: Quan hệ kế thừa - kết tập: Bài toán bệnh viện

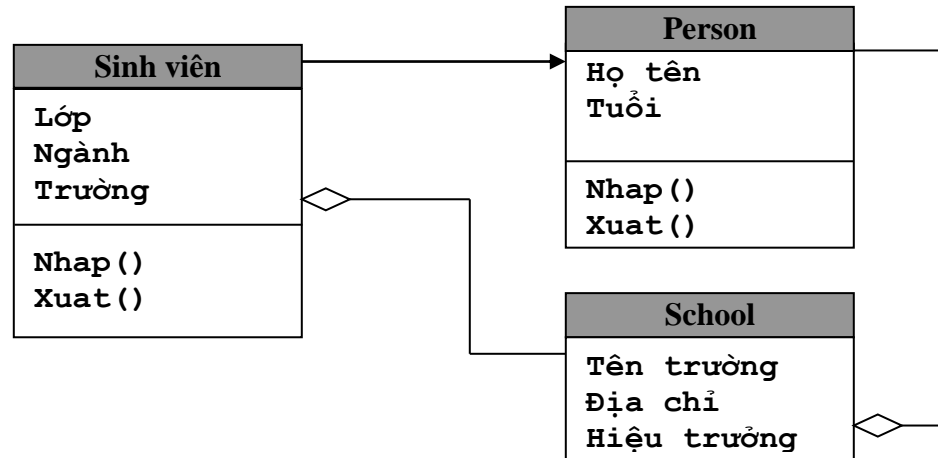
Hãy cài đặt các lớp theo sơ đồ thiết kế sau



Viết chương trình chính nhập vào danh sách n bệnh nhân, in ra các bệnh nhân của bệnh viện Bạch Mai.

Bài tập 41: Quan hệ kế thừa- kết tập: Bài toán quản lý sinh viên

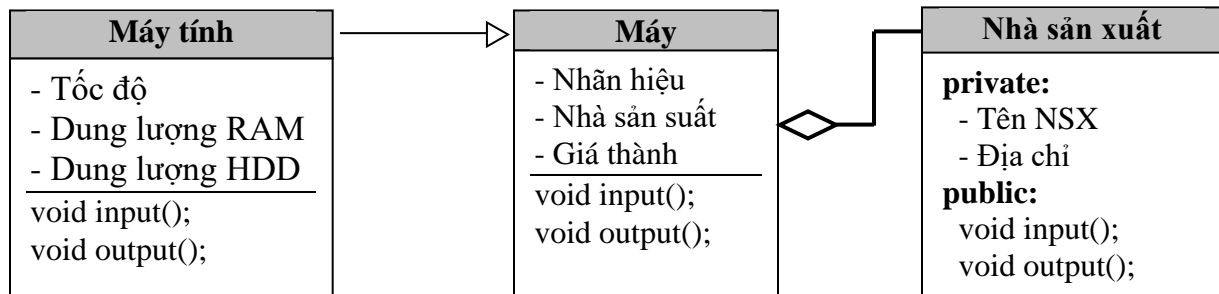
Hãy cài đặt các lớp theo sơ đồ thiết kế sau



Viết chương trình chính nhập vào danh sách n sinh viên, in ra các sinh viên của trường ĐHCNHN

Bài tập 42: Quan hệ kế thừa - kết tập: Bài toán Quản lý máy tính

Cài đặt các lớp theo biểu đồ sau:



- Với input và output là các phương thức nhập, xuất thông tin của các thuộc tính của lớp).
Viết chương trình chính nhập vào danh sách n máy tính. In ra thông tin của các máy tính của nhà sản xuất Intel. Sắp xếp danh sách các máy tính theo chiều giảm dần của giá thành và in danh sách đã sắp ra màn hình. Cho biết giá thành trung bình của mỗi chiếc máy tính?

Bài tập 43: Quan hệ kế thừa- lớp trừu tượng :Bài toán hình phẳng

Yêu cầu: Hãy mô tả bài toán hình học.

- Tìm cha cho các hình Vuông, chữ nhật, tròn. Tìm các thuộc tính chung và các phương thức chung trong đó có 2 phương thức tính chu vi và tính diện tích chưa có cách thực hiện rõ ràng
- Xây dựng các lớp con kế thừa cụ thể hóa phương thức trừu tượng kể trên.

Hướng dẫn:

```
abstract Class HinhPhang{
    abstract float tinhCV();
    abstract float tinhDT();
}
class HinhVuong extends HinhPhang {
    public float canh;
    public float tinhCV() {
        return 4*canh;
    }
    public float tinhDT() { return canh*canh; }}
```

- Tương tự cài đặt các lớp Hình tròn, vuông, chữ nhật cũng là trừu tượng và cụ thể hóa phương thức tính chu vi và diện tích
- Viết hàm main minh họa: khai báo mỗi lớp con kế thừa một đối tượng, nhập, xuất dữ liệu của chúng.

Bài tập 44: Quan hệ kế thừa- lớp trừu tượng : Bài toán quản lý nhân viên

Yêu cầu: Một công ty sản xuất có 2 loại nhân viên là nhân viên sản xuất và nhân viên văn phòng. Hai loại nhân viên được quản lý thông qua các thông tin: Họ tên; Năm vào làm, lương, phụ cấp. Lương của hai loại nhân viên được tính như sau:

- Nhân viên sản xuất: lương = sản phẩm x 10000.
- Nhân viên văn phòng: lương = mức lương – ngày nghỉ x 10000.

Ngoài lương được tính như trên, mỗi nhân viên còn được phụ cấp một khoảng tiền là 100000. Và khoảng tiền này cứ tăng thêm 20000 cho mỗi năm công tác ở công ty.

Viết chương trình: Nhập vào danh sách nhân viên công ty. Tính tổng số tiền công ty phải trả cho nhân viên mỗi tháng.

Hướng dẫn:

- Liệt kê các thuộc tính, phương thức của các lớp

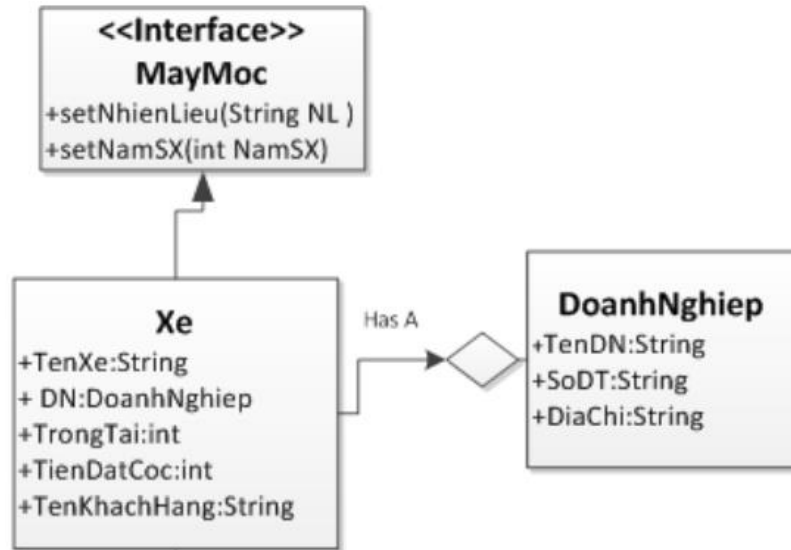
- Vẽ sơ đồ lớp và viết mã nguồn cho lớp trầu tượng cha

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are approximately 20 lines visible. The paper has a slightly textured appearance. In the bottom left corner, there are some small, faint marks that look like dots or artifacts from the scanning process.

Vẽ sơ đồ lớp:

Bài tập 46: Quan hệ đa kế thừa- Interface: Bài toán Máy móc

Cho sơ đồ lớp như sau:



Yêu cầu:

1. Xây dựng giao diện MayHoc như hình vẽ có 2 phương thức có đối số nhiên liệu và năm sản xuất
2. Xây dựng lớp DoanhNghiep thiết đặt các phương thức get/set để đưa thông tin kế hợp vào lớp Xe trong câu 3.
3. Xây dựng lớp Xe
 - a. có các thuộc tính như gợi ý trong đó có thuộc tính kết nối với lớp DoanhNghiep.
 - b. Vì lớp xe cụ thể hóa 2 phương thức trong giao tiếp MayMoc nên ngoài các thuộc tính như hình vẽ ta thêm 2 thuộc tính nhiên liệu, năm sản xuất.
 - c. Khi nhập dữ liệu cho lớp xe lưu ý sử dụng 2 phương thức `setNhiemLieu(...)` và `setNamSX(...)` để gán dữ liệu cho 2 thuộc tính kể trên
4. Xây dựng lớp chứa hàm main nhập vào n xe. In thông tin các xe sau khi nhập và đưa ra xe nào có trọng tải lớn nhất

```
public interface MayMoc {
    public void setNhiemLieu(String NL);
    public void setNamsx(int namSX);
}

public class DoanhNghiep {
    private String tenDN,soDT,diaChi;
    public void nhap()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("nhap ten DN, sdt, dia chi:");
        tenDN=sc.nextLine();
        soDT=sc.nextLine();
        diaChi=sc.nextLine();
    }
    public String toString() {
        return "ten:"+tenDN + "- dt:" + soDT + "-diaChi:" + diaChi;
    }
}

public class Xe implements MayMoc{
    private String tenXe,tenKH,NL;
```

```

private DoanhNghiep dn= new DoanhNghiep();
private float trongTai,tienDatCoc;
private int namSX;
//cụ thể hóa 2 phương thức trong MayMoc
public void setNhiemLieu(String NL){    this.NL=NL;    }
public void setNamsx(int namSX){
    if (namSX>0)
        this.namSX=namSX;
    else
        this.namSX=2016;
}
public void nhap(){
    Scanner s= new Scanner(System.in);
    System.out.print("Nhap ten xe: ");
    tenXe = s.nextLine();
    System.out.print("Nhap trong tai =");    trongTai = s. nextFloat();
    //-----
    //gọi 2 phương thức đã làm trong đa kế thừa
    System.out.print("Nhap nam san xuat xe: ");
    setNamsx(s.nextInt());
    s.nextLine();//bỏ qua enter của dòng
    System.out.print("Nhap nhien lieu của xe: ");
    setNhiemLieu( s.nextLine());
    //-----
    System.out.print("Nhap doanh nghiep: ");    dn.nhap();
    System.out.print("Nhap tien dat coc: "); tienDatCoc= s.nextFloat();
    s.nextLine();    //bỏ qua enter khi nhập số trước nhập xau
    System.out.print("Nhap ten khách hàng thuê xe: ");    tenKH= s.nextLine();
}
public static void inTieuDe(){
    System.out.printf("%10s %10s %10s %25s %10s %10s %10s%n",
        " ten xe", "ten kh"," nhien lieu", " Doanh nghiep", " tai trọng", " tien DC", " Nam sx");
}
public void inDL(){
    System.out.printf("%10s %10s %10s %25s %10.1f %10.1fs %10d %n",
        tenXe,tenKH,NL,dn,trongTai,tienDatCoc,namSX);
}
public float getTrongTai() {
    return trongTai;
}
}

public class DemoXe {
    static ArrayList<Xe> dsXe= new ArrayList<Xe>();
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        int chon;
        do{
            System.out.print("1\tNhap danh sach xe\n");
            System.out.print("2\tin danh sach xe\n");
            System.out.print("3\ttim kiem xe co trong tai max\n");
            System.out.print("4\txoa xe theo stt\n");
            System.out.print("7\tthoat\n");

            chon=sc.nextInt();
            switch(chon)
            {
                case 1:
                    nhapXe(); break;
                case 2:
                    inDSXe(); break;
                case 3:

```

```

        timXeTrongTaiMax();
        break;
    case 4:
        System.out.print("nhap so tt xe can xoa");
        int stt=sc.nextInt();
        xoaXe(stt); break;
    }
    } while(chon!=7);
}
public static void nhapXe()
{
    while(true){
        Xe xe = new Xe();
        System.out.println("dung nhap trong tai =0");
        xe.nhap();
        if (xe.getTrongTai()==0f) break;
        dsXe.add(xe);
    }
}
public static void inDSXe()
{
    Xe.inTieuDe();
    for (int i = 0; i < dsXe.size(); i++)
    {
        dsXe.get(i).inDL();
    }
}
public static boolean xoaXe(int stt)
{
    if ((stt>=0)&&(stt<=dsXe.size()))
    {dsXe.remove(stt);
    return true;
    }
    else
    { System.out.println("vi tri xe ngoai pham vi");
    return false;
    }
}

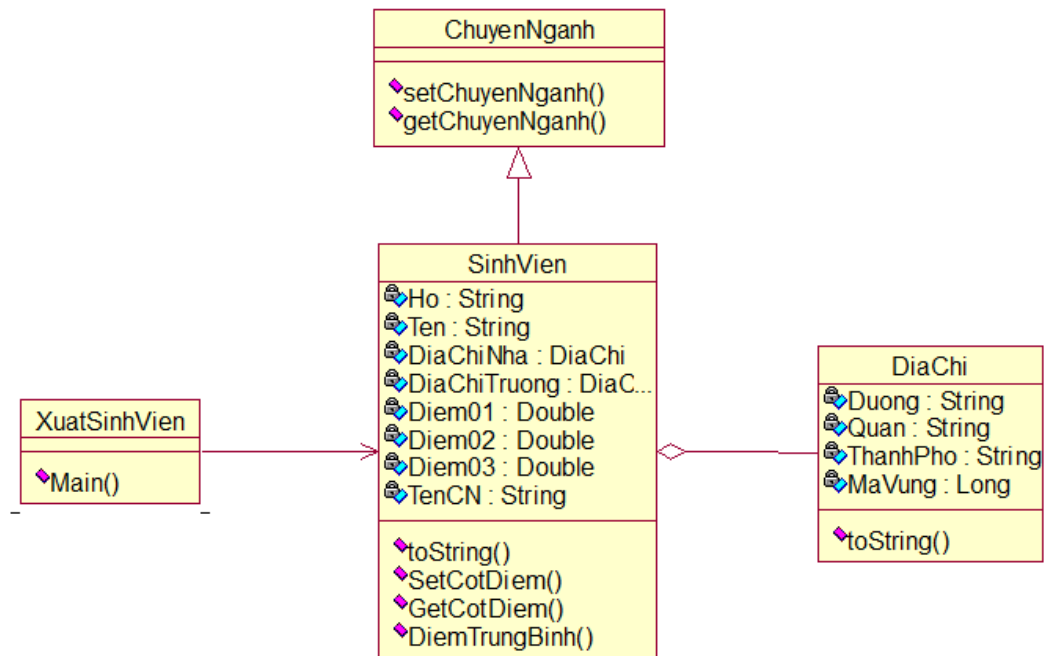
}
public static void timXeTrongTaiMax()
{
    Comparator<Xe> c=new Comparator<Xe>() {

@Override
        public int compare(Xe t, Xe t1) {
return Float.compare(t.getTrongTai(), t1.getTrongTai());
        }
    };
    Xe x=Collections.max(dsXe,c);
    System.out.println("DS xe co trong tai max=");
    Xe.inTieuDe();
    float max=x.getTrongTai();
    for (int i = 0; i < dsXe.size(); i++) {
        if (dsXe.get(i).getTrongTai()==max) {
            dsXe.get(i).inDL();
        }
    }
}
}
}

```

Bài tập 47: Quan hệ đa kế thừa- Interface: Bài toán Sinh viên

Để quản lý và xử lý lương và thông tin nhân viên ta có sơ đồ lớp sau



Yêu cầu:

- Xây dựng giao tiếp ChuyenNganh có 2 phương thức như hình vẽ trên.
- Xây dựng lớp DiaChi: Có các thuộc tính như hình vẽ, phương thức toString() là phương thức ghi đè phương thức mặc định của lớp trả lại một chuỗi là thông tin của địa chỉ.
- Xây dựng lớp SinhVien:
- Có thuộc tính như hình vẽ, trong đó thuộc tính DiaChiNha và DiaChiTruong là các thuộc tính tham chiếu đến lớp DiaChi; phương thức toString() ghi đè phương thức mặc định toString() của lớp trả lại thông tin của SinhVien.
- Xây dựng phương thức SetCotDiem có 2 đối số truyền vào: 1 là loại điểm, 1 là dữ liệu truyền vào cho điểm tương ứng.
- Xây dựng phương thức và GetCotDiem có 1 đối số truyền vào là loại điểm cần lấy giá trị
- Xây dựng phương thức DiemTrungBinh trả lại điểm trung bình của sinh viên
- Xây dựng lớp XuatSinhVien chứa hàm main với yêu cầu
- Nhập vào 1 danh sách n sinh viên. In danh sách sau khi nhập (có in kèm theo điểm trung bình)
- Đưa ra sinh viên trong danh sách có điểm trung bình cao nhất.

Chương 4: Lỗi thực thi và quá trình gom rác

Nội dung kiến thức thực hành sử dụng lớp đã có của java để xử lý lỗi ngoại lệ:

- Xử lý dữ liệu với try-catch
- Ném ngoại lệ
- Tạo lớp ngoại lệ riêng

Bài tập 48: Kiểm soát lỗi thực thi: Bài toán chia cho 0.

Yêu cầu: Viết chương trình cho nhập vào 2 số nguyên, xuất kết quả phép chia 2 số này. Yêu cầu kiểm tra việc nhập số (không được nhập chữ), phép chia cho 0.

Gợi ý:

```
public class Bai1 {
    static int a,b;
    public static void tinh()
    {
        Scanner in=new Scanner(System.in);
        String s;
        try{
            System.out.println("Nhap a: ");      s=in.nextLine();      a=Integer.parseInt(s);
            System.out.println("Nhap b: ");      s=in.nextLine();      b=Integer.parseInt(s);
            System.out.println("thuong: "+a/b);
        }
        catch (NumberFormatException n){
            System.out.println("Lỗi: không phải số!");
        }
        catch (ArithmeticException e)
        {
            System.out.println("Lỗi: phép chia cho 0!");
        }
    }
    public static void main(String[] args) {
        tinh();
    }
}
```

Bài tập 49: Kiểm soát lỗi thực thi: bài toán ràng buộc dữ liệu.

Yêu cầu:- Làm lại bài 1 với yêu cầu kiểm tra việc nhập 2 số phải là số dương, viết lớp xử lý riêng.

```
public class DuLieu {
    public int a,b;
    public void nhap() {
        Scanner in=new Scanner(System.in);
        String s;
        try{
            System.out.println("Nhap vao so a: ");s=in.nextLine();a=Integer.parseInt(s);
            System.out.println("Nhap vao so b: "); s=in.nextLine(); b=Integer.parseInt(s);
        }
        catch (NumberFormatException nb) {
            System.out.println("Dữ liệu phải là số!");
        }
    }
    public void tinh() throws Exception {
        if (b==0) throw new ChiaCho0Exception();
        System.out.println("thuong 2 so: "+a/b);
    }
}

public class ChiaCho0Exception extends Exception {
    public ChiaCho0Exception() {
        System.out.println("Lỗi: phép chia cho 0!");
    }
}

public class Main {
```

```

    public static void main(String[] args){
        DuLieu dl=new DuLieu();
        dl.nhap();
        try{
            dl.tinh();
        }
        catch (ChiaCho0Exception c) { c.getMessage(); }
        catch (Exception ex) { ex.getMessage(); }
    }
}

```

Bài tập 50: Kiểm soát lỗi thực thi : Bài toán lớp xử lý ngoại lệ 1.

Yêu cầu: Viết lớp xử lý ngoại lệ StringTooLongException, lớp này giúp thông báo 1 chuỗi nào đó có quá nhiều ký tự.

- Viết hàm main, cho người dùng nhập vào từng chuỗi cho đến khi người dùng nhập “DONE”. Khi một chuỗi được nhập vào, cần kiểm tra xem chuỗi đó có vượt quá 20 ký tự không, nếu vượt thì chương trình thông báo cho người dùng biết và kết thúc chương trình.
- Yêu cầu sử dụng lớp StringTooLongException để xử lý lỗi này.

Bài tập 51: Kiểm soát lỗi thực thi: Bài toán viết lớp xử lý ngoại lệ 2.

Yêu cầu: - Làm lại bài 3 với yêu cầu nếu người dùng nhập vào một chuỗi vượt quá 20 ký tự thì chương trình thông báo lỗi và vẫn tiếp tục thực hiện.

Bài tập 52: Kiểm soát lỗi thực thi: Tự viết hàm xử lý ngoại lệ

Hãy tạo một class tên là OutOfRangeException để kiểm tra việc nhập dữ liệu của người sử dụng. Sinh viên hãy tạo thêm một class để sử dụng OutOfRangeException. Ví dụ như yêu cầu nhập vào 1 số n có giá trị từ -113 tới 113 từ bàn phím, nếu không nằm trong đoạn giá trị này thì dùng OutOfRangeException để thông báo lỗi.

Bài tập 53: Xử lý ngoại lệ: Bài toán sinh viên kiểm tra hợp lệ dữ liệu.

Viết chương trình nhập vào thông tin sinh viên: tên, tuổi. Yêu cầu bắt ngoại lệ tên không được để trống và tuổi phải trong khoảng 0 đến 10.

Gợi ý thực hiện: Chú ý bắt lỗi lan truyền. nếu không bắt tại dòng lệnh gây lỗi thì đẩy lỗi ra bên ngoài cho hàm gọi xử lý.

```

public class SinhVien {
    String ten;
    int tuoi;
    public void setTen(String t) throws Exception {
        if (t.equals("")) {
            throw new Exception("Ten khong duoc de trong...");
        } else {
            ten = t;
        }
    }
    void setTuoi(int t) throws Exception {
        if (t <= 0) {
            throw new Exception("Tuoi phai > 0 ");
        } else {
            tuoi = t;
        }
    }
    void nhap(){ //hàm nhập sẽ bắt lỗi lan truyền bằng try-catch
        Scanner s = new Scanner(System.in);
        // while(true)
        do
        {
            try {
                System.out.print("Nhap Ten: ");
                setTen(s.nextLine());
                System.out.print("Nhap Tuoi : ");
            }

```

```

        setTuoi(s.nextInt());
        // break;
    }
    catch (java.util.InputMismatchException ex) {
        System.out.println("Du lieu k hop le. nhap lai");
    }
    catch (Exception ex) {
        System.out.println("Loi Nhap du lieu...");
        System.out.println(ex.toString());
    }
}
while(tuoi<=0||ten==null);
}
public String toString() {
    return "SinhVien{" + "ten=" + ten + ", tuoi=" + tuoi + '}';
}
public static void main(String[] args) {
    SinhVien v=new SinhVien();
    v.nhap();
    System.out.println(v);
}
}

```

Chương 5: Các lớp cơ sở và cấu trúc dữ liệu.

Nội dung kiến thức thực hành thao tác lớp xử lý tập hợp các đối tượng.

- Thao tác trên mảng dữ liệu cơ sở
- Thao tác trên mảng đối tượng
- Viết lớp tập hợp sử dụng mảng và ArrayList
- Sắp xếp mảng đối tượng trong tập hợp với Comparable interface

Bài tập 54: Các thao tác trên tập hợp: Bài toán nhập xuất, sắp xếp tập hợp nguyên thủy

Mục đích:

Yêu cầu: Sử dụng tập hợp ArrayList lưu trữ dữ liệu. Viết chương trình thực hiện các yêu cầu sau:

- Nhập / xuất danh sách số nguyên bất kỳ
- Tìm kiếm giá trị bất kỳ trong tập hợp nhập vào từ bàn phím
- Xuất tất cả các số nguyên tố trong tập hợp vừa nhập
- Sắp xếp tập hợp sử dụng phương thức sắp xếp của Collections.
- Xóa một giá trị bất kỳ trong tập hợp
- Kiểm tra mảng có đối xứng hay tập hợp

Bài tập 55: Các thao tác trên tập hợp: Bài toán tìm, thêm, xóa phần tử lớp Circle

Yêu cầu:

(a) Viết lớp Circle, biết lớp có một thuộc tính là radius (bán kính). Viết hàm getArea() tính diện tích hình tròn, và hàm toString() trả về chuỗi gồm bán kính và diện tích hình tròn. Kiểm tra lớp Circle.

(b) Viết lớp CircleCollection, trong đó viết các hàm:

- addCircle: thêm 1 hình tròn vào mảng;
- getSize: lấy số lượng hình tròn trong mảng;
- getCircle/setCircle: lấy/gán hình tròn tại vị trí xác định trong mảng;
- toString: trả về thông tin của tập các hình tròn trong mảng;
- Hàm tính tổng diện tích hình tròn;
- Hàm tìm diện tích lớn nhất;
- Hàm lấy hình tròn có diện tích nhỏ nhất.

(c) Viết chương trình tạo một mảng N hình tròn, với bán kính được phát sinh ngẫu nhiên. Thực hiện các yêu cầu sau:

- Xuất thông tin của các hình tròn đã nhập.
- Xuất tổng diện tích của chúng.
- Xuất diện tích lớn nhất.
- Xuất thông tin hình tròn có diện tích nhỏ nhất.

Bài tập 56: Các thao tác trên tập hợp: Bài toán quản lý của Công ty TrueLove

Công ty TrueLove cần lưu tên của các nhân viên của mình. Mỗi tháng một nhân viên sẽ được chọn ngẫu nhiên để nhận một quà tặng. Hãy dùng tuyển tập để viết chương trình quản lý danh sách nhân viên.

Công ty TrueLove cần đặt tên cho sản phẩm mới, tên sản phẩm được chọn từ tên của nhân viên, vì vậy tên không được trùng, tên chỉ được dùng có 1 lần. Hãy dùng tuyển tập để viết chương trình cung cấp tên cho sản phẩm.

Công ty TrueLove muốn dùng tên phổ biến nhất cho sản phẩm của họ, tên phổ biến là tên giống nhau nhiều nhất. Hãy dùng tuyển tập để viết chương trình cung cấp tên cho sản phẩm.

Công ty TrueLove muốn cho nhân viên đi du lịch, chính sách được tạo ra là ưu tiên cho những người đăng ký trước. Hãy dùng tuyển tập để viết chương trình đăng ký du lịch.

Công ty TrueLove muốn tạo danh sách các khách hàng theo thứ tự tăng dần theo doanh số. Hãy dùng tuyển tập để viết chương trình quản lý danh sách khách hàng.

Vẽ sơ đồ biểu diễn mối quan hệ giữa các lớp tìm được:

Bài tập 57: Các thao tác trên tập hợp: Bài toán quản lý phòng học

Phòng học được quản lý trong một trường đại học gồm: phòng học lý thuyết, phòng máy tính và phòng thí nghiệm .

- Mỗi **phòng học** đều có mã phòng, dãy nhà, diện tích, số bóng đèn.
- **Phòng học lý thuyết** thì cần quan tâm xem có máy chiếu không.
- **Phòng máy tính** thì cần biết là trang bị bao nhiêu máy tính.
- **Phòng thí nghiệm** thì thêm thông tin chuyên ngành, sức chứa, có bồn rửa không (*rửa dụng cụ thí nghiệm / rửa tay*).
- Ngoài ra, người quản lý cần phải xem xét phòng học có đạt chuẩn không.
- Thực hiện cài đặt tường minh cho mỗi loại phòng cụ thể trên. Phòng học đạt chuẩn nếu:
 - Tất cả các phòng đều phải đủ ánh sáng (trung bình 10m² - 1 bóng đèn), và
 - Phòng lý thuyết, nếu có máy chiếu.
 - Phòng máy tính, nếu trung bình 1.5m² đặt một máy.
 - Phòng thí nghiệm, nếu có bồn rửa đi kèm.
- Hãy Xây dựng lớp quản lý danh sách phòng học.
- Dùng một List (ArrayList, LinkedList, Vector) để lưu trữ danh sách phòng học.
- Tạo constructor khởi tạo danh sách.
- Viết phương thức thêm một phòng học vào danh sách (*thêm thành công nếu không bị trùng mã phòng*).
- Viết phương thức tìm kiếm một phòng học nào đó khi biết mã phòng.
- Viết phương thức in toàn bộ danh sách các phòng học.
- Viết các phương thức để in danh sách các phòng học đạt chuẩn.
- Viết phương thức để sắp xếp danh sách tăng dần theo cột dãy nhà.
- Viết phương thức để sắp xếp danh sách giảm dần theo cột diện tích
- Viết phương thức để sắp xếp danh sách tăng dần theo cột số bóng đèn.
- Viết phương thức để cập nhật số máy tính cho một phòng máy tính nào đó khi biết mã phòng.
- Viết phương thức để xóa một phòng học nào đó khi biết mã phòng. Lưu ý khi test chương trình, khi xóa cần phải xác minh rằng có chắc chắn xóa không?
- Viết phương thức để in ra tổng số phòng học.
- Viết phương thức để in ra tổng số phòng học
- Viết các phương thức để in danh sách các phòng máy có 60 máy
- Tạo lớp cho phần thử nghiệm, với menu lựa chọn để thực hiện các chức năng theo yêu cầu.

Vẽ sơ đồ biểu diễn mối quan hệ giữa các lớp tìm được:

Bài tập 58: Các thao tác trên tập hợp: Quản lý khách hàng xếp hàng mua vé tại nhà ga.

Thông tin lưu trữ cho khách hàng gồm: số CMND khách hàng (String), Tên khách hàng, Ga đến, giá tiền (double).

Hệ thống menu gồm các mục:

- Thêm một khách hàng mới vào hàng đợi mua vé.
- Bán một vé cho khách hàng. Chỉ bán cho người đăng ký trước.
- Hiện thị danh sách khách hàng.
- Hủy một khách hàng ra khỏi danh sách. (khách hàng không mua vé nữa).
- Thống kê tình hình bán vé
- Lưu danh sách vào file
- Hiện thị danh sách các ga đang chờ mua vé.
- Hiện thị danh sách các ga đang chờ mua vé và số vé tương ứng cho ga.

Lưu ý:

- Số khách hàng trong danh sách hiện tại là số khách đang chờ, nhưng chưa có vé. Khi một khách hàng đã mua vé, thì loại khách hàng này ra khỏi danh sách chờ mua vé.
- Việc mua vé phải có thứ tự: ai vào trước thì mua vé trước (FIFO).
- Mỗi khi khách hàng mua được vé phải lưu lại khách hàng này để dùng cho việc thống kê.
- Mỗi khi thêm một khách hàng mới, nếu Số CMND khách hàng đã có thì không tạo phân tử mới mà chỉ cập nhật lại ga và giá tiền đến cho khách hàng đó.
- Mục thống kê tình hình: cho biết còn bao nhiêu khách hàng chờ nhận vé, bao nhiêu khách hàng đã nhận vé, tổng số tiền đã thu về là bao nhiêu
- Việc lưu danh sách: chỉ lưu các khách hàng chờ mua vé. Các khách hàng đã nhận vé xem như kết sổ trong ngày không cần lưu lại.
- Khi chương trình vừa được chạy, lập tức tự động nạp toàn bộ danh sách khách hàng từ file (cách khách hàng chưa có vé).
- Khi hiện thị danh sách các ga đến đang chờ mua vé, chỉ hiện thị tên ga đó một lần. (Ví dụ: giả sử 10 khách hàng nhưng đăng ký đi đến 2 ga, thì chỉ hiện thị 2 hàng).

Vẽ sơ đồ biểu diễn mối quan hệ giữa các lớp tìm được:

Bài tập 59: Các thao tác trên tập hợp: Quản lý đĩa CD

Yêu cầu: sử dụng ArrayList thay cho mảng thông thường.

CD House là công ty chuyên về cho thuê và bán băng đĩa audio, video với hơn 700 đầu đĩa khác nhau. Hiện nay công ty đang muốn xây dựng hệ thống quản lý để tự động hóa các hoạt động của công ty. Giả sử bạn là nhân viên phòng phát triển phần mềm được yêu cầu để xây dựng hệ thống này.

Xây dựng một ứng dụng bằng Java để quản lý danh mục CD với những chức năng cơ bản như: Bổ sung CDs mới, tìm kiếm CDs, Hiển thị danh sách chi tiết CDs

Chương trình bắt đầu với menu như sau:

- * 1. Add(Thêm CD mới)
- * 2. Search(Tìm kiếm CD theo tên)
- * 3. Display(Hiển thị danh mục CD)
- * 4. Exit(Kết thúc)

Xây dựng menu để hiển thị như ở trên. Khi người dùng chọn 1 mục trên menu, chức năng tương ứng được gọi. Khi người dùng chọn mục Exit(Kết thúc), chương trình kết thúc và in ra thông báo.

Khi chương trình chạy và người dùng chọn mục đầu tiên (Thêm CD mới), chương trình kiểm tra xem danh sách còn trống không. Nếu không có chỗ trống, chương trình in ra câu thông báo "Không thể thêm CD mới". Ngược lại cho phép nhập Thông tin chi tiết của CDs bao gồm:

Thể loại của CD(game/movie/music), Loại CD(Audio/Video), Tiêu đề, Giá, Mã số CD và năm xuất bản. Sau khi nhập xong Thông tin chi tiết CD được lưu trong mảng chứa Cds và biến cdCounter đếm số CDs được tăng lên.

Nếu chọn mục 2 để tìm CD, chương trình sẽ kiểm tra xem có CDs nào trong danh sách không? Nếu có thì việc tìm kiếm sẽ được thực hiện. Chương trình sẽ hiển thị chi tiết của CD nếu tìm thấy còn không sẽ hiển thị thông báo không tìm thấy.

Nếu chọn mục 3 chương trình sẽ hiển thị thông tin chi các CDs trong danh sách

Khi chọn mục 4 sẽ kết thúc chương trình.

Vẽ sơ đồ biểu diễn mối quan hệ giữa các lớp tìm được:

Bài tập 60: Các thao tác trên tập hợp: Quản lý mua vé tàu

Quản lý khách hàng xếp hàng mua vé tại nhà ga. Thông tin lưu trữ cho khách hàng gồm: số CMND khác hàng (String), Tên khách hàng, Ga đến, giá tiền (double).

Hệ thống menu gồm các mục:

- Thêm một khách hàng mới vào hàng đợi mua vé.
- Bán một vé cho khách hàng. Chỉ bán cho người đăng ký trước.
- Hiện thị danh sách khách hàng.
- Hủy một khách hàng ra khỏi danh sách. (khách hàng không mua vé nữa).
- Thống kê tình hình bán vé
- Lưu danh sách vào file
- Hiện thị danh sách các ga đang chờ mua vé.
- Hiện thị danh sách các ga đang chờ mua vé và số vé tương ứng cho ga.

Lưu ý:

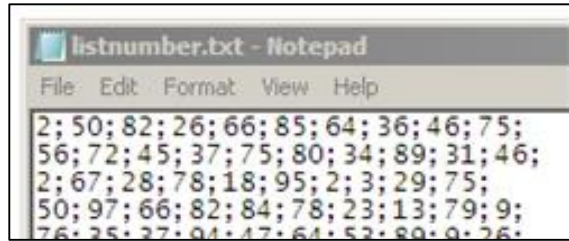
- Số khách hàng trong danh sách hiện tại là số khách đang chờ, nhưng chưa có vé. Khi một khách hàng đã mua vé, thì loại khách hàng này ra khỏi danh sách chờ mua vé
- Việc mua vé phải có thứ tự: ai vào trước thì mua vé trước (FIFO).
- Mỗi khi khách hàng mua được vé phải lưu lại khách hàng này để dùng cho việc thống kê.
- Mỗi khi thêm một khách hàng mới, nếu Số CMND khách hàng đã có thì không tạo phần tử mới mà chỉ cập nhật lại ga và giá tiền đến cho khách hàng đó.
- Mục thống kê tình hình: cho biết còn bao nhiêu khách hàng chờ nhận vé, bao nhiêu khách hàng đã nhận vé, tổng số tiền đã thu về là bao nhiêu.
- Việc lưu danh sách: chỉ lưu các khách hàng chờ mua vé. Các khách hàng đã nhận vé xem như kết sổ trong ngày không cần lưu lại.
- Khi chương trình vừa được chạy, lập tức tự động nạp toàn bộ danh sách khách hàng từ file (cách khách hàng chưa có vé).
- Khi hiện thị danh sách các ga đến đang chờ mua vé, chỉ hiện thị tên ga đó một lần. (Ví dụ: giả sử 10 khách hàng nhưng đăng ký đi đến 2 ga, thì chỉ hiện thị 2 hàng).

Vẽ sơ đồ biểu diễn mối quan hệ giữa các lớp tìm được:

Chương 6: Các luồng vào ra và thao tác với tệp dữ liệu

Bài tập 61: Thao tác với file text

- a) Viết hàm cho phép lưu tập tin dưới dạng text file, yêu cầu khởi tạo là 10 dòng, mỗi dòng sẽ có 10 số ngẫu nhiên cách nhau bởi dấu “;”. Xem hình minh họa



- b) Viết hàm cho phép đọc tập tin từ câu a, xuất ra tổng giá trị của các phần tử trên mỗi dòng.
Ghi chú: Trường hợp này có thể phát sinh lỗi IOException.

Hướng dẫn:

- Sinh viên xem cách lưu tập tin dưới dạng TextFile và cách đọc dữ liệu lên. Từ ví dụ này hãy kết hợp với StringTokenizer hoặc hàm split để tiến hành tách số trong quá trình đọc từng dòng dữ liệu, cộng dồn các giá trị này là chúng ta sẽ có tổng giá trị các phần tử ngẫu nhiên trên mỗi dòng.

Hàm lưu TextFile:

```
public static void lưuTextFile(String filename) {
    try {
        FileOutputStream fOut=new FileOutputStream(filename);
        PrintWriter print=new PrintWriter(fOut, true);
        Random rd=new Random();
        for(int i=0;i<10;i++) {
            String line="";
            for(int j=0;j<10;j++) {
                line+=rd.nextInt(100)+" ";
            }
            print.println(line);
        }
        print.close();
        fOut.close();
    }
    catch (FileNotFoundException e) { e.printStackTrace(); }
    catch (IOException e) { e.printStackTrace(); }
}
```

Hàm đọc TextFile:

```
public static void docTextFile(String filename) {
    try {
        FileInputStream fln=new FileInputStream(filename);
        Scanner sc=new Scanner(fln);
        while(sc.hasNextLine())
        {
            System.out.println(sc.nextLine());
        }
        fln.close();
        sc.close();
    }
    catch(IOException ex) { ex.printStackTrace(); }
}
```

Bài tập 62: Tuần tự hóa đọc ghi danh sách đối tượng vào file (Object Serializable)

Yêu cầu:

- Tất cả các lớp phải cài đặt giao diện Serializable:

```
public class Sanpham implements Serializable{...}
public class DanhMucSanPham implements Serializable{...}
```

- Lớp chứa phương thức lưu và đọc đối tượng:

- o Lưu đối tượng:

```
public static void luuDoiTuong(Object obj, String fileName) {
    try
    {
        FileOutputStream fOut=new FileOutputStream(fileName);
        ObjectOutputStream out=new ObjectOutputStream(fOut);
        out.writeObject(obj);
        out.close();
    }
    catch(Exception ex) {
        ex.printStackTrace();
    }
}
```

- Đọc đối tượng

```
public static Object docDoiTuong(String fileName)
{
    try
    {
        FileInputStream fln=new FileInputStream(fileName);
        ObjectInputStream in=new ObjectInputStream(fln);
        Object obj=in.readObject();
        in.close();
        return obj;
    }
    catch(Exception ex) { ex.printStackTrace();
    }
    return null;
}
```

- Trong testMain:

```
DanhMucSanPham dsDienTu=(DanhMucSanPham )MyFile.docDoiTuong("data.txt");
if(dsDienTu!=null) {
    // xử lý danh sách điện tử.....
}
//lưu lại danh sách
MyFile.luuDoiTuong(dsDienTu, "data.txt");
```

Bài tập 63: Bài tập tổng hợp: Quản lý phòng thi

Cho lớp **Thí sinh** gồm các thông tin: Số báo danh, họ tên, điểm toán, điểm lý, điểm hóa, điểm toán.

Lớp Phòng Thi: mã phòng thi, tên phòng thi, địa chỉ phòng thi và danh sách các thí sinh trong phòng

Lớp Quản lý phòng thi thực hiện thêm, sửa, xóa, tìm kiếm thí sinh trong phòng thi.

Vẽ sơ đồ biểu diễn mối quan hệ giữa các lớp tìm được:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Hướng dẫn:

//tạo lớp thí sinh

```
public class ThiSinh implements Serializable{
```

```
    private String soBD,hoTen;
```

```
    private float diemToan,diemLy, diemHoa;
```

```
    //xây dựng các hàm get các thuộc tính. Sinh viên tự làm
```

```
    //hàm set giá trị cho các thuộc tính không bắt ngoại lệ ngay mà đẩy ra ngoài để nơi nào sử dụng sẽ xử lý tiếp
```

```
    public void setSoBD(String soBD) throws Exception{
```

```
        if(soBD.trim().equals("")) throw new Exception("Số báo danh không được trống!");
```

```
        this.soBD = soBD;
```

```
    }
```

```
    public void setHoTen(String hoTen)throws Exception {
```

```
        .....
        .....
        .....
    }
```

```
    public void setDiemToan(float diemToan) throws Exception{
```

```
        .....
        .....
        .....
    }
```

```

//thực hiện tương tự cho các thuộc tính còn lại nếu có bất lỗi. Sinh viên tự làm
.....
.....
.....
.....
.....
.....
.....
//Xây dựng một số hàm tạo có 1 đối để tìm kiếm theo đối tạo , .....
.....
.....
.....
.....
.....
//Xây dựng 1 constructor không đối dự phòng
.....
.....
.....
.....
.....
//Xây dựng 1 hàm tạo có đầy đủ các đối để dùng trong một số tình huống minh họa đối tượng
.....
.....
.....
.....
.....
//Ghi đè các hàm hashCode &equals chọn soBD để chọn khóa và chỉ ra tiêu chí so sánh đ.tượng
}
public class PhongThi implements Serializable {
    private String msPT;
    private String diachiPT;
    private int luongTS;

    //đối tượng dùng chứa tập các thí sinh
    private ArrayList<ThiSinh> dsTS;

    //insert code constructor và sửa lại:
    public PhongThi() {
        .....
        .....
        .....
        .....
        .....
    }
    public PhongThi(String msPT, String diachiPT, int luongTS) {
        .....
        .....
        .....
        .....
        .....
    }
}

```

```

//insert code hashCode and equal chọn mã phòng thi

// xây dựng hàm thêm thí sinh vào phòng thi
public boolean themThiSinh(ThiSinh ts) {
    //Nếu thí sinh đã tồn tại thì không cho thêm
    .....
    .....
    .....
}
//hàm xóa thí sinh khỏi phòng thi
public boolean XoaThisinh(String soBD)throws Exception {
    .....
    .....
    .....
}
//hàm sửa thông tin thí sinh trong phòng
public boolean SuaThongtinTS(String soBD, ThiSinh newTS)throws Exception {
    .....
    .....
    .....
}
//hàm lấy thông tin thí sinh khi biết số báo danh
public ThiSinh LayThongtinTS(String soBD)throws Exception {
    .....
    .....
    .....
}
//lấy thông tin thí sinh khi biết vị trí
public ThiSinh LayThongtinTS(int index) {
    .....
    .....
    .....
}
//Hàm trả lại số thí sinh thật sự trong danh sách
public int soThisinh()
{
    .....
}
//xây dựng lớp xử lý thông tin lưu và đọc file
public class DBEngine {
    public void LuuFile(String fileName, Object obj) throws Exception{
        //Tạo luồng ghi file
        FileOutputStream fs=new FileOutputStream(fileName);
        //Tạo luồng để serial đối tượng
        ObjectOutputStream os=new ObjectOutputStream(fs);
        //chuyển tải đối tượng tới đích (tập tin)
        os.writeObject(obj);
    }
}

```



```

        //đóng luồng
        fs.close(); os.close();
    }
    // tạo phương thức đọc file
    public Object docFile(String fileName) throws Exception{
        Object kp=null;
        //Tạo luồng đọc file đã được serial
        FileInputStream fi=new FileInputStream(f);
        //Tạo luồng để Deserialize đối tượng
        ObjectInputStream ois=new ObjectInputStream(fi);
        //Tiến hành khôi phục đối tượng
        kp=ois.readObject();
        //đóng luồng
        fi.close();ois.close();
        return kp;
    }
}
//xây dựng lớp quản lý phòng thi chứa hàm main thực hiện
    - Khai báo phòng thi
    - Hiện thị menu cho chọn các công việc, thêm, sửa, xóa thí sinh vào phòng thi
    public class QuanLyPhongThi {

        private static PhongThi phongthi=null;
        static String fileName="src/MyFile/data.txt";
        static DBEngine db=new DBEngine();
        public static void main(String[] args) {
            System.out.println("*****QUẢN LÝ PHÒNG THI*****");
            //=====
            phongthi=new PhongThi("pt01SE","Phòng 502, A9",20);
            //=====
            Scanner sc=new Scanner(System.in);
            do {
                System.out.println("=====");
                System.out.println("1. Thêm Thí sinh mới");
                System.out.println("2. Hiệu chỉnh thông tin thí sinh");
                System.out.println("3. Xóa thí sinh khỏi phòng thi");
                System.out.println("4. Lấy thông tin Thí sinh khi biết số báo danh.");
                System.out.println("5. Lấy thông tin Thí sinh khi biết số thứ tự.");
                System.out.println("6. In danh sách thí sinh");
                System.out.println("7. Lưu phòng thi vào file");
                System.out.println("8. Đọc phòng thi từ file");
                System.out.println("9. Thoát");
                System.out.println("*****");
                System.out.print("\t**Chọn lựa của bạn? <1->9>:");
                int tl=sc.nextInt();
                switch(tl)
                {
                    case 1: ThemTS();break;
                    case 2: SuaTTTS();break;
                    case 3: XoaTS();break;

```

[illegible]


```
static void layTTTS_soBD() {
```

```
    try {
```

```
        .....
        .....
        .....
        .....
        .....
```

```
        ThiSinh ts=phongthi.LayThongtinTS(soBD);
```

```
        System.out.println("~~~~~Kết quả~~~~~");
```

```
        if(ts==null)
```

```
            System.out.println("Không có");
```

```
        else
```

```
            System.out.println(ts);
```

```
        System.out.println("~~~~~");
```

```
    } catch (Exception ex) {
```

```
        System.out.println(ex.getMessage());
```

```
    }
```

```
}
```

```
static void InDanhSachTS() {
```

```
    System.out.println("=====DANH SÁCH THÍ SINH=====");
```

```
    System.out.println("Số BD \tHọ tên \tĐiểm toán \tĐiểm Lý \tĐiểm Hóa");
```

```
    for (int i = 0; i < phongthi.soThisinh(); i++) {
```

```
        try {
```

```
            .....
            .....
            .....
            .....
            .....
```

```
        } catch (Exception ex) {
```

```
            System.out.println(ex.getMessage());
```

```
        }
```

```
    }
```

```
    System.out.println("=====");
```

```
}
```

```
}
```