**BÀI TẬP THỰC HÀNH LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

# Lập trình hướng đối tượng và các khái niệm cơ bản

+ Sử dụng JDK để biên dịch và thực thi chương trình.

+ Sử dụng Eclipse hoặc netBean để soạn thảo, biên dịch và thực thi chương trình.

+ Khai báo và sử dụng biến, đối tượng.

+ Sử dụng các cấu trúc điều khiển.

## Làm quen công cụ

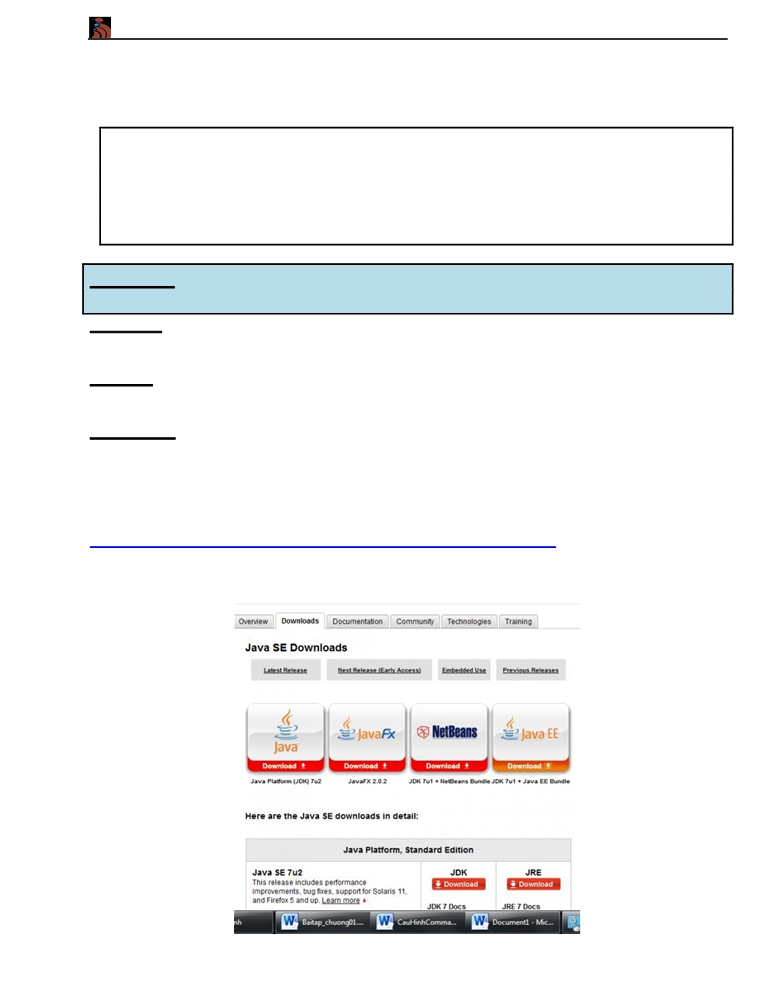
* Mục đích: Download và cài đặt được JDK.
* Yêu cầu: Ttải và cài đặt JDK

***Hướng dẫn:***

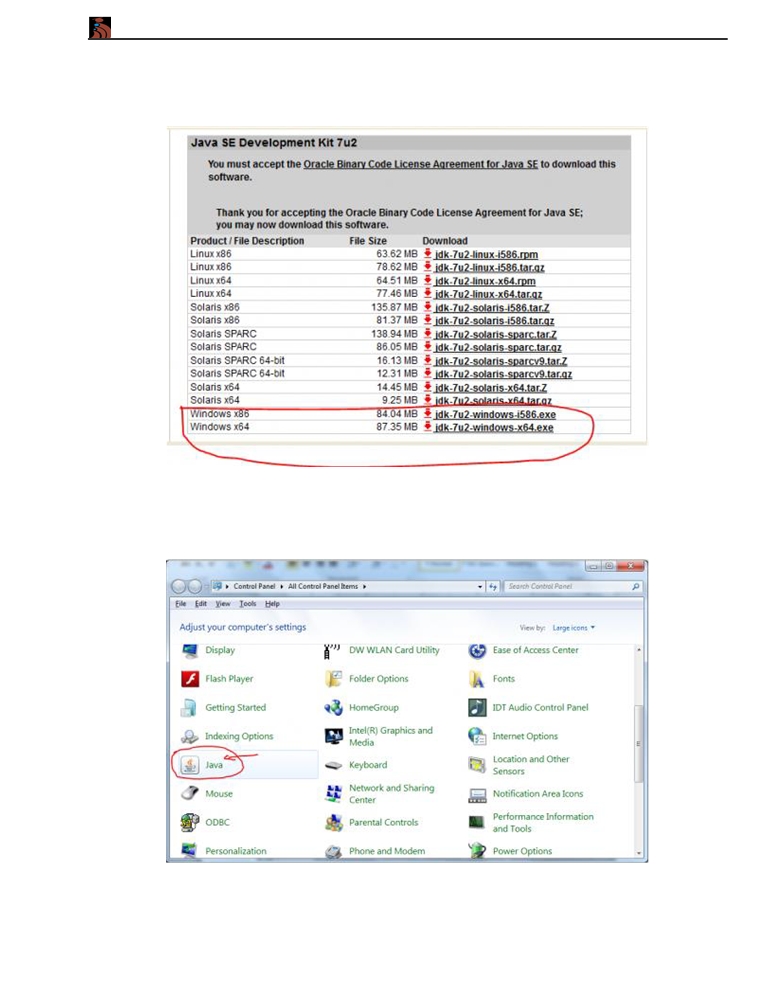
* Để biên dịch được các source code Java, máy tính của chúng ta phải có máy Ảo Java Để có được máy Ảo Java các bạn vào link bên dưới để download bộ JDK

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

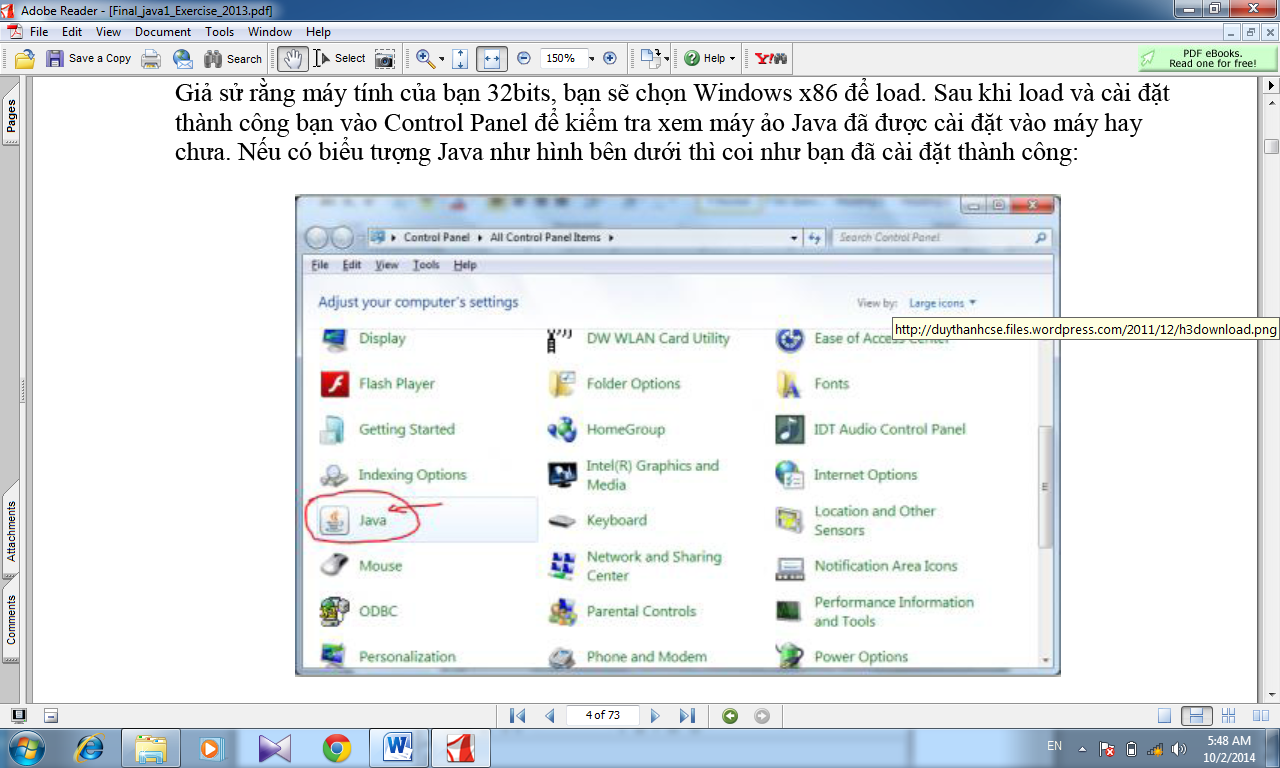
* Sau khi click vào đường link bên trên, danh mục JAVA Kit sẽ hiển thị như hình bên.



chọn JDK để Download, sau khi click chọn Download thì một màn hình mới sẽ được hiển thị ra, trong màn hình mới này tùy vào cấu hình của máy tính mà chúng ta chọn tập tin Download cho hợp lý.



Giả sử rằng máy tính của bạn 32bits, bạn sẽ chọn Windows x86 để load. Sau khi load và cài đặt thành công bạn vào Control Panel để kiểm tra xem máy ảo Java đã được cài đặt vào máy hay chưa. Nếu có biểu tượng java như bên dưới khi đó bạn đã cài đặt java thành công



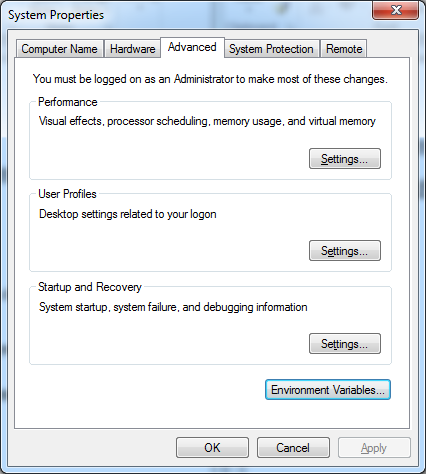
## Thực hiện java ở chế độ command line

Yêu cầu: Thiết lập được biến môi trường để chạy java ở chế độ command line.

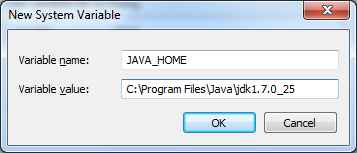
Hướng dẫn:

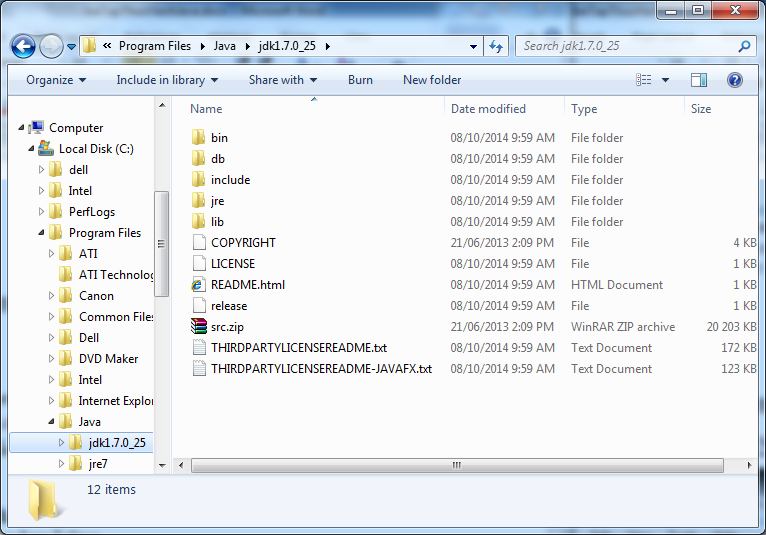
Bước 1: Bấm chuột phải vào biểu tượng Computer/ chọn Properties

Bước 2: chọn Advanced System Settings . chọn Tab Advanced.

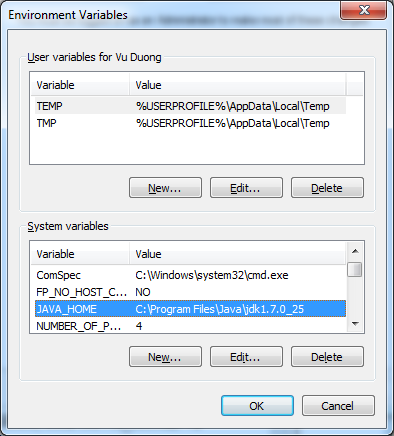


Bước 3: Trong mục System variables, click chọn button New



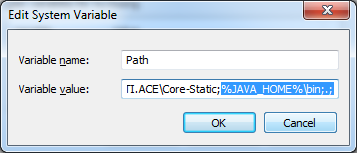
* Mục Variable name, chúng ta nhập vào biến JAVA\_HOME
* Mục Variable value nhận giá trị là đường dẫn cài đặt JDK, trong ví dụ này thì JDK được cài đặt như hình minh họa. Tức là nếu máy của bạn cài JDK ở đâu thì copy paste đường dẫn đó vào mục này.
* 

Sau khi chọn OK, bạn quan sát trong vùng System Variables, biến JAVA\_HOME và giá trị của biến xuất hiện như hình minh họa bên dưới:



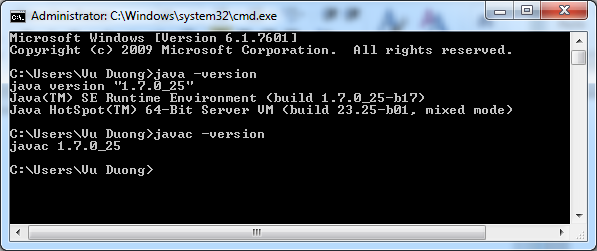
tiếp theo, tìm tới biến Path trong mục System variables, click chọn Edit:

Trong mục Variable value, các bạn di chuyển tới cuối, nhập vào các giá trị như hình minh họa : ;%JAVA\_HOME%\bin;.;



6. Kiểm tra lại cấu hình có chính xác hay không:

Để kiểm tra xem máy tính của bạn đang cài JDK version bao nhiêu, vào Start/ Run. Hoặc gõ tổ hợp phím Windows+R để hiển thị cửa sổ Run, trong cửa sổ này các bạn gõ vào lệnh cmd rồi nhấn phím Enter



Nếu như cửa sổ chạy ra có dạng như hình là OK. Nếu ra thông báo: 'javac' is not recognized as an internal or external command, operable program or batch file, thì bạn thiết lập sai.. Thực hiện kiểm tra theo các bước đã thực hiện.

## Soạn thảo và viết chương trình java bằng notepad

Mục đích: Viết được chương trình Java bằng notepad và biên dịch bằng command - line

Yêu cầu: Soạn thảo đoạn code sau sử dụng Notepad:

class Welcome

{

public static void main(String []arg)

{

System.out.println("\nMy first Java Program");

}

}

2. Chọn menu File\ Save As và gõ tên file ‘Welcome.java’ (giống tên của class)

3. Chọn Start \ Run và đánh lệnh ‘cmd’ vào trong hộp thoại Run

4. Click vào button OK để cửa sổ command prompt xuất hiện. Thay đổi đường dẫn thư mục lưu file trong bước 3.

5. Biên dịch chương trình sử dụng lệnh ‘javac’

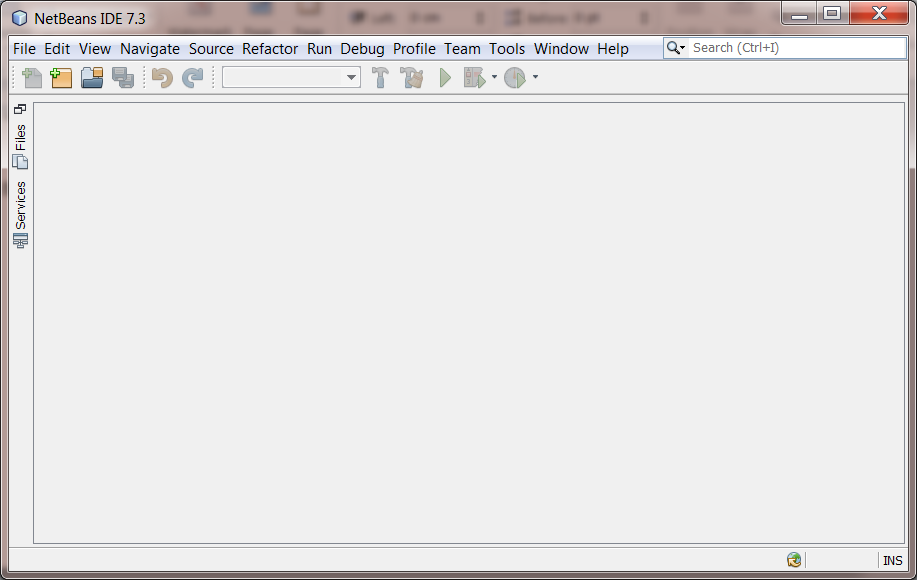
6. Chạy chương trình sử dụng ‘java’

## Ví dụ đầu tiên.

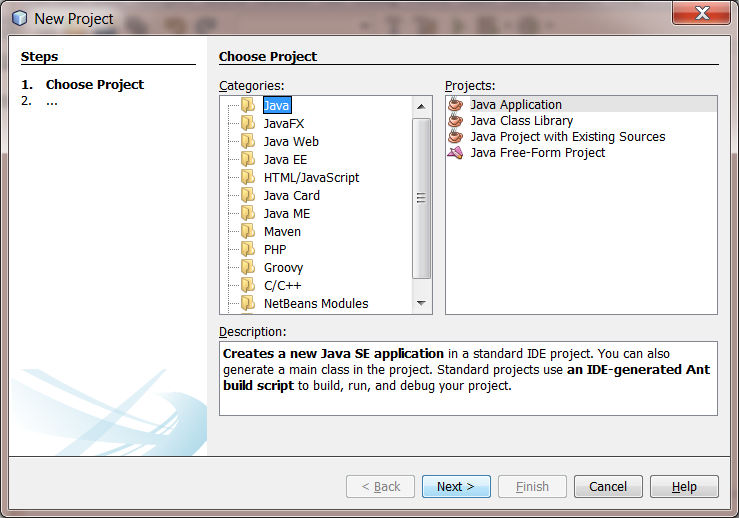
**Mục đích**: Sử dụng được netBeans

**Yêu cầu**:

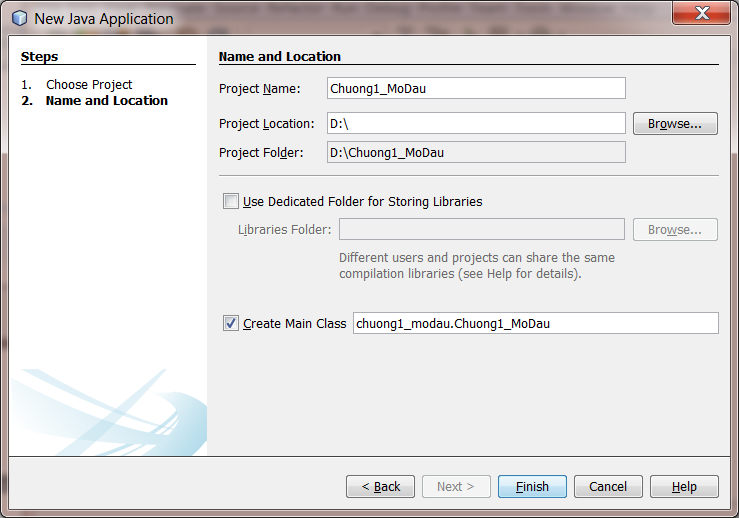
* **Khởi động NetBeans:** Sau khi download và giải nén, các bạn khởi động netBeans sẽ có giao diện như hình bên dưới:



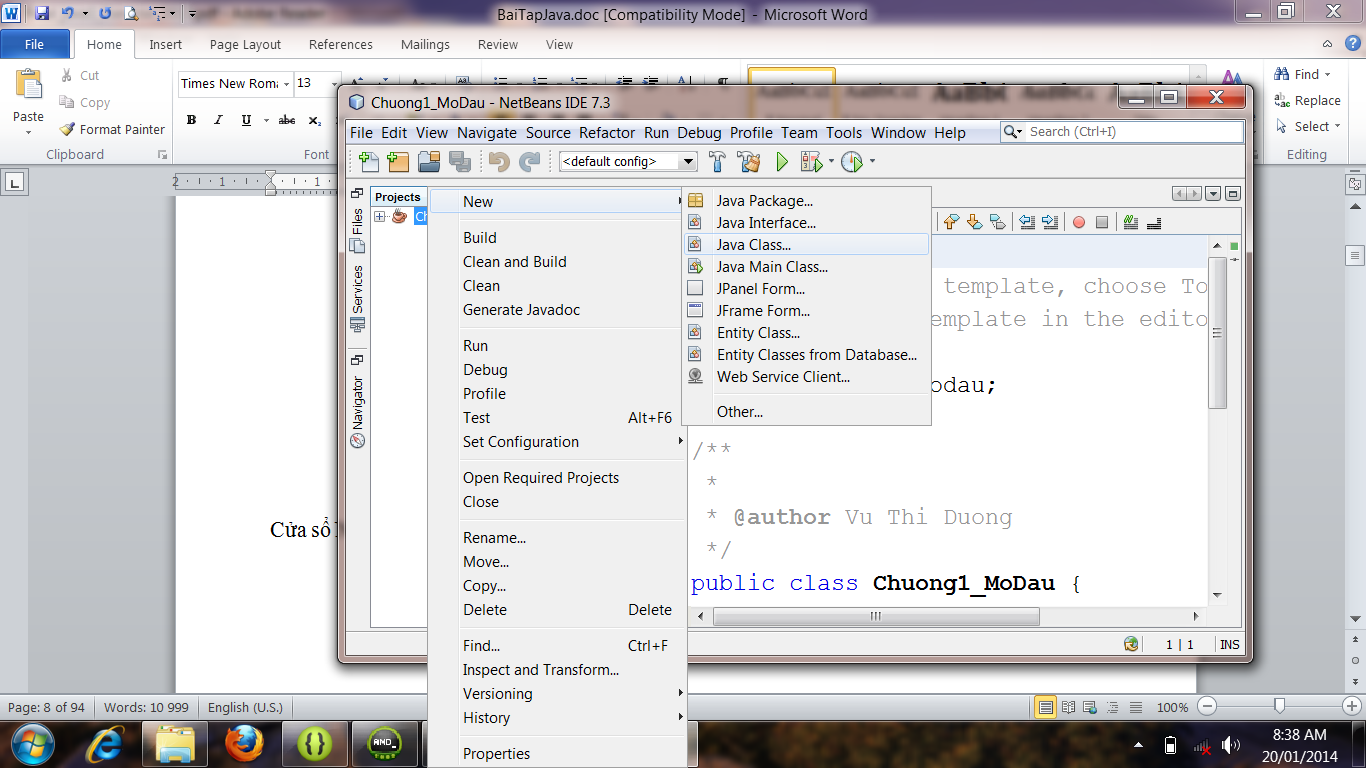
* Cách tạo Java Project: Vào menu File / New / new Project ; java->Java application;



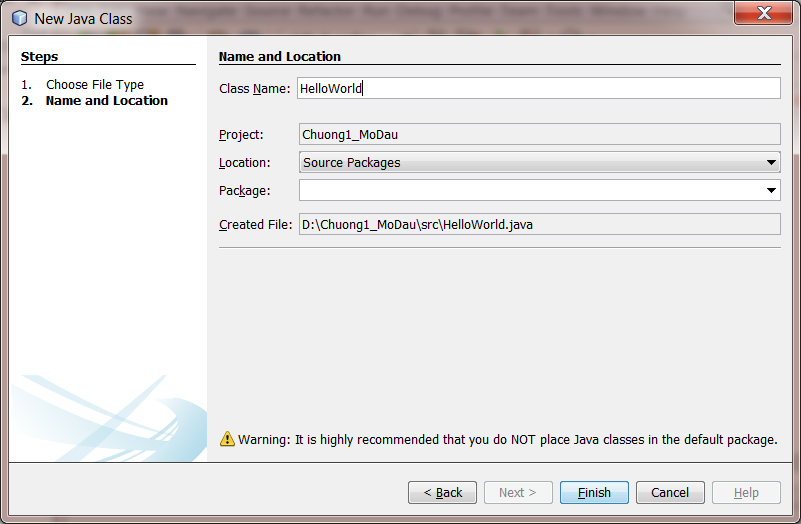
Chọn next. Màn hình hiển thị như sau



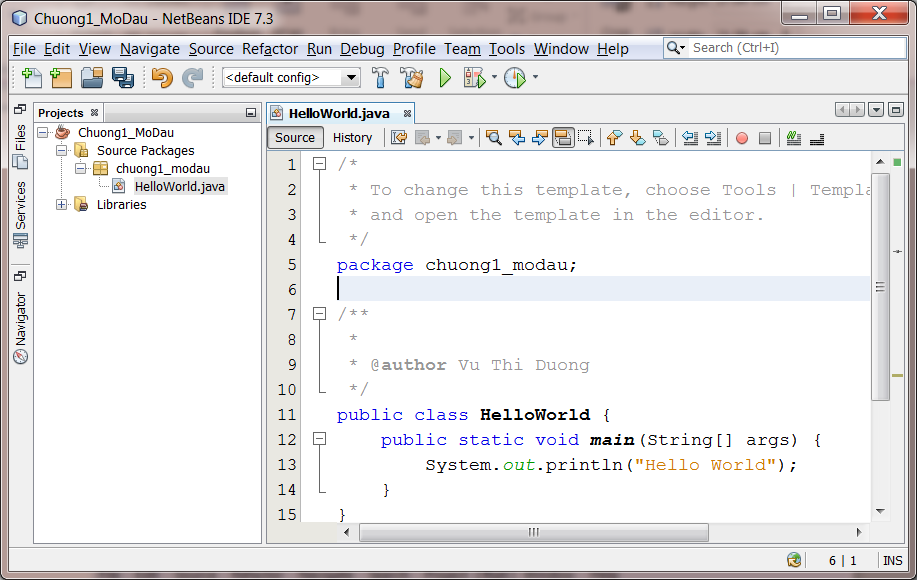
* Project Name: đặt tên prj theo quy tắc đặt tên
* Project location: chọn vị trí đặt project.
* Project folder: Mặc định có tên giống với tên prj. Bạn có thể đổi tên lại nếu cần.
* Chọn Finish kết thúc quá trình tạo mới 1 prj.
* Tạo 1 class tên là HelloWorld bằng cách bấm chuột phải vào Project / New/ chọn Class (xem hình minh họa bên dưới).



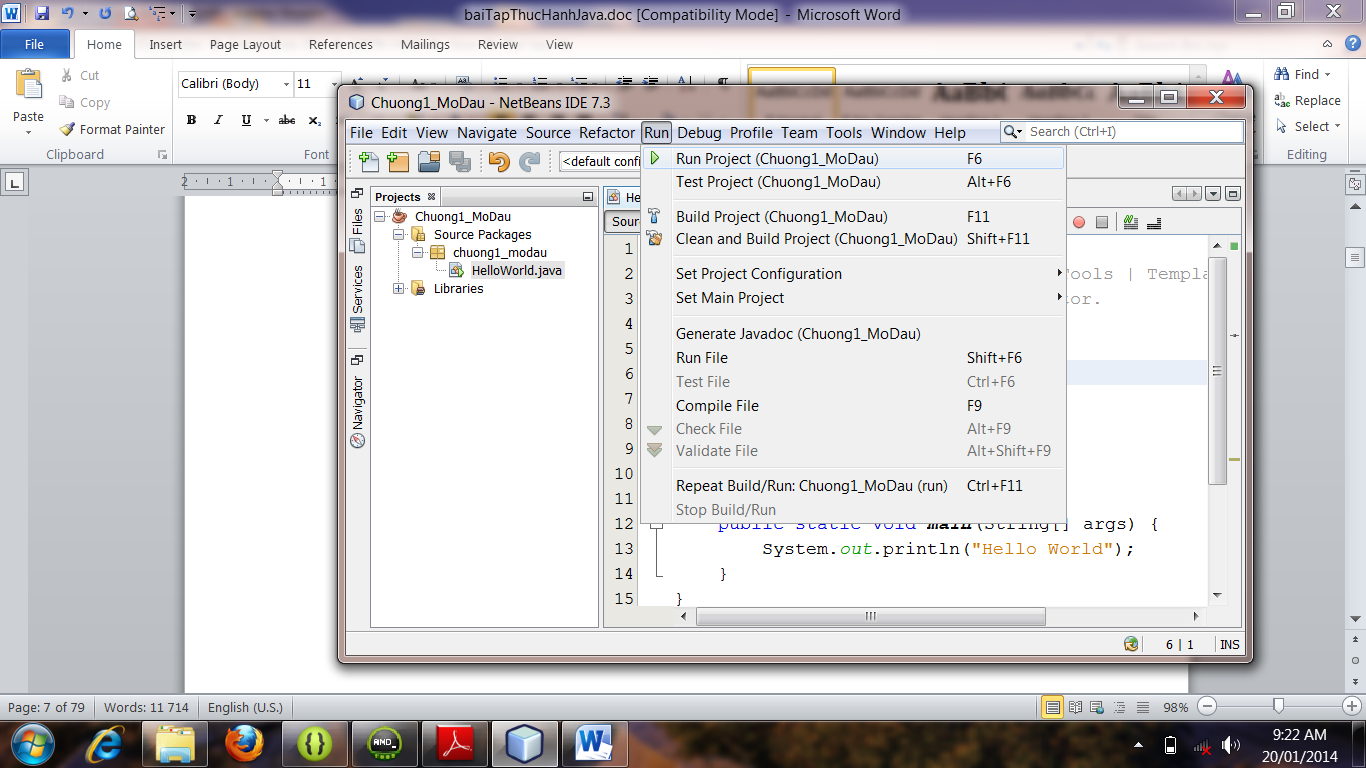
Cửa sổ New Java Class sẽ xuất hiện như bên dưới:



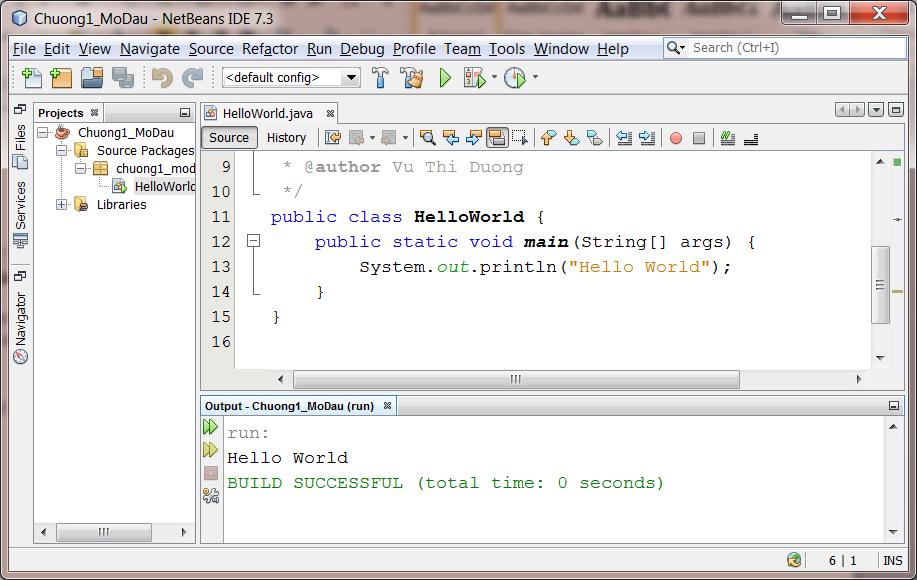
* Đặt tên cho class là **HelloWorld**. Chọn **finish**
* Các bạn quan sát trong hàm main nếu viết dòng lệnh: **sout** gõ phím Tab hoặc ctrt+ SpaceBar thì NetBeans sẽ hiệu chỉnh thành **System.out.println("");**
* Bạn hoàn thành lại: **System.out.println("Hello world!”);** Hàm này có tác dụng xuất dữ liệu ra màn hình dòng chữ “**Hello world!” .**



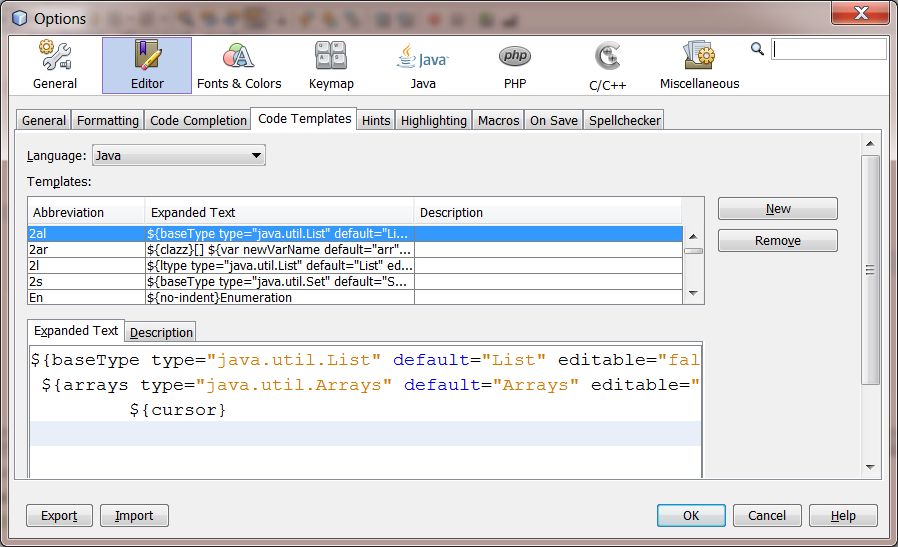
Chạy chương trình: Để biên dịch và thực thi chương trình, ta vào menu Run/Run (hoặc nhấn tổ hợp phím F6) hoặc kích phải chuột vào file **HelloWord.java**; chọn **Run file**.



Sau khi chọn Run, các bạn quan sát cửa sổ Console ở bên dưới, kết quả sẽ được hiển thị ra như sau:

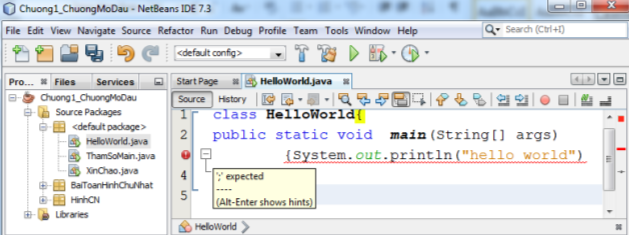


* Một số editing template hay dùng:



Gõ sout sau đó nhấn Tab sẽ cho System.out.println();

* Gõ psvm sau đó nhấn Tab sẽ cho public static void main(String[] args) {}
* Để tìm hiểu thêm, vào menu Tool\Options; mục Editor; Code Templates
* Một số chức năng cần lưu ý:
* Thực thi chương trình: Nhấn chuột phải lên lớp cần chạy, chọn menu Run file
* Mở Project trong NetBean : File\open Project; chọn prj cần mở ;
* Tên lớp luôn bắt đầu bằng 1 ký tự chữ IN HOA. Tên lớp thường được đặt theo mẫu Title-Case.
* Một vài vấn đề thường gặp
* NetBeans tự động biên dịch code và báo lỗi. Nếu bạn có lỗi hay warning thì bên trái của dòng lỗi. Ví dụ như sau:



Khi đó đưa trỏ chuột vào dấu x đỏ đầu dòng sẽ hiện thị tooltip mang nội dung lỗi.

* Export file jar tự chạy(executable jar file) trong NetBeans.
* Nhấn chuột phải lên Project cần export, chọn Clean and build. Khi kết thúc quá trình bạn nhận được đường dẫn đến file jar trong thư mục dist của project..
* Tiếp theo thực thi file jar dưới dạng câu lệnh java –jar [đường dẫn đền file .jar vủa tạo]. Ví dụ project Chuong1\_ChuongMoDau nằm trong thư mục D:\MON HOC\Java\Giai Bai Tap Java\Chuwowng1\_ChuongMoDau\ thì khi built ta cò file .jar nằm trong thư mục dist. Vậy ta thực hiện dòng lệnh như sau ở của sổ command.java -jar D:\MON HOC\Java\Giai Bai Tap\java\Chuwowng1\_ChuongMoDau\dist\Chuong1\_ChuongMoDau.jar

# Các thành phần cơ sở của java

Mục đích: Nắm vững khai báo hằng, biến, chú thích trong java

* Nắm vững định dạng xuất, nhập dữ liệu
* Nắm vững cấu trúc điều khiển

## Thao tác với toàn tử trong java

* Yêu cầu: Nắm vững cách khai báo các toán tử và cơ chế hoạt động

//tạo file **CacToanTuToanHoc.java**

public class **CacToanTuToanHoc** {

//Xây dựng 2 hàm in có chú thích số thực và số nguyên

static void **inSoNguyen**(String s, int i) { System.out.println(s + " = " + i); }

static void **inSoThuc**(String s, float f) { System.out.println(s + " = " + f); }

//Xây dựng hàm main gọi in minh họa các số thực và số nguyên

public static void main(String[] args)

{ Random random = new Random();   
int i, j, k;

//Tự động sinh các sô từ 1 đến 100   
j = random.nextInt (100) + 1;   
k = random.nextInt (100) + 1;   
inSoNguyen ("j", j);

inSoNguyen ("k" , k);

i = j + k; inSoNguyen ("j + k=", i);

i = j/ k; inSoNguyen ("j/k=", i);

i = j \* k; inSoNguyen ("j \* k=", i);

i = j % k; inSoNguyen ("j % k=", i);

j %= k; inSoNguyen ("j % k=" , j);

// kiem tra so thuc

float u, v, w;

v = random.nextFloat (); inSoThuc ("v=", v);

w = random.nextFloat (); inSoThuc ("w=", w);

u = v + w; inSoThuc ("v + w=", u);

u = v \* w; inSoThuc ("v \* w=", u);

u = v / w; inSoThuc ("v/w=", u);

u -= v; inSoThuc ("u -= v: ", u);

u \*= v; inSoThuc ("u \*= v: ", u);

u /= v; inSoThuc ("u /= v: ", u);

} }

Kết quả được xuất ra là gì? Giải thích các toán tử trên.

## Thao tác với toán tử tự động tăng giảm

Yêu cầu: Viết hàm biểu thị hoạt động của các toán rử tăng giảm.

//Tạo file **ToanTuTangGiam.java**

public class **ToanTuTangGiam**

{

public static void **main**(String[] args)

{

int i = 1;

System.out.println("i : " + i);

System.out.println("++i : " + ++i); // Pre-increment

System.out.println("i++ : " + i++); // Post-increment

System.out.println("i : " + i);

System.out.println("--i : " + --i); // Pre-decrement

System.out.println("i-- : " + i--); // Post-decrement

System.out.println("i : " + i);

}

}

Kết quả được xuất ra là gì? Giải thích.

## Thao tác với toán tử logic

Yêu cầu: Viết chương trình biểu thị giá trị của các toán tử logic trong java.

//Tạo file **ToanTuLoGic**.java có nội dung sau

public class **ToanTuLoGic**

{

public static void **main**(String[] args)

{

Random random = new Random();   
int i = random.nextInt (100);

int j = random.nextInt (100);

// Using Relational Operators

System.out.println("i = " + i);

System.out.println("j = " + j);

System.out.println("i > j là " + (i > j));   
System.out.println("i < j là " + (i < j));   
System.out.println("i >= j là " + (i >= j));   
System.out.println("i <= j là " + (i <= j));

System.out.println("i == j là " + (i == j)); S

ystem.out.println("i != j là " + (i != j)); //sử dụng toán tử phủ định NOT

System.out.println("(i < 10) && (j<10) là " + ((i<10) && (j<10)) );

System.out.println("(i < 10) || (j<10) là " + ((i<10) || (j<10)) );

}

}

Kết quả được xuất ra là gì? Giải thích

## Sử dụng lớp Scanner để nhập số; Sử dụng lớp Math.

Yêu cầu: Viết chương trình tính thể tích và diện tích bề mặt của một hình cầu với bán kính **r** nhập vào (r>=0). Kết quả chỉ cần lấy 4 chữ số thập phân. Công thức tính: Thể tích = ; Diện tích bề mặt = 4PI\*r2

**Hướng dẫn** : Dùng lớp DecimalFormat để định dạng kết quả với 4 chữ số thập phân:

DecimalFormat df = new DecimalFormat("0.####");

df.format(x)- in x theo định dạng đã định nghĩa như ở trên.

## Làm quen cấu trúc điều khiển rẽ nhánh

Yêu cầu: Viết chương trình cho nhập vào kích thước 3 cạnh a, b, c của một tam giác. Tính và xuất diện tích của tam giác, sử dụng công thức Heron. Trước khi tính phải kiểm tra xem 3 số a, b, c có lập thành một tam giác không (a, b, c lập thành tam giác khi tổng 2 số bất kỳ luôn lớn hơn số còn lại).

**Hướng dẫn** : Diện tích =, biết **s** bằng 1/2 chu vi tam giác.

## Làm quen cấu trúc điều khiển rẽ nhánh

Yêu cầu: Viết chương trình tính khoảng cách giữa 2 điểm có tọa độ (x1,y1) và (x2,y2), biết công thức để tính khoảng cách là: distance = 

## Làm quen cấu trúc chọn.

Yêu cầu: Viết chương trình nhập vào tháng, năm dương lịch bất kỳ. in ra số ngày của tháng đó.

## Làm quen cấu trúc chọn

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 tin ra chính a và b.

## Làm quen với cấu trúc lặp

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.

## Làm quen với cấu trúc lặp

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

## Thao tác với xâu ký tự - lấy 1 ký tự trong xâu

Yêu cầu: Viết chương trình cho nhập một chuỗi và in mỗi ký tự trong chuỗi trên một dòng.

**Hướng dẫn**:

*Dùng hàm charAt(int) để lấy ra kí tự tại vị trí trong chuỗi*

*Scanner in = new Scanner(System.in);*

*System.out.println("nhap vao 1 chuoi:");*

*String s = in.nextLine();*

*System.out.println("In chuoi ra man hinh");*

*for (int i = 0; i < s.length(); i++) System.out.println(s.charAt(i));*

## Thao tác với xâu ký tự.- kiểm tra ký tự trắng trong xâu

Yêu cầu: Viết chương trình cho nhập một chuỗi và đếm số khoảng trắng có trong chuỗi đó.

**Hướng dẫn**: *Dùng hàm isSpace(char) của lớp Character để kiểm tra khoảng trắng*

## Thao tác với xâu ký tự - kiểm tra ký tự trong xâu

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.*

## Thao tác với xâu ký tự- 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**)

## Thao tác với xâu ký tự- 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”.

## Thao tác với xâu ký tự - 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.

## Xây dựng phương thức tĩnh gọi trong hàm main()

Yêu cầu: Viết chương trình giải phương trình bậc hai ax2 + bx + c = 0,

**Hướng dẫn**:

* Viết hàm giaiPTB1(float a, float b)- giải phương trình bậc 1 với hệ số a,b
* Viết hàm giải PT2(float a, float b, float c)- gọi đến hàm giaiPTB1 nếu a=0.

## Xây dựng phương thức

Yêu cầu: Viết hàm sumRange, hàm có hai tham số kiểu số nguyên biểu thị cho khoảng giá trị. Nếu tham số thứ hai nhỏ hơn tham số thứ nhất thì xuất ra một thông báo lỗi và trả về giá trị 0, ngược lại hàm trả về tổng của các giá trị trong khoảng đó.

Test hàm vừa viết bằng cách thực hiện vài lần, mỗi lần với các giá trị truyền khác nhau.

## Xây dựng phương thức

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), tức là 13 =8+5, 21=13+8 ,

- 2 số Fib đầu tiên trong dãy luôn luôn là 1

## Xây dựng phương thức

Yêu cầu: xây dựng phương thưc countA(), phương thức này nhận một tham số kiểu String và trả về số lần xuất hiện ký tự “A‟ trong chuỗi đó. Test hàm vừa viết bằng cách thực hiện vài lần, mỗi lần với giá trị truyền khác nhau.

## Xây dựng phương thức

Yêu cầu: Hãy viết phương thức tên **Panlyndrome** nhận 1 tham số có kiểu String và trả về true nếu tham số là chuỗi panlyndrome (chuỗi panlyndrome là chuỗi có chuỗi đảo ngược bằng chính nó)

**Hướng dẫn**:

public static boolean isPanlyndrome(String s)   
 { for(int i=0;i<s.length();i++)

if(s.charAt(i)!=s.charAt(s.length()-i-1)) return false;

return true;

}

## Nạp chồng hàm.

Yêu cầu:

1. Viết một phương thức có tên square, hàm có 1 tham số x kiểu số thực. Phương thức này tính toán và trả về bình phương của x.
2. Đưa phương thức trên vào một chương trình và nó được gọi thực hiện bởi hàm main. Test hàm vừa viết bằng cách thực hiện vài lần, mỗi lần với các giá trị truyền khác nhau.

**Hướng dẫn:**

* Trong lớp chứa hàm main khai báo hàm theo yêu cầu (chú ý có phạm vi static)
* Trong hàm main, khai báo số thực, nhập số thực vào từ bàn phím , truyền tham số cho hàm và và gọi phương thức vừa xây dựng, hiển thị kết quả.

## Nạp chồng hàm.

Yêu cầu: xây dựng phương thức có một tham số nguyên, phương thức trả về giá trị false nếu số đó là số chẵn, ngược lại trả về giá trị true.

Test phương thức vừa viết bằng cách thực hiện vài lần, mỗi lần với các giá trị truyền khác nhau.

## Nạp chồng hàm.

Yêu cầu:

* Viết phương thức **average** có hai tham số kiểu số nguyên. Phương thức trả về giá trị trung bình của hai số đó.
* Overload hàm **average** với ba tham số nguyên, hàm trả về giá trị trung bình của ba số

Test các hàm vừa viết bằng cách thực hiện vài lần, mỗi lần với các giá trị truyền khác nhau.

# Lớp và các thành phần của lớp các đối tượng

Nội dung kiến thức thực hành:

- Khai báo và sử dụng lớp

- xây dựng hàm tạo có đối và không đối.

- Viết các lớp với mối quan hệ composition

- Cách truyền tham số trong Java

- Cách sử dụng từ khóa static

- Tạo lớp con kế thừa từ lớp cha .

- Đa hình thông qua kế thừa

- Viết Interface và thực thi

## Tìm hiểu cách viết lớp trong Java.

Yêu cầu: Cho lớp Hình chữ nhật như sau:

public class HinhChuNhat1Lop

{ private float dai,rong;

public HinhChuNhat1Lop() {

this.dai = 5f;

this.rong = 6f;

}

public float dienTich() { return dai\*rong; }

@Override

public String toString() {

return "HCN + dai= " + dai + ", rong= " + rong + " dien tich= "+dienTich();

}

public static void main(String[] args) {

HinhChuNhat hcn=new HinhChuNhat();

System.out.println(hcn);

}

}

* Tìm hiểu lớp này và giải thích kết quả. Nếu chương trình bị lỗi, hãy sửa lỗi.
* Tách hàm main ra lớp riêng tên là DistaceDemo. Chạy và so sánh kế quả.

## Biết cách viết và sử dụng ghi đè hàm tạo.

Yêu cầu:Trong bài 1 đã mô tả lớp hình chữ nhật có thuộc tính dài, rộng; có phương thức khởi tạo không đối, tính diện tích hình chữ nhật.

Hãy xây dựng thêm hàm tạo có 2 đối số. Viết lớp chứa hàm main() gọi và minh họa các phương thức trong lớp hình chữ nhật trên. (hàm main viết trong 1 lớp khác).

* Mở rộng bài toán. Hàm main khai báo và nhập vào một mảng n hình chữ nhật. In danh sách hình chữ nhật sau khi nhập. Nỗi hình chữ nhật trên 1 dòng

## Xây dựng lớp và hàm thành phần

**Yêu cầu**: Viết lớp **HinhTron** dùng để tính diện tích và chu vi hình tròn từ bán kính. Viết hàm main để kiểm tra lớp này.

**Hướng dẫn:**  khai báo lớp hình tròn gồm các hàm khởi tạo, tính chu vi, diện tích . Xây dựng 1 lớp độc lập chứa hàm **main** để kiểm nghiệm lớp hình tròn trên.

## Xây dựng lớp và hàm thành phần

Yêu cầu: Cho các lớp sau, hãy tìm hiểu và giải thích cách sử dụng lớp **NgayThang**, lớp **NhanVien**. Cho biết kết quả của chương trình là gì?

**// 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 c**heckDay**( 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 calss 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;

}

@Override

public String toString()

{ return ten + ", " + hoDem + " ngay vao lam: " + ngayVaoLam.toString() + " sinh ngay: " + ngaySinh.toString();

}

}

**//** NhanVienDemo**.java**

import javax.swing.JOptionPane;   
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 );

}

}

//thử cài đặt lại ngày sai và tháng sai, xem bạn nhận được thông báo gì?.

## Tìm hiểu cách truyền tham số cho hàm trong Java.

Yêu cầu: Cho các lớp dưới đây, cho biết kết quả chương trình và giải thích. // Num.java

//tạo file Num.java

**public class Num**   
{

private int value;

public **Num** (int update) { value = update; }

public void  **setValue** (int update) { value = update; }

public String **toString** () { return value + ""; }

}

// ParameterTester.java

**public class ParameterTester**   
{

public static void changeValues (int f1, Num f2, Num f3)   
{

System.out.println ("Before changing the values:");

System.out.println ("f1\tf2\tf3");

System.out.println (f1 + "\t" + f2 + "\t" + f3 + "\n"); f1 = 999;

f2.setValue (888);

f3 = new Num (777);

System.out.println ("After changing the values:");

System.out.println ("f1\tf2\tf3");

System.out.println (f1 + "\t" + f2 + "\t" + f3 + "\n");

}

public static void main (String[] args)   
{

int a1 = 111;

Num a2 = new Num (222);   
Num a3 = new Num (333);

System.out.println ("Before calling changeValues:");

System.out.println ("a1\t a2\t a3");

System.out.println (a1 + "\t" + a2 + "\t" + a3 + "\n");

changeValues (a1, a2, a3);

System.out.println ("After calling changeValues:");

System.out.println ("a1\t a2\t a3");

System.out.println (a1 + "\t" + a2 + "\t" + a3 + "\n");

} }

## Hiểu cách sử dụng thuộc tính và hàm static.

**Yêu cầu**: Cho các lớp dưới đây, cho biết kết quả chương trình và giải thích.

// HoSo.java

**class HoSo {**

static int soNguoi;   
String hoTen;

**HoSo**( String ht ){ hoTen = ht; **soNguoi++;** }

static void **tongKet()**{ System.out.println( "Ho khau 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();

}

}

## Xây dựng lớp và hàm thành phần.

Yêu cầu: (i) Xây dựng lớp CHinhTamGiac như sau:   
**ChinhTamGiac.java**

- ma : int

- mb : int

- mc : int

+ CHinhTamGiac()

+ CHinhTamGiac(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() : boolean

+ laTamGiac(int, int, int) : boolean

+ getChuVi() : int

+ getDienTich() : double

**Trong đó:** Thộc tính ma, mb, mc là ba cạnh của tam giác.

Khởi tạo CHinhTamGiac(int, int, int): yêu cầu kiểm tra:

- nếu giá trị truyền có số âm thì thông báo và gán thuộc tính tương ứng bằng 0;

- nếu 3 giá trị truyền vào không lập thành một hình tam giác thì gán 3 thuộc tính bằng 0.

**Hướng dẫn**:

* Các giá trị lập thành một hình tam giác khi và chỉ khi các canh 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():** trả về giá trị true khi ba giá trị **ma, mb, mc** lập thành một hình tam giác, ngược lại trả về giá trị **false**.
* **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**.

**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 giac, giá trị chu vi và diện tích của nó.
* Mở rộng bài toán: Nhập vào 1 mảng tam giác, in mảng sau khi nhập

## Xây dựng lớp và hàm thành phần.

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

- maNV : String

- soSP : int

+ NhanVien()

+ NhanVien (ma: String, sp : int)

+ coVuotChuan() : boolean   
+ getTongKet() : String   
+ getLuong(): double   
+ toString() : String

* Trong các hàm **khởi tạo có đối** 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 20000, và nếu số sản phẩm của nhân viên vượt chuẩn thì phần vượt chuẩn được tính đơn giá là 30000.
* toString() : trả về chuỗi chứa thông tin nhân viên gồm các cột như nội dung tiêu đề.

**(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.

**(c)**. Mở rộng bài toán: Nhập vào 1 mảng Nhân viên, in mảng sau khi nhập

Hướng dẫn:

public class NhanVien {

private String maNV;

private int soSP;

//Tự sinh constructor và sửa lại như sau

public NhanVien() {maNV=””; soSP=0;}

public NhanVien(String maNV, int soSP) {

this.maNV = maNV;

this.soSP = soSP;

}

//Tự động sinh các hàm SET\_GET

public String getMaNV() {return maNV; }

public void setMaNV(String maNV) {this.maNV = maNV;}

public int getSoSP() {return soSP;}

public void setSoSP(int soSP) {

if(soSP>0)

this.soSP = soSP;

else

this.soSP = 0;

}

// ham kiem tra vuot chuan

public boolean coVuotChuan(){

if(soSP > 500) return true;

return false;

}

public String getTongKet(){

if(soSP > 500) return "Vuot";

return "";

}

public double getLuong(){

if(soSP > 500)

return (soSP-500)\*30000 + 500\*20000;

return soSP\*20000;

}

public static void xuatTieuDe(){

System.out.println("MaNV\t\tSoSP\t\tLuong\t\tTongKet");

}

@Override

public String toString() {

DecimalFormat df=new DecimalFormat("###,###.0#");

return maNV +"\t\t" + soSP + "\t\t" + df.format(getLuong()) + "\t\t" + getTongKet();

}

## Xây dựng lớp và các hàm thành phần

Yêu cầu:

1. **Xây dựng lớp CNhanVien, biết:**

* mHo, mTen, mSoSP lần lượt là các thuộc tính họ, tên và số sản phẩm của nhân viên.
* Viết phương thứckhởi 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.
* Viết các phương thứclấy và gán giá trị cho thuộc tính của lớp (**các hàm get/set).**
* Viết hàm **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.
* Viết phương thức**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

**(ii) 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()**.

## Xây dựng lớp và các hàm thành phần.

Yêu cầu: Xây dựng lớp **SinhVien** với các thuộc tính của sinh viên là mã sinh viên, họ tên, điểm lý thuyết, điểm thực hành.

Các phương thức: Hàm dựng không đối- khởi gán các giá trị mặc định cho thuộc tính

* hàm dựng có đầy đủ các đối- khởi gán giá trị như truyền vào
* hàm nhap()- nhập thông tin Sinh viên và tính luôn điểm trung bình = (đlt+dth)/2).
* ghi đè phương thức toString()- kết xuất xâu mô tả Sinh Viên

Viết hàm main cho phép thực hiện:

* Nhập thông tin cho một 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).

## Xây dựng lớp và các hàm thành phần.

Xây dựng lớp NhanVien, biết cấu trúc của một nhân viên như sau:

- Mã NV: kiểu số nguyên

- Họ tên: kiểu chuỗi.

- Địa chỉ: kiểu chuỗi.

- CBQL: kiểu logic, có giá trị true nếu nhân viên này là cán bộ quản lý.

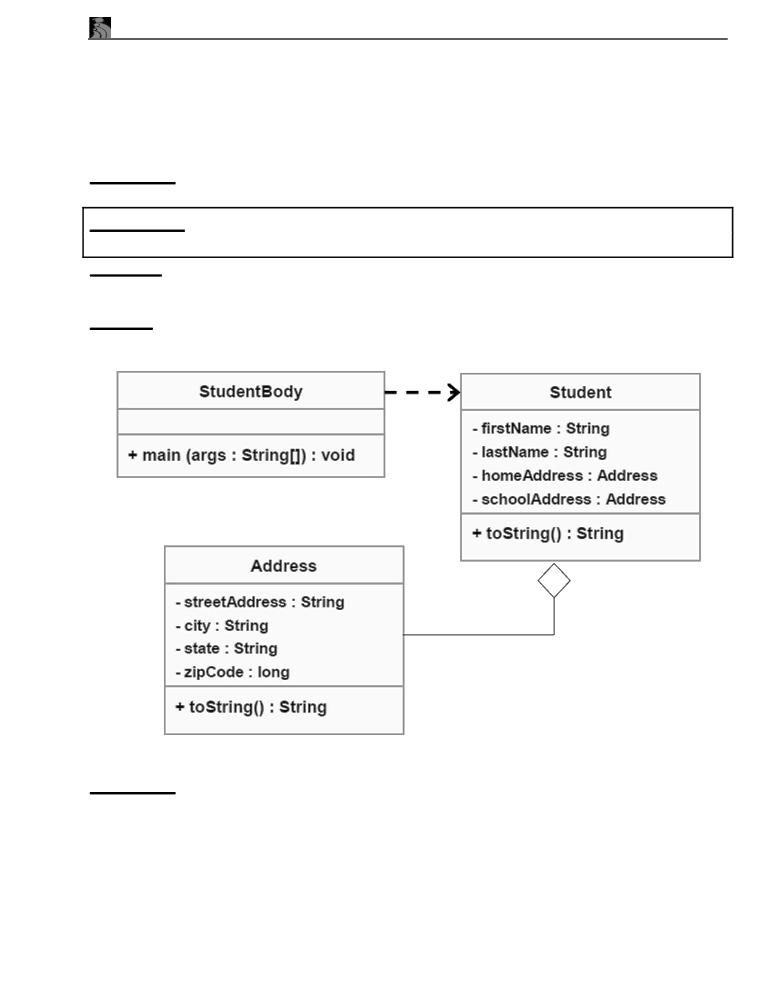
Các phương thức:

* hàm dựng có đầy đủ các đối- khởi gán giá trị như truyền vào
* hàm nhap()- nhập thông tin NhanVien và tính luôn điểm trung bình = (đlt+dth)/2).
* ghi đè phương thức toString()- kết xuất xâu mô tả NhanVien

Viết hàm main cho phép khởi tạo một số đối tượng thuộc lớp NhanVien, sau đó in ra mã, họ tên của các nhân viên là cán bộ quản lý.

## Xây dựng lớp có mối quan hệ cấu thành, tập hợp

Yêu cầu: Cho lược đồ các lớp như sau :



Trong đó sinh viên có hai thuộc tinh **địa chỉ nhà và địa trường** là thuộc tính kết nối (kiểu đối tượng) . Lớp **StudentBody** chứa hàm main. Hãy cài đặt mô hình trong đó hàm main yêu cầu nhập vào một mảng Student và xuất ra mỗi sinh viên trên 1 dòng.

Hướng dẫn: Xây dựng mỗi lớp là 1 file .java.

* Cài đặt lớp Address.java
* Cài đặt lớp Student.java
* Cài đặt lớp StudentBody có thuộc tính là: số sinh viên trong danh sách, mảng sinh viên; xây dựng hàm main nhập số sinh viên, khởi tạo danh sách, nhập danh sách- in danh sách.

## Cách viết lớp kế thừa.

Yêu cầu: Cho các lớp sau:

**// Vehicle.java**

**public class Vehicle**

{ private int soBanhXe;

public Vehicle(int x){ soBanhXe=x;}

public void laiXe(){ System.out.println(“ lai xe co so banh la “+soBanhXe);}

}

**//khai bao lop ke thua**

**public class Car extentds Vehicle**

{ private int soCua;

public Car(int d, int x) { super(x); soCua=d;}

public void laiXe(){ System.out.println(“lai xe co so banh la”+ soBanhXe+” co “+soCua);}

}

a. Bổ sung các hàm get/set cho các lớp.

b. Lớp Car bị lỗi biên dịch. Hãy sửa lỗi.

c. Giả sử lớp Car được sửa lỗi, kết quả chương trình là gì khi hàm **main** viết:

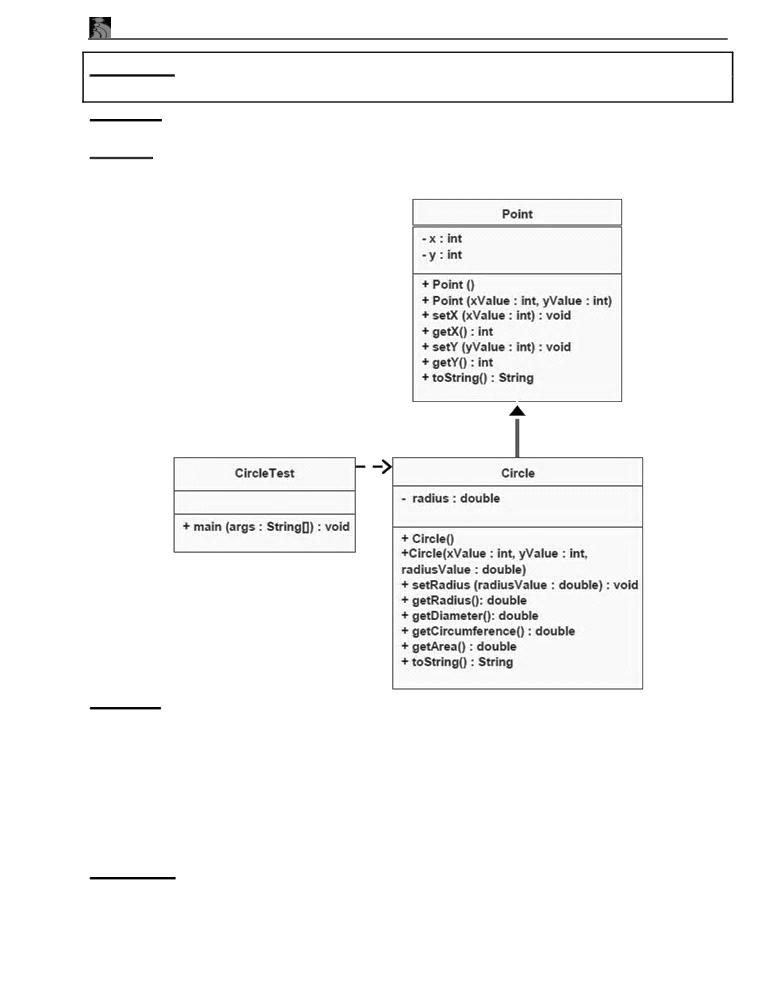
TH1:

Car audi = new Car(); Car.drive();

TH2:

Car p = new Car(2, 4); p.drive();

## Cài đặt cho lược đồ sau:



Trong đó:

+ toString(): xuất giá trị x, y, radius của Point hoặc Circle tương ứng (x, y xuất kiểu tọa độ [x,y])

+ setRadius(): kiểm tra giá trị đưa vào.

+ getDiameter(): tính đường kính hình tròn.

+ getCircumference(): tính chu vi hình tròn.

+ getArea(): tính diện tích hình tròn.

+ Hàm main() yêu cầu nhập một mảng **n đối tượng Circle**, xuất ra đường kính, chu vi, diện tích của chúng, kết quả hiển thị **2 số lẻ**.

**Hướng dẫn**: Xem lại **Hướng dẫn** hiển thị số thực và cách dùng mảng ở các bài trước

## Đa hình thông qua kế thừa

Yêu cầu:

a. 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”.

b. 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 \*.

c. 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].

- Trong lớp **RightTriangle**, viết lại hàm **draw**() của lớp **Shape** để vẽ ra hình tam giác vuông bởi các dấu \*, với số lượng dấu \* mỗi cạnh tùy thuộc vào các thuộc tính cạnh vuông. Ví dụ thuộc tính cạnh vuông là 4 thì hàm **draw**() vẽ được hình: tam giác bằng dấu \*.

d. Viết lớp **Artist**, lớp này có hàm drawShape(), hàm này không trả về giá trị và có một tham số kiểu Shape, trong hàm này gọi hàm draw() để vẽ hình tùy theo tham số Shape truyền vào.

e. Viết lớp **ArtistDemo**, lớp này chứa hàm **main**(): khai báo biến **Shape**; dùng biến này để khởi tạo cho đối tượng **Rectangle** với chiều dài là 8, chiều rộng là 4; tạo đối tượng **Artist** để vẽ hình chữ nhật này; dùng lại biến Shape để khởi tạo cho đối tượng **RightTriangle** với cạnh là 20, dùng đối tượng **Artist** để vẽ hình tam giác này. Viết lại hàm main này với các giá trị chiều dài, chiều rộng, cạnh vuông do người dùng nhập vào.

## Tìm hiểu cách viết và sử dụng lớp trừu tượng

Yêu cầu:

* Khai báo lớp trừu tượng HìnhPhang
* Cài đặt các lớp theo mô hình dưới đây:

**class HìnhPhẳng**

**+float tinhCV();**

**+float tinhDT();**

**class HìnhTròn**

**+float tinhCV();**

**+float tinhDT();**

**class HCN**

**+float tinhCV();**

**+float tinhDT();**

**Class HìnhVuông**

**+float tinhCV();**

**+float tinhDT();**

**Hướng dẫn:**

class HinhVuong extends HinhPhang

{ public float canh;

HinhVuong(float canh)

{ super();

this.canh=canh;

}

public float tinhCV() { return 4\*canh; }

public float tinhDT() { return canh\*canh; }

}

**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**

class HinhCN extends HinhPhang

{ public float dai, rong;

HinhCN (float dai, float rong)

{

super();

this.dai=dai;

this.rong=rong;

}

public float tinhCV() { return 2\*(dai+rong); }

public float tinhDT() { return dai\*rong; }

}

* Viết hàm main minh họa: khai báo mỗi lớp con kế thừa một đối tượng, nhập, xuất dữ liệu của chúng.

public class HinhPhangDemo

{ public static void main(String[] args)

{

HinhVuong v=new HinhVuong(60);

System.out.println("chu vi hinh vuong ="+ v.tinhCV()+ " dien tich="+v.tinhDT());

HinhCN cn=new HinhCN (15,20);

System.out.println("chu vi chu nhat ="+ cn.tinhCV()+ " dien tich="+cn.tinhDT());

HinhTron tr=new HinhTron (15);

System.out.println("chu vi hinh tron ="+ cn.tinhCV()+ " dien tich="+cn.tinhDT());

}

}

## Xây dựng lớp trừu tượng

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 lượng – 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
* Tìm lớp cha và các phương thức trừu tượng
* Xây dựng lớp con kế thừa cụ thể hóa phương thức trừu tượng

## Xây dựng lớp trừu tượng

Công ty Media Net quản lý thông tin thuê bao Internet của khách hàng. Thuê bao được chia làm 2 nhóm: huê bao trọn gói và thuê bao theo dung lượng. Thông tin chung của các thuê bao bao gồm: + Họ tên ,+ Địa chỉ ,+ Sốđiện thoại  
Ngoài ra, mỗi loại thuê bao còn có thông tin riêng sau:   
Loại thuê bao:

- Thuê bao trọn gói (số MB sử dụng = -1)   
- Thuê bao theo dung lượng. (số MB sử dụng)

Công ty quy định cách tính tiền cước hàng tháng như sau:

- Với thuê bao trọn gói: Cước = 350.000đ   
- Với thuê bao theo dung lượng: Cước = tiền thuê bao (30.000đ) + cước dung lượng.

Cước dung lượng tính như sau:

2000MB đầu: giá 55đ/MB   
Các MB tiếp theo: giá 41đ/MB   
 Nếu tiền cước vượt quá mức trần 400.000đ thì chỉ tính 400.000đ

VD: Khách hàng sử dụng 1900MB s có cước là: 30.000 + 1900 \* 55 = 134.500đ   
Khách hàng sử dụng 3000MB s có cước là: 30.000 + 2000 \* 55 + (3000 – 2000) \* 41 = 181.000đ

**Yêu cầu:**a) Xây dựng các lớp với các thành phần dữ liệu tương ứng. Xây dựng các hàm constructor cho các lớp đó. (ít nhất 2 constructor : không tham số và đầy đủ tham số)

b) Xây dựng hàm nhập/ xuất thông tin của các thuê bao.

c) Xây dựng hàm tính tiền cước cho các thuê bao.

d) Xây dựng lớp CongTyMediaNet chứa danh sách gồm n thuê bao. iết hàm main liệt kê thông tin (họ tên, địa chỉ, sốđt và tiền cước) của các thuê bao có mức tiền cước cao nhất.

## Xây dựng lớp trừu tượng

Công ty Suối Nhạc quản lý các ca sỹ theo 2 nhóm: ca sỹ “chưa” nổi tiếng và ca sỹ nổi tiếng.

Thông tin chung của cả 2 nhóm bao gồm:   
+ Họ tên   
+ Số năm làm việc   
+ Số đĩa bán được   
+ Số buổi trình diễn

Ngoài ra, ca sỹ nổi tiếng được mời tham gia nhiều gameshow nên còn có thêm thông tin:   
số gameshow tham gia   
Quy định trả lương như sau:

- Với ca sỹ “chưa” nổi tiếng:

Lương = 3.000+ 500\* số năm làm việc + số đĩa bán được + 150\* số buổi trình diễn   
( Nếu sốđĩa bán được > 3000 đĩa thì sẽđược thư ng thêm 1.000 )

- Với ca sỹ nổi tiếng:

Lương = 5.000+ 500\* số năm làm việc + 1.2 \* sốđĩa bán được + 300\* số buổi trình   
diễn + 500\* số gameshow

**Yêu cầu:**  
a) Xây dựng các lớp với các thành phần dữ liệu tương ứng. Xây dựng các hàm constructor cho các lớp đó. (ít nhất 2 constructor : không tham số và đầy đủ tham số)   
b) Xây dựng hàm nhập/ xuất thông tin của các ca sỹ.   
c) Xây dựng hàm tính tiền lương cho các ca sỹ.   
d) Xây dựng lớp CongTySuoiNhac chứa danh sách gồm n ca sỹ. Viết hàm Nhập/Xuất cho lớp này.   
e) Viết hàm liệt kê thông tin (họ tên, số năm làm việc, sốđĩa bán được, số buổi trình diễn và tiền lương) của các ca sỹ có mức tiền lương cao nhất.

## Đa kế thừa

Yêu cầu: Cài đặt các lớp theo mô hình dưới đây:

**Abstract class Animal**

-name, weight

+ grow(),

+ show()

**interface Flyable**

+fly()

**Class Chicken**

-name, weight

+grow(), +show()

**Class Sparrow**

* name, weight

+ grow(),

+show()

* Xây dựng lớp trừu tượng Animal có dữ liệu là name, weight phương thức **show**(), phương thức trừu tượng **grow** (tăng trọng lượng),
* Xây dựng giao diện Flyable có phương thức **fly**()
* Xây dựng lớp Chicken kế thừa lớp Animal cài đặt phương thức grow để mỗi lần grow thì tăng 0.3 kg. Lớp Sparrow kế thừa lớp Animal mỗi lần tăng trọng lượng 0.05kg và hiện thực hóa giao diện Flyable

**abstract** class Animal

{

String name;

double weight;

public Animal(String name, double weight)

{

this.name = name;

this.weight = weight;

}

**abstract** void grow();

public void show(){ …..}// sinh viên tự triển khai nội dung lớp

}

**interface Flyable{**

public void Fly(); // chỉ định nghĩa phương thức mà không triển khai nội dung

}

**class Chicken extends Animal**

{

public Chicken(String name, double weight) { super(name,weight); }

public void **grow**()

{ weight = weight + 0.3d; }

}

**class Sparrow extends Animal implements Flyable**

{

public **Sparrow**(String name, double weight)

{ super(name, weight); }

public void grow()

{ weight += 0.05d; }

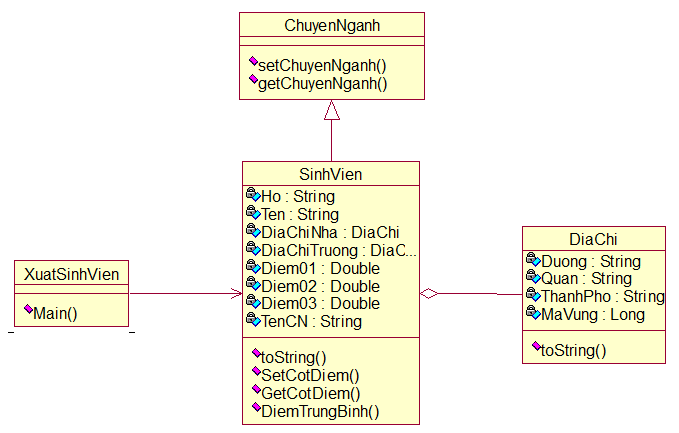
public void Fly()

{ System.out.println("Sparrow can fly"); }

}

## Cài đặt và hiện thực hóa interface

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

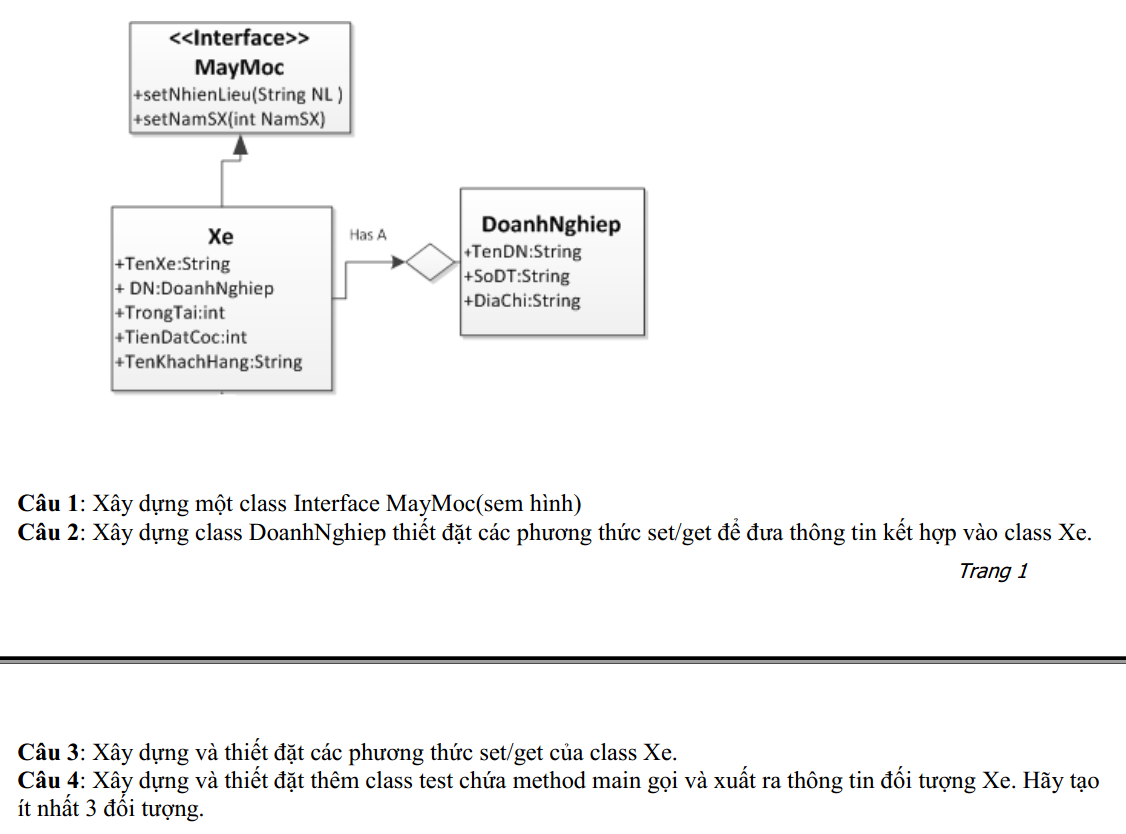


Yêu cầu:

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

## Xây dựng và hiện thực đa kế thừa

Cho sơ đồ lớp như sau:



Yêu cầu:

1. Xây dựng giao diện MayHọc như hình vẽ
2. Xây dựng lớp DoanhNghiệp thiết đặt các phương thức get/set để đưa thông tin kế hợp vào lớp Xe trong câu 3.
3. Xây dựng lớp Xe có các thuộc tính như gợi ý trong đó có thuộc tính kết nối với lớp DoanhNghiep
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

# Lỗi thực thi và quá trình gom rác

Nội dung kiến thức thực hành:

- Xử lý dữ liệu với try-catch

- Ném ngoại lệ

- Tạo lớp ngoại lệ riêng

## Kiểm tra dữ liệu nhập của người dùng bằng cách sử dụng try-catch.

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("Loi: khong phai so!") ;

}

catch (ArithmeticException e)

{

System.out.println("Loi: phep chia cho 0!");

}

}

public static void main(String[] args) {

tinh(); }}

## Kiểm tra dữ liệu nhập của người dùng bằng cách viết lớp xử lý ngoại lệ.

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("Du lieu phai la so!"); }

}

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("Loi: phep 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();

}

}

}

## Kiểm tra dữ liệu nhập của người dùng bằng cách viết lớp xử lý ngoại lệ.

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.

## Kiểm tra dữ liệu nhập của người dùng bằng cách viết lớp xử lý ngoại lệ.

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.

## 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 OutOfRangException. 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 OutOfRangException để thông báo lỗi.

# Các lớp cơ sở và cấu trúc dữ liệu

Nội dung kiến thức thực hành:

- 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 với Comparable interface

## Sử dụng mảng – sắp xếp phần tử trong mảng.

Viết chương trình cho nhập mảng n số nguyên với các giá trị ngẫu nhiên. Xuất tổng, giá trị nhỏ nhất của mảng. Sắp xếp mảng.

**Yêu cầu** viết hàm để xử lý mảng và viết lớp riêng chứa hàm main gọi hàm đó.

## Sử dụng mảng - sắp xếp sử dụng lớp RandomArray

Làm lại bài tập 1 bằng cách viết lớp RandomArray để xử lý mảng.

HD: lớp RandomArray có 2 thuộc tính là mảng số nguyên và vị trí hiện hành, trong đó viết hàm thêm 1 phần tử vào mảng tại vị trí hiện hành, hàm in mảng, hàm tính tổng mảng, hàm tìm giá trị nhỏ nhất, hàm sắp xếp mảng.

## Sử dụng tập hợp- dùng ArrayList thay cho mảng thông thường

Yêu cầu: Làm lại bài tập 2 bằng cách sử dụng ArrayList thay cho mảng thông thường.

**Hướng dẫn**: lớp RandomArray chỉ có 1 thuộc tính là mảng số nguyên.

## Các thao tác trên tập hợp- nhập xuất, sắp xếp mảng sử dụng thuật toán trên tập hợp

Mục đích:

Yêu cầu: Thao tác trên mảng số nguyên, viết hàm để thực hiện các yêu cầu sau:

- Nhập / xuất mảng nguyên bất kỳ

- Tìm kiếm giá trị bất kỳ nhập vào từ bàn phím

- Xuất tất cả các số nguyên tố trong mảng

- Sắp xếp mảng: chọn một trong các cách- BubbleSort, SelectionSort, InsertionSort, QuickSort

- Xóa một giá trị bất kỳ trong mảng

- Kiểm tra mảng có đối xứng hay không

## Viết và sử dụng lớp mảng đối tượng- tìm, thêm, xóa phần tử.

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.

## Thao tác thêm, sửa, xóa, tìm thông tin trong tập hợp

Để quản lý thông tin về hàng hóa người ta mô tả các mặt hàng như sau:

* Hàng điện máy <mã hàng, tên hàng, nhà sản xuất, giá, thời gian bảo hành, điện áp, công suất>
* Hàng sành sứ < mã hàng, tên hàng, nhà sản xuất, giá, loại nguyên liệu>

Viết chương trình thực hiện

* Nhập vào một danh sách các mặt hàng điện máy. Quá trình nhập dừng lại khi mã hàng trống.
* In danh sách các mặt hàng vừa nhập
* Nhập vào một mã hàng và kiểm tra hàng có trong danh sách không. Nếu có thì sửa mặt hàng đó. Yêu cầu mã mới sửa phải chưa xuất hiện trong danh sách hàng. In danh sách sau sửa.
* Nhập vào một mã hàng và kiểm tra hàng có trong danh sách không. Nếu có thì xóa hàng vừa nhập. in danh sách sau khi xóa.

## Sử dụng ArrayList.

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 thoại 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.

## Sử dụng Arraylist

Yêu cầu: Viết chương trình cho nhập vào danh sách sinh viên (sử dụng arrayList) của một trường và xuất danh sách đó

theo từng lớp. Biết rằng trường có nhiều lớp và một lớp có nhiều sinh viên. Lớp gồm các thông tin: mã lớp, tên lớp, sĩ số. Sinh viên gồm các thông tin: mã sinh viên, họ tên, nơi sinh, mã lớp.

* 1. In danh sách sinh viên sau khi nhập
  2. Nhập vào mã sinh viên và tìm xem có sinh viên trong danh sách không. Nếu có hiển thị sinh viên
  3. Nhập vào mã sinh viên và tìm xem có sinh viên trong danh sách không. Nếu sửa thông tin cho sinh viên. hiển thị sinh viên sau khi sửa.
  4. Nhập vào mã sinh viên và tìm xem có sinh viên trong danh sách không. Nếu có thì xóa thông tin cho sinh viên. hiển thị sinh viên sau khi xóa.

## Sử dụng ArrayList:

Yêu cầu: Viết chương trình quản lý bán hàng được mô tả như bên dưới, yêu cầu dùng ArrayList :

Mỗi một danh mục sản phẩm sẽ có nhiều sản phẩm, thông tin danh mục sản phẩm gồm: Mã

danh mục, tên danh mục. Thông tin sản phẩm gồm: mã sản phẩm, tên sản phẩm, giá cả, xuất xứ.

1. Viết hàm cho phép thông tin danh mục sản phẩm và sản phẩm
2. Xuất danh sản phẩm thuộc một danh mục bất kỳ nhập vào từ bàn phím
3. Cho phép cập nhật thông tin sản phẩm
4. Xóa sản phẩm bất kỳ trong danh mục
5. Thống kê tổng giá trị của các mặt hàng
6. Liệt kê các sản phẩm có xuất xứ từ trung quốc

# Các luồng vào ra và thao tác với tệp dữ liệu

## 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 luuTextFile(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++)

{

ine+=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 fIn=new FileInputStream(filename);

Scanner sc=new Scanner(fIn);

while(sc.hasNextLine())

{

System.out.println(sc.nextLine());

}

fIn.close();

sc.close();

}

catch(IOException ex) { ex.printStackTrace(); }

}

## 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{…}

Viết một class MyFile có 2 phương thức:

* 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 fIn=new FileInputStream(fileName);

ObjectInputStream in=new ObjectInputStream(fIn);

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……**

}

MyFile.luuDoiTuong(dsDienTu, "data.txt"); //lưu lại danh sách

## Bài tập tổng hợp

**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.

**Hướng dẫn:**

**//tạo lớp thí sinh**

**public class ThiSinh implements Serializable{**

private String soBD;

private String hoTen;

private float diemToan;

private float diemLy;

private float diemHoa;

//insert code getter/setter các thuộc tính

**public String getSoBD()** {return soBD;}

//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 String getHoTen()** { return hoTen; }

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

if(hoTen.trim().equals("")) throw new Exception("Họ tên không được trống!");

this.hoTen = hoTen;

}

**public float getDiemToan**() { return diemToan;}

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

if(diemToan<0||diemToan>10) **throw new Exception**("Điểm toan không hợp lệ");

this.diemToan = diemToan;

}

**public float getDiemLy()** { return diemLy;}

**public void setDiemLy(float diemLy)** **throws** Exception{

if(diemLy<0||diemLy>10**) throw new Exception** ("Điểm ly không hợp lệ");

this.diemLy = diemLy;

}

**public float getDiemHoa**() { return diemHoa; }

public void setDiemHoa(float diemHoa) **throws** Exception{

if(diemHoa<0||diemHoa>10) **throw new Exception**("Điểm hoa không hợp lệ");

this.diemHoa = diemHoa;

}

//insert code constructor có 1 đối, không đối, tất cả đối

**public ThiSinh()** {

soBD="";hoTen="no-name";diemToan=diemLy=diemHoa=0f;

}

**public ThiSinh**(String soBD, String hoTen, float diemToan, float diemLy,float diemHoa) **throws** Exception {

setSoBD(soBD);

setHoTen(hoTen);

setDiemToan(diemToan);

setDiemLy(diemLy);

setDiemHoa(diemHoa);

}

**public ThiSinh(String soBD)** **throws** Exception { setSoBD(soBD); }

**publi c String toString()** {

String tsValue;

tsValue= "SBD:"+soBD+"\t Ho ten:"+hoTen+"\tdiem toan:"+diemToan+"\t điểm lý";

tsValue=tsValue +diemLy+"\tđiem hóa"+diemHoa;

return tsValue;

}

**//insert code hashCode &equals chọn soBD**

……………………………………………

}

**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()**

{ super();

this.msPT = "";

this.diachiPT = "";

this.luongTS = 0;

dsTS=new ArrayList<ThiSinh>();

}

**public PhongThi(String msPT, String diachiPT, int luongTS) {**

super();

this.msPT = msPT;

this.diachiPT = diachiPT;

this.luongTS = luongTS;

dsTS=new ArrayList<ThiSinh>();

}

//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

if(dsTS.contains(ts)) return false;

if(dsTS.size()+1>luongTS)//nếu đã đủ lượng thí sinh

return false;

return dsTS.add(ts);

}

//hàm xóa thí sinh khỏi phòng thi

**public boolean XoaThisinh(String soBD) throws Exception {**

ThiSinh ts=new ThiSinh(soBD);

if(!dsTS.contains(ts)) return false;

return dsTS.remove(ts);

**}**

//hàm sửa thông tin thí sinh trong phòng

**public boolean SuaThongtinTS(String soBD, ThiSinh newTS) throws Exception**

{

ThiSinh ts=new ThiSinh(soBD);

if(!dsTS.contains(ts)) return false;

dsTS.set(dsTS.indexOf(ts),newTS);

return true;

}

//hàm lấy thông tin thí sinh khi biết số báo danh

**public ThiSinh LayThongtinTS(String soBD) throws Exception {**

ThiSinh ts=new ThiSinh(soBD);

if(!dsTS.contains(ts)) return null;

ts=dsTS.get(dsTS.indexOf(ts));

return ts;

}

//lấy thông tin thí sinh khi biết vị trí

**public ThiSinh LayThongtinTS(int index)**

{

if(index<0||index>dsTS.size()) return null;

return dsTS.get(index);

}

//Hàm trả lại số thí sinh thật sự trong danh sách

**public int soThisinh()**

**{**

**return** dsTS.size();

**}**

//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
* Hiển 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. Luu phong thi vao file");

System.out.println("8. doc phong thi tu 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;

case 4: layTTTS\_soBD();break;

case 5: layTTTS\_chiso();break;

case 6: InDanhsachTS(); break;

case 7: luuFile();break;

case 8: docFile(); break;

case 9: System.out.println("BYE"); System.exit(1);

}

}while(true);

}

**static void luuFile()**

{

try {

db.LuuFile(fileName, phongthi);

} catch (Exception ex) {

System.out(“co loi luu file : “+ex.toString();

}

}

**static void docFile()**

{

try {

phongthi=(PhongThi) db.docFile(fileName);

} catch (Exception ex) {

System.out(“co loi doc file : “+ex.toString();

}

}

**static void ThemTS() {**

try {

String soBD,ten;

float toan,ly,hoa;

Scanner s=new Scanner(System.in);

System.out.println("nhap SBD, ten=");

soBD=s.nextLine();

ten=s.nextLine();

System.out.println("nhap diem toan, ly, hoa=");

toan=s.nextFloat();

ly=s.nextFloat();

hoa=s.nextFloat();

ThiSinh ts=new ThiSinh(soBD,ten,toan,ly,hoa);

//ts.nhap();

if(phongthi.themThiSinh(ts))

System.out.println("Thêm thành công");

else

System.out.println("Không thêm được.");

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

**static void XoaTS() {**

try {

Scanner input=new Scanner(System.in);

System.out.print("\tNhập số báo danh cần xóa:");

String soBD=input.nextLine();

input.nextLine();// dua con tro xuong dong tiep, tranh lay ky ty enter

if(!phongthi.XoaThisinh(soBD))

System.out.println("Không xóa được!");

else

System.out.println("Xóa thành công");

} catch (Exception ex) {

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

}

}

**static void SuaTTTS() {**

try {

Scanner input=new Scanner(System.in);

System.out.print("\tNhập số báo danh cần sửa:");

String soBD=input.nextLine();

String soBD1,ten;

float toan,ly,hoa;

Scanner s=new Scanner(System.in);

System.out.println("nhap SBD, ten=");

soBD1=s.nextLine();

ten=s.nextLine();

System.out.println("nhap diem toan, ly, hoa=");

toan=s.nextFloat();

ly=s.nextFloat();

hoa=s.nextFloat();

ThiSinh newTS=new ThiSinh(soBD1,ten,toan,ly,hoa);

if(phongthi.SuaThongtinTS(soBD, newTS))

System.out.println("Sửa thành công");

else

System.out.println("Không sửa được!");

} catch (Exception ex) {

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

}

}

**static void layTTTS\_chiso() {**

try {

Scanner input=new Scanner(System.in);

System.out.println("Nhập số thứ tự cần lấy thông tin:");

int stt=input.nextInt();

ThiSinh ts= phongthi.LayThongtinTS(stt);

if(ts==null)

System.out.println("không có");

else

System.out.println(ts);

} catch (Exception ex) {

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

}

}

**static void layTTTS\_soBD() {**

try {

Scanner input=new Scanner(System.in);

System.out.print("\tNhập số báo danh cần lấy thông tin:");

String soBD=input.nextLine();

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.s**oThisinh**(); i++) {

try {

ThiSinh ts=phongthi.LayThongtinTS(i);

System.out.println(ts);

} catch (Exception ex) {

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

}

}

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

}

}