Bài 1: TƯ DUY LẬP TRÌNH – Bài tập

1. Giải phương trình bậc nhất Ax + B = 0

* Bước 1: Các danh từ: phương trình bậc nhất, số A, số B, biến x
* Bước 2:
  + Phương trình bậc nhất: không có miền giá trị → đối tượng (object)
  + Số A: miền giá trị là R → thuộc tính (attribute)
  + Số B: như trên
  + Biến x: như trên
* Bước 3: Các đối tượn g có mặc trong bài toán:
  + Phương trình bậc nhất
    - Thuộc tính: số A, số B, biến x, câu trả lời.
    - Phương thức:
      * Giải phương trình (động từ được tìm thấy trong bài toán)
      * Nhập thông tin để nhập giá trị đầu cho 2 thuộc tính số A, số B
      * In câu trả lời

Đối tượng sẽ thực hiện những việc như sau thì bài toán được giải quyết:

Phương trình bậc nhất → Nhập thông tin (giá trị số A, giá trị số B): để gán giá trị cho 2 thuộc tính số A và số B

Phương trình bậc nhất → Giải phương trình: biện luận nghiệm của phương trình theo hai thuộc tính số A và số B, sử dụng biến x và câu trả lời để lưu kết quả

Phương trình bậc nhất → In câu trả lời: sử dụng Câu trả lời để in kết quả ra màn hình

1. Viết chương trình nhập vào tháng và năm sau đó tính và in ra số ngày của tháng thuộc năm đã nhập

* Bước 1: Các danh từ: tháng, năm, số ngày của tháng, số ngày
* Bước 2:
  + Số ngày trong tháng: không có miền giá trị xác định (không phải số nguyên nào cũng bằng số ngày trong tháng)
  + Tháng: có miền giá trị là các tháng trong năm
  + Năm: có miền giá trị là N
  + Số ngày: như trên
* Bước 3:
  + Số ngày trong tháng
    - Thuộc tính: Tháng, Năm, số ngày
    - Phương thức:
      * Nhập giá trị tháng và năm
      * Tính số ngày có trong tháng của năm đó
      * In câu trả lời

Đối tượng sẽ thực hiện những việc như sau thì bài toán được giải quyết:

Số ngày trong tháng → Nhập giá trị cho thuộc tính tháng và năm

Số ngày trong tháng → Tính số ngày có trong tháng của năm vừa nhập, sử dụng thuộc tính số ngày để lưu kết quả

Số ngày trong tháng → In kết quả: sử dụng số ngày để in kết quả

Bài 3: Hãy in thông tin chu vi, diện tích, độ dài cạnh, chiều dài, chiều rộng của hình vuông và hình chữ nhật nếu biết thông tin 4 đỉnh của hình vuông và hình chữ nhật đó

* Bước 1: các danh từ: chu vi, diện tích, độ dài cạnh, chiều dài, chiều rộng, hình vuông, hình chữ nhật, thông tin 4 đỉnh
* Bước 2:
  + Chu vi, diện tích, độ dài cạnh, chiều dài, chiều rộng, thông tin 4 đỉnh: miền giá trị thuộc R
  + Hình vuông, hình chữ nhật: không xác định được miền giá trị
* Bước 3:
  + Đỉnh:
    - Thuộc tính: x, y thuộc R
    - Phương thức:
      * Nhập thông tin: nhập giá trị cho 2 thuộc tính x và y
      * In giá trị x, y
  + Hình vuông:
    - Thuộc tính: chu vi, diện tích, độ dài cạnh, đỉnh
    - Phương thức:
      * Nhập thông tin: nhập giá trị cho 4 thuộc tính đỉnh A, B, C, D
      * Tính độ dài cạnh: sử dụng giá trị của 4 đỉnh để tính, lưu giá trị cho thuộc tính độ dài cạnh.
      * Tính chu vi: sử dụng thuộc tính độ dài cạnh để tính, lưu giá trị cho thuộc tính chu vi
      * Tính diện tích: sử dụng thuộc tính độ dài cạnh để tính, lưu giá trị cho thuộc tính diện tích
      * In kết quả:
        + In thông tin 4 đỉnh A, B, C, D. Cho từng đỉnh thực thi hành vi in thông tin của mình
        + In thông tin độ dài cạnh, chu vi, diện tích của hình vuông
  + Hình chữ nhật:
    - Thuộc tính: chu vi, diện tích, chiều dài, chiều rộng, đỉnh
    - Phương thức:
      * Nhập thông tin: nhập giá trị cho 4 thuộc tính đỉnh A, B, C, D
      * Tính chiều dài, chiều rộng: sử dụng giá trị của 4 đỉnh để tính, lưu giá trị cho thuộc tính chiều dài, chiều rộng.
      * Tính chu vi: sử dụng thuộc tính chiều dài và chiều rộng để tính, lưu giá trị cho thuộc tính chu vi
      * Tính diện tích: sử dụng thuộc tính chiều dài và chiều rộng để tính, lưu giá trị cho thuộc tính diện tích
      * In kết quả:
        + In thông tin 4 đỉnh A, B, C, D. Cho từng đỉnh thực thi hành vi in thông tin của mình
        + In thông tin chiều dài, chiều rộng, chu vi, diện tích của hình chữ nhật

Bài 2: TÍNH TRỪU TƯỢNG HOÁ – Bài tập

Yêu cầu:

* Áp dụng tư duy hướng đối tượng để phân tích bài toán
* Áp dụng trừu tượng hoá để tối ưu các đối tượng
* Xây dựng mối quan hệ giữa các đối tượng và tối ưu các đối tượng
* Xây dựng các đối tượng như các VD ở cuối bài 2

Bài 1: Giải phương trình bậc 2: ax2 + bx + c = 0

Class PTBacHai {

// Khai báo các thuộc tính

Double soA, soB, soC, x;

String cauTraLoi;

//Khai báo và khởi tạo các phương thức

Void nhap3SoABC(double gTriSoA, double gTriSoB, double gTriSoC) {

//Nội dung phương thức

//Sử dụng gTriSoA, gTriSoB, gTriSoC để gán giá trị cho các thuộc tính soA, soB, soC

}

Void timNghiem() {

//Nội dung phương thức

//Sử dụng soA, soB, soC, cauTraLoi để thực hiện biện luận nghiệm

}

Void inKetQua() {

//Sử dụng cauTraLoi để in kết quả

}

}

Bài 2: Một số được gọi là số thuận nghịch độc nếu ta đọc từ trái sang phải hay từ phải sang trái số đó ta vẫn nhận được một số giống nhau. Hãy liệt kê tất cả các số thuận nghịch độc có sáu chữ số (Ví dụ số 558855)

Class thuanNghichDoc {

//Khai báo các thuộc tính

Int soNguyen

//Khai báo và khởi tạo các phương thức

Void nhapSoNguyen(int gTriSoNguyen) {

//Nội dung phương thức

//Sử dụng gTriSoNguyen để gán giá trị cho thuộc tính soNguyen

}

Boolean kTraSTND() {

//Nội dung phương thức

//Sử dụng soNguyen để kiểm tra xem nó có phải số thuận nghịch độc hay không

//Xây dựng phương thức theo cơ chế trả về kết quả

Return false; //return giá trị mặc định vì chưa xây dựng nội dung

}

}

Class DSSoThuanNghich() {

//Khai báo các thuộc tính

Int chanDuoi, chanTren;

thuanNghichDoc soTND;

String cauTraLoi;

Void nhapChan(int gTriDuoi, int gTriTren) {

//Nội dung phương thức

//Sử dụng gTriDuoi, gTriTren để gán giá trị cho chanDuoi, chanTren

}

Void lietKeDSSoTND() {

//Nội dung phương thức

//Duyệt các số nguyên từ chanDuoi đến chanTren

//Với mỗi số nguyên i thì nhập đầu vào cho soTND và tiến hành gọi phương thức kTraSTND

//Nếu kết quả trả về là true thì sẽ gán giá trị vào câu trả lời.

}

Void inDSSTND() {

//Nội dung phương thức

//Sử dụng câu trả lười để in danh sách các số thuận nghịch độc

}

}

Bài 3: Một ứng dụng đọc truyện cười khi người dùng sử dụng trên máy tính/điện thoại sẽ hiển thị các chủ đề của truyện.

Một chủ đề truyện bao gồm các thông tin về mã chủ đề, tên chủ đề, các truyện cười trong chủ đề đó, mỗi chủ đề có 3 truyện cười, người dùng có thể tìm kiếm truyện theo tên hoặc theo mã id truyện, sau khi tìm kiếm thành công, truyện tìm thấy sẽ được trả về lại người dùng, ngoài ra cũng có thể:

* Thêm mới truyện vào danh sách truyện trong chủ đề
* Xoá truyện khỏi chủ đề khi biết mã id của truyện
* Đọc truyện khi biết mã id của truyện

Khi người dùng đọc truyện thì nội dung truyện đó sẽ được hiển thị lên màn hình cho người dùng xem.

Một truyện cười gồm những thông tin mã id truyện, tên truyện, tác giả, ngày xuất bản, nội dung truyện. Người dùng quản lý 3 chủ đề, có hành vi:

* Tạo chủ đề mới
* Xoá chủ đề, tìm kiếm chủ đề khi biết mã chủ đề.

Các đối tượng có mặt trong bài toán: Truyện cười, Chủ đề, Quản lý

* Truyện cười:
  + Thuộc tính: Tên truyện, mã ID, tác giả, ngày xuất bản, nội dung
  + Phương thức:
    - Nhập thông tin
    - In thông tin
* Chủ đề:
  + Thuộc tính: Tên chủ đề, mã chủ đề, truyện cười
  + Phương thức:
    - Nhập thông tin
    - In thông tin
    - Thêm truyện vào chủ đề
    - Xoá truyện khỏi chủ đề
    - Tìm truyện trong chủ đề
    - Đọc truyện
* Quản lý:
  + Thuộc tính: danh sách <Chủ đề>, danh sách <Truyện cười>
  + Phương thức:
    - Thêm truyện mới
    - Thêm chủ đề mới
    - Tìm kiếm chủ đề
    - Tìm kiếm truyện
    - Xoá truyện
    - Xoá chủ đề
    - Hiển thị danh sách truyện
    - Hiển thị danh sách chủ đề

Class TruyenCuoi {

//Khai báo các thuộc tính

String maID, tenTruyen, tacGia, ngayXB, noiDung;

//Khai báo và khởi tạo các phương thức

Void nhapTruyen(string ma, string ten, string tGia, string ngXB, strign nDung) {

//Nội dung phương thức

//Sử dụng ma, ten, tGia, ngXB, nDung để gán giá trị cho maID, tenTruyen, tacGia, ngayXB, noiDung

}

Void docTruyen() {

//Nội dung phương thức

//Sử dụng thuộc tính noiDung để đọc truyện

}

Void inTruyen(){

//Nội dung phương thức

//In thông tin truyện dựa trên giá trị các thuộc tính maId, tenTruyen, tacGia, ngayXB, noiDung

}

}

Class ChuDe {

//Khai báo các thuộc tính

String maCD, tenCD;

TruyenCuoi trCuoi;

//Khai báo và khởi tạo các phương thức

Void nhapCD(string ma, string ten) {

//Nội dung phương thức

//Sử dụng ma, ten, để gán giá trị cho các thuộc tính maCD, tenCD

}

Void themTruyen() {

//Nội dung phương thức

//Thêm truyện vào chủ đề

//Gán giá trị vào trCuoi

}

Void xoaTruyen() {

//Nội dung phương thức

//Xoá truyện khỏi chủ đề

}

Void inChuDe() {

//Nội dung phương thức

//In thông tin chủ đề

{

}

Class QuanLy() {

//Khai báo các thuộc tính

TruyenCuoi truyen;

ChuDe cDe;

//Khai báo và khởi tạo phương thức

Void themTruyen() {

//Nội dung phương thức

//Sử dụng phương thức nhapTruyen của thuộc tính truyen

}

Void themCD() {

//Nội dung phương thức

//Sử dụng phương thức nhapCD của thuộc tính cDe

}

Void timKiem() {

//Nội dung phương thức

//Tìm kiếm truyện hoặc chủ đề theo mã

//Sử dụng phương thức inChuDe và inTruyen để in thông tin chủ đề/truyện vừa tìm được

}

Void xoa() {

//Nội dung phương thức

//Xoá truyện hoặc chủ đề theo mã

}

Void inDSTruyen() {

//Nội dung phương thức

}

Void inDSChuDe() {

//Nội dung phương thức

}

}

Bài 3: TÍNH KẾ THỪA – Bài tập

Ví dụ 1:

Class ConNguoi() {

String hoTen, gioiTinh, diaChi;

Int tuoi;

Void nhapTT(string ht, string gt, string dc, int t) {

//Nhập thông tin cho các thuộc tính hoTen, gioiTinh, diaChi, tuoi

}

Void anUong() {

//Nội dung phương thức ăn uống

}

Void diChuyen() {

//Nội dung phương thức di chuyển

}

Void inTT() {

//In thông tin con người gồm: hoTen, gioiTinh, diaChi, tuoi

}

}

**Class HocSinh extends ConNguoi** {

String maHS, lop;

Int diemToan, diemVan, diemAnh;

**Void nhapTT(string ht, string gt, string dc, int tuoi, string ma, string lop, string diemToan, string diemVan, string diemAnh) {**

**//Gọi đến phương thức nhapTT của đối tượng cha**

**nhapTT(ht, gt, dc, tuoi);**

**//Nhập thông tin cho các thuộc tính ma, lop, diemToan, diemVan, diemAnh**

**}**

Void hocBai() {

//Nội dung phương thức học bài

}

Void tinhDTB() {

//Nội dung phương thức tính điểm t/bình

Return -1; //Trả về giá trị mặc định

}

**@Override**

**Void inTT() {**

**//Gọi đến phương thức inTT của đối tượng cha**

**Supper.intt();**

**//In thêm thông tin cho các thuộc tính maHS, lop, diemToan, diemVan, diemAnh**

**}**

}

**Class NhanVien extends ConNguoi** {

String maNV, chucVu;

Int luong;

**Void nhapTT(string ht, string gt, string dc, int tuoi,string ma, string chucVu, int luong) {**

**//Gọi đến phương thức nhapTT của đối tượng cha**

**nhapTT(ht, gt, dc, tuoi);**

**//Nhập thông tin cho các thuộc tính ma, chucVu, luong**

**}**

Void lamViec() {

//Nội dung phương thức làm việc

}

Void vietBaoCao() {

//Nội dung phương thức viết báo cáo

}

**@Override**

**Void inTT() {**

**//gọi đến phương thức inTT của đối tượng cha**

**Supper.inTT();**

**//In thêm thông tin cho các thuộc tính maNV, chucVu, luong**

**}**

}

Class QuanLy() {

//Khai báo danh sách nhân viên

NhanVien[] dsNhanVien;

//Khai báo danh sách học sinh

HocSinh[] dsHocSinh;

Void themNhanVien(NhanVien nv) {

//thêm nhân viên vào danh sách

}

Void thmeHocSinh(HocSinh hs) {

//thêm học sinh vào danh sách

}

NhanVien timKiemNhanVien(string maNV) {

//Tìm kiếm và trả về nhân viên theo mã nhân viên

Return null; //Trả về giá trị mặc định

}

HocSinh timKiemHocSinh(string maHS) {

//Tìm kiếm và trả về học sinh theo mã học sinh

Return null; //Trả về giá trị mặc định

}

Void xoaNhanVien(string maNV) {

//Xoá nhân viên có mã nhân viên là maNV

}

Void xoaHocSinh(string maHS) {

//Xoá học sinh có mã học sinh là maHS

}

Void hienThiDsHS() {

//Hiển thị danh sách thông tin từng học sinh

}

Void hienThiDsNV() {

//Hiển thị danh sách thông tin từng nhân viên

}

}

Ví dụ 2:

**Abstract Class ConNguoi()** {

String hoTen, gioiTinh, diaChi;

Int tuoi;

Void nhapTT(string ht, string gt, string dc, int t) {

//Nhập thông tin cho các thuộc tính hoTen, gioiTinh, diaChi, tuoi

}

**Abstract void anUong(); //Định nghĩa anUong là phương thức trừu tượng**

**Abstract void diChuyen() //Định nghĩa diChuyen là phương thức trừu tượng**

Void inTT() {

//In thông tin con người gồm: hoTen, gioiTinh, diaChi, tuoi

}

}

Class HocSinh extends ConNguoi {

String maHS, lop;

Int diemToan, diemVan, diemAnh;

Void nhapTT(string ht, string gt, string dc, int tuoi, string ma, string lop, string diemToan, string diemVan, string diemAnh) {

//Gọi đến phương thức nhapTT của đối tượng cha

nhapTT(ht, gt, dc, tuoi);

//Nhập thông tin cho các thuộc tính ma, lop, diemToan, diemVan, diemAnh

}

Void hocBai() {

//Nội dung phương thức học bài

}

Void tinhDTB() {

//Nội dung phương thức tính điểm t/bình

Return -1; //Trả về giá trị mặc định

}

**@Override**

**Void anUong() {**

**//Xây dựng hành vi ăn uống của Học Sinh**

**//ghi đè hoàn toàn nên không có super**

**}**

**@Override**

**Void diChuyen() {**

**//Xây dựng hành vi di chuyển của Học Sinh**

**//ghi đè hoàn toàn nên không có super**

**}**

@Override

Void inTT() {

//Gọi đến phương thức inTT của đối tượng cha

Supper.intt();

//In thêm thông tin cho các thuộc tính maHS, lop, diemToan, diemVan, diemAnh

}

}

Class NhanVien extends ConNguoi {

String maNV, chucVu;

Int luong;

Void nhapTT(string ht, string gt, string dc, int tuoi,string ma, string chucVu, int luong) {

//Gọi đến phương thức nhapTT của đối tượng cha

nhapTT(ht, gt, dc, tuoi);

//Nhập thông tin cho các thuộc tính ma, chucVu, luong

}

Void lamViec() {

//Nội dung phương thức làm việc

}

Void vietBaoCao() {

//Nội dung phương thức viết báo cáo

}

**@Override**

**Void anUong() {**

**//Xây dựng hành vi ăn uống của Học Sinh**

**//ghi đè hoàn toàn nên không có super**

**}**

**@Override**

**Void diChuyen() {**

**//Xây dựng hành vi di chuyển của Học Sinh**

**//ghi đè hoàn toàn nên không có super**

**}**

@Override

Void inTT() {

//gọi đến phương thức inTT của đối tượng cha

Supper.inTT();

//In thêm thông tin cho các thuộc tính maNV, chucVu, luong

}

}

Class QuanLy() {

//Khai báo danh sách nhân viên

NhanVien[] dsNhanVien;

//Khai báo danh sách học sinh

HocSinh[] dsHocSinh;

Void themNhanVien(NhanVien nv) {

//thêm nhân viên vào danh sách

}

Void thmeHocSinh(HocSinh hs) {

//thêm học sinh vào danh sách

}

NhanVien timKiemNhanVien(string maNV) {

//Tìm kiếm và trả về nhân viên theo mã nhân viên

Return null; //Trả về giá trị mặc định

}

HocSinh timKiemHocSinh(string maHS) {

//Tìm kiếm và trả về học sinh theo mã học sinh

Return null; //Trả về giá trị mặc định

}

Void xoaNhanVien(string maNV) {

//Xoá nhân viên có mã nhân viên là maNV

}

Void xoaHocSinh(string maHS) {

//Xoá học sinh có mã học sinh là maHS

}

Void hienThiDsHS() {

//Hiển thị danh sách thông tin từng học sinh

}

Void hienThiDsNV() {

//Hiển thị danh sách thông tin từng nhân viên

}

}

Ví dụ 3: Hãy in thông tin chu vi, diện tích, độ dài cạnh, chiều dài, chiều rộng của hình vuông và hình chữ nhật nếu biết thông tin 4 đỉnh của hình vuông, hình chữ nhật đó.

Các đối tượng có mặt trong bài toán: Hình Vuông, Hình Chữ Nhật, Tứ Giác, Đỉnh. Trong đó:

* Tứ giác là đối tượng trừu tượng
* Tính chu vi, tính diện tích là 2 hành vi trừu tượng

Class Dinh {

Double x, y;

Void nhapTT(double gTriX, double gTriY) {

//Nhập vào thông tin 2 thuộc tính x và y

}

}

Abstract class TuGiac {

//Khai báo các thuộc tính

Dinh dinhA, dinhB, dinhC, dinhD;

Double chuVi, dienTich;

Void nhapTT(Dinh a, Dinh b, Dinh c, Dinh d) {

//Nhập thông tin cho 4 thuộc tính dinhA, dinhB, dinhC, dinhD

}

//Định nghĩa hành vi trừu tượng: tính chu vi

Abstract void tinhChuVi();

//Định nghĩa hành vi trừu tượng: tính diện tích

Abstract void tinhDienTich();

Void inTT() {

//in thông tin giá trị các thuộc tính

//4 đỉnh A, B, C, D

//giá trị chu vi, diện tích

}

}

Class HinhVuong extends TuGiac {

//Khai báo thuộc tính đối tượng Hình Vuông

Double doDaiCanh;

//Không ghi đề phương thức nhập thông tin của Tứ Giác vì nó đã đầy đủ để có thể sử dụng luôn

Void tinhDoDaiCanh() {

//Sử dụng 2 thuộc tính đỉnh bất kỳ để tính độ dài cạnh

}

@Override

Void tinhChuVi() {

//ghi đè hoàn toàn phương thức của đối tượng cha

//xây dựng nội dung phương thức tính chu vi = 4 lần độ dài cạnh

}

@Override

Void tinhDienTich() {

//ghi đè hoàn toàn phương thức của đối tượng cha

//xây dựng nội dung phương thức diện tích = bình phương độ dài cạnh

}

@Override

Void inTT() {

Super.inTT(); //ghi đè không hoàn toàn phương thức của đối tượng cha

//in thêm thông tin độ dài cạnh

}

}

Abstract class HinhChuNhat extends TuGiac {

//Khai báo thuộc tính đối tượng Hình Chữ Nhật

Double chieuDai, chieuRong;

//Không ghi đè phương thức nhập thông tin của Tứ Giác vì nó đã đầy đủ có thể sử dụng luôn

Void tinhCDCR() {

//Sử dụng 4 thuộc tính đỉnh để tính chiều dài, chiều rộng

}

@Override

Void tinhChuVi() {

//ghi đè hoàn toàn phương thức của đối tượng cha

//xây dựng nội dung phương thức tính chu vi

}

@Override

Void tinhDienTich() {

//Ghi đè hoàn toàn phương thức của đối tượng cha

//Xây dựng nội dung phương thức tính diện tích

}

@Override

Void inTT() {

Super.inTT(); //ghi đè không hoàn toàn phương thức của đối tượng cha

//in thêm thông tin chiều dài, chiều rộng

}

}

Bài tập:

* Các đối tượng có mặt trong bài toán: Ghi chú tường thuật, ghi chú nhắc nhở, quản lý, báo thức, thời gian
  + Thời gian:
    - Thuộc tính: ngày, tháng, năm, giờ, phút, giây
    - Phương thức:
      * Nhập thông tin: nhập thông tin cho các thuộc tính ngày, tháng, năm, giờ, phút, giây
      * In thông tin: in thông tin các thuộc tính ngày, tháng, năm, giờ, phút, giây
  + Báo Thức
    - Thuộc tính: thời gian, số lần, âm báo
    - Phương thức:
      * Nhập thông tin: nhập thông tin các thuộc tính thời gian, số lần, âm báo
      * In thông tin: in thông tin các thuộc tính thời gian, số lần, âm báo
      * Rung chuông: rung chuông với âm có giá trị của thuộc tính âm báo, với số lần và tại thời gian đã được nhập
  + Ghi Chú Tường Thuật:
    - Thuộc tính: mã ghi chú, tiêu đề, nội dung, thời gian, địa điểm, người liên quan, kết quả, đánh giá
    - Phương thức:
      * Nhập thông tin: nhập giá trị cho các thuộc tính mã ghi chú, tiêu đề, nội dung, thời gian, địa điểm, người liên quan, kết quả, đánh giá
      * In thông tin: in thông tin mã ghi chú, tiêu đề, nội dung, thời gian, địa điểm, người liên quan, kết quả, đánh giá
  + Ghi Chú Nhắc Nhở:
    - Thuộc tính: mã ghi chú, tiêu đề, nội dung, thời gian, địa điểm, báo thức, xác nhận tham gia
    - Phương thức:
      * Nhập thông tin: nhập thông tin cho các thuộc tính mã ghi chú, tiêu đề nội dung, thời gian, địa điểm, lời xác nhận. Cho thuộc tính báo thức thực hiện hành vi nhập thông tin của mình.
      * Nhắc nhở: Cho thuộc tính báo thức thực hiện hành vi rung chuông của mình, sau đó cho người dùng xác nhận tham gia hay không tham gia vào sự kiện, điền giá trị cho thuộc tính lời xác nhận
      * Liệt kê sự kiện đã lỡ: liệt kê các sự kiện đã quá thời gian nhưng chưa được xác nhận tham gia hay không
      * In thông tin: in thông tin cho các thuộc tính nhập thông tin cho các thuộc tính mã ghi chú, tiêu đề nội dung, thời gian, địa điểm, lời xác nhận. Cho thuộc tính báo thức thực hiện hành vi nhập thông tin của mình.
  + Quản lý:
    - Thuộc tính: danh sách ghi chú tường thuật, danh sách ghi chú nhắc nhở
    - Phương thức:
      * Tạo ghi chú
      * Xoá ghi chú
      * Tìm kiếm ghi chú
      * Xem nội dung
* Tìm kiếm mối quan hệ kế thừa:

|  |  |  |
| --- | --- | --- |
| Đối tượng | Thuộc tính chung | Phương thức chung |
| Ghi chú tường thuật  Ghi chú sự kiện | Mã ghi chú  Tiêu đề  Nội dung  Thời gian  Địa điểm | Nhập thông tin  In thông tin |

Hình thành đối tượng cha Ghi Chú chứa các đặc điểm chung của Ghi chú tường thuật và Ghi chú sự kiện. Vậy, bài toán có 6 đối tượng: thời gian, báo thức, ghi chú, ghi chú tường thuật, ghi chú sự kiện, quản lý

class ThoiGian {

    int ngay, thang, nam, gio, phut, giay;

    void nhapTT(int day, int month, int year, int hour, int minute, int sec) {

*//Nhập thông tin cho các thuộc tính ngày, tháng, năm, giờ, phút, giây*

    }

    void inTT() {

*//In thông tin các thuộc tính ngày, tháng, năm, giờ, phút, giây*

    }

}

class BaoThuc {

    ThoiGian tGian;

    int soLan;

    char amBao;

    void nhapTT(ThoiGian thoiGian, int lan, char am) {

*//Nhập thông tin cho các thuộc tính tGian, soLan, amBao*

    }

    void inTT() {

*//In thông tin cho các thuộc tính tGian, soLan, amBao*

    }

    void rungChuong() {

*//rung chuông với âm có giá trị của thuộc tính âm báo, với số lần và tại thời gian đã được nhập*

    }

}

class GhiChu {

    char maGhiChu, tieuDe, noiDung, diaDiem;

    void nhapTT(char maGhiChu, char tieuDe, char noiDung, char diaDiem) {

*// Nhập thông tin cho các thuộc tính maGhiChu, tieuDe, noiDung, diaDiem*

    }

    void inTT() {

*// In thông tin cho các thuộc tính maGhiChu, tieuDe, noiDung, diaDiem*

    }

}

class GhiChuTuongThuat extends GhiChu {

    char ngLienQuan, kQua, danhGia;

    void nhapTT(char maGhiChu, char tieuDe, char noiDung, char diaDiem, char ngLienQuan, char kQua, char danhGia) {

        nhapTT(maGhiChu, tieuDe, noiDung, diaDiem);

*// Nhập thông tin cho các thuộc tính còn lại*

    }

    @Override

    void inTT() {

**super**.inTT();

*//in thêm thông tin các thuộc tính còn lại*

    }

}

class GhiChuSuKien extends GhiChu {

    BaoThuc bThuc;

    boolean xacNhan;

    void nhapTT(char maGhiChu, char tieuDe, char noiDung, char diaDiem, BaoThuc baoThuc, boolean xacNhan) {

        nhapTT(maGhiChu, tieuDe, noiDung, diaDiem);

*//nhập thông tin cho các thuộc tính còn lại*

    }

    void nhacNho() {

*//sử dụng phương thức rung chuông của thuộc tính bThuc;*

*//Xác nhận có hay không tham gia sự kiện*

    }

    void lietKeSKLo() {

*// liệt kê các sự kiện chưa được xác nhận*

    }

    @Override

    void inTT() {

**super**.inTT();

*// in thêm thông tin các thuộc tính còn lại*

    }

}

class QuanLy {

    GhiChuTuongThuat tuongThuat;

    GhiChuSuKien suKien;

    void taoGhiChu() {

*//tạo một ghi chú mới*

    }

    void xoaGhiChu(char maGhiChu) {

*// xoá một ghi chú khi biết mã ghi chú*

    }

    void timGhiChu(char maGhiChu, char tieuDe, char noiDung) {

*// tìm ghi chú khi biết 1 trong 3 thuộc tính trên*

    }

    void xemGhiChu(char maGhiChu) {

*// xem nội dung ghi chú khi biết mã ghi chú*

    }

}