# Nạp chồng hàm.

Yêu cầu:

1. Viết một phương thức có tên square, hàm có 1 tham số x kiểu số thực. Phương thứcnày tính toán và trả về bình phương của x.
2. Viết tiếp phương thức vẫn có tên square, hàm có 1 tham số x kiểu số nguyên. Phương thứcnày tính toán và trả về bình phương của x.
3. Vẫn viết hàm có tên như trên nhưng đối số truyền vào là 1 xâu trả lại xâu đó
4. Đưa phương thứctrên 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 thứcvừa xây dựng, hiển thị kết quả.

# 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

Overload hàm **average** với 2 tham số nguyên, và 1 tham số kiểu thực 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.

# Tính thể tích hình cầu

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: lấy Pi ta dùng Math.PI. in kết quả có định dạng ###,###.##

* Nó là cái gì: hình cầu
* Mô tả hình cầu: bán kính
* Hình cầu ấy làm gì: tính thể tích và diện tích bề mặt
* Vậy lớp hình cầu: thuộc tính: r,
* Phương thức: nhập, xuất, tính thể tích và diện tích bề mặt,

# Bài toán hình chữ nhật

Viết ứng dụng nhập chiều dài, chiểu rộng hình chữ nhật. Tính và in ra diện tích hình chữ nhật (rectangle) cùng thông tin của hình đó (in có định dạng 2 chữ số sau phần thập phân).

Gợi ý:

* 1. Khai báo thuộc tính dài rộng hình chữ nhật là riêng tư: private; phương thức tính chu vi và diện tích là công cộng.
  2. Xây dựng hàm main trong cùng lớp HCN
  3. Thay đổi cho hàm main trong 1 lớp khác, nhưng cùng gói với lớp hình chữ nhật ;
  4. Copy lớp chứa hàm main sang gói khác với gói chứa lớp HCN. Thay đổi lại thuộc tính truy cập cho các phương thức tính chu vi và diện tích thành private, protected, hay không khai báo và quan sát kết quả chương trình thực hiện

**Mở rộng bài toán: nhập vào 1 dãy hình chữ nhật. in ra thông tin dãy hình chữ nhật sau khi nhập**

* 1. Dùng mảng:
* Khai báo mảng: kiểu\_ dl biến\_mảng[];
* Nhập kích thước mảng n
* Xin cấp phát mảng có kích thước n phần tử: biến\_mảng= new kiểuDL[n];
* Sử dụng mảng như thông thường
  1. Dùng danh sách không giới hạn phần tử
* Khai báo danh sách: ArrayList<tên kiểu dl> tênDS=new ArrayList<tên kiểu dl>();
* Thêm phần tử vào danh sách: tênDS.add(tên biến),
* Kích thước danh sách: tênDS.size()

# Bài toán phân số

Cho phân có có tử số, mẫu số là các số thực. Viết chương trình thực hiện

* Nhập phân số
* In phân số
* In tổng 2 phân số, hiệu 2 phân số

Mở rộng bài toán: nhập vào 1 dãy phân số, in ra tổng các phân số, hiệu các phân số.

# 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:

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

Hãy thiết kế và cài đặt lớp Vehicle với các thuộc tính (attributes) và các phương thức (methods) phù hợp. Lớp phải có các hàm tạo (constructor) và phải bảo đảm tính đóng gói dữ liệu(encapsulation)

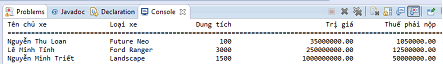
Xây dựng class chứa hàm main. Hàm main in ra menu lựa chọn các công việc:

1. Nhập thông tin và tạo các đối tượng xe1, xe2, xe3

2. Xuất bảng kê khai tiền thuế trước bạ của các xe.

3. Thoát.

Mẫu thiết kế xuất của chương trình:



# 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. chú ý ngày tháng năm nhập sai có báo lỗi.

# Xây dựng lớp và hàm thành phần.

Yêu cầu: (i) Xây dựng lớp TamGiac 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

**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 với 2 cạnh kia, 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**.

**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 danh sách tam giác, in danh sách 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)

+ getMaNV() : String

+ setMaNV(ma : String) : void

+ getSoSP() : int

+ setSoSP(sp : int) : void   
+ coVuotChuan() : boolean   
+ getTongKet() : String   
+ getLuong(): double   
+ XuatTieuDe() : static void   
+ toString() : String

Trong các hàm 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.
* 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 và cho biết nhân viên có vượt chuẩn không.

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

d). In danh sách nhân viên vượt chuẩn.

# Quản lý tài sản

Cho mẫu phiếu kiểm kê tài sản theo mẫu in ra. Thực hiện các yêu cầu sau:

1. Xây dựng các lớp biểu diễn thông tin cho bài toán sử dụng phương pháp hướng đối tượng. Yêu cầu các thuộc tính đều đặt phạm vi truy cập riêng tư
2. Xây dựng lớp chưa hàm main thực hiện các việc sau:
   1. Tạo một phiếu kiểm kê**:** cho phép nhập các thông tin chung của phiếu (mã phiếu, ngày kiểm kê);
   2. Nhập thông tin về nhân viên kiểm kê;
   3. Nhập thông tin về phòng ban đang kiểm kê;
   4. Nhập thông tin về các tài sản kiểm kê.
   5. Hiển thị Thông tin phiếu kiểm kê theo định dạng mẫu (không kẻ bảng cũng được)

Mẫu phiếu:

|  |  |  |
| --- | --- | --- |
| PHIẾU KIỂM KÊ TÀI SẢN  Mã phiếu: *PH01*. Ngày kiểm kê: *1/1/2007*  Nhân viên kiểm kê: *Kiều Thị Thanh*  Chức vụ: *Kế toán viên*  Kiểm kê tại phòng: *Tổ chức hành chính* Mã phòng:  *PTC* Trưởng phòng: *Hoàng Bích Hảo* | | |
| **Tên tài sản** | **Số lượng** | **Tình trạng** |
| Máy vi tính | 5 | Tốt |
| Máy vi tính | 3 | Hết khấu hao - hỏng |
| Bàn làm việc | 6 | Tốt |

Số tài sản đã kiểm kê*: 3. Tổng số lượng: 14*

# 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 thứcsoSanhNV(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().