import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.Scanner;

abstract class PhongHoc {

    protected String maPhong;

    protected String dayNha;

    protected double dienTich;

    protected int soBongDen;

    abstract boolean xetDatChuan();

    public String getMaPhong() {

        return maPhong;

    }

    public void setMaPhong(String maPhong) throws Exception {

        if(maPhong.length() > 0)

            this.maPhong = maPhong;

        else

            throw new Exception("Khong duoc de trong!");

    }

    public String getDayNha() {

        return dayNha;

    }

    public void setDayNha(String dayNha) throws Exception {

        if(dayNha.length() > 0)

            this.dayNha = dayNha;

        else

            throw new Exception("Khong duoc de trong!");

    }

    public double getDienTich() {

        return dienTich;

    }

    public void setDienTich(double dienTich) throws Exception {

        if(dienTich >= 50)

            this.dienTich = dienTich;

        else

        throw new Exception("Khong duoc de trong!");

    }

    public int getSoBongDen() {

        return soBongDen;

    }

    public void setSoBongDen(int soBongDen) throws Exception {

        if(soBongDen > 0)

            this.soBongDen = soBongDen;

        else

        throw new Exception("Khong duoc de trong!");

    }

    @Override

    public int hashCode() {

        final int prime = 31;

        int result = 1;

        result = prime \* result + ((maPhong == null) ? 0 : maPhong.hashCode());

        return result;

    }

    @Override

    public boolean equals(Object obj) {

        if (this == obj)

            return true;

        if (obj == null)

            return false;

        if (getClass() != obj.getClass())

            return false;

        PhongHoc other = (PhongHoc) obj;

        if (maPhong == null) {

            if (other.maPhong != null)

                return false;

        } else if (!maPhong.equals(other.maPhong))

            return false;

        return true;

    }

    public PhongHoc(String maPhong, String dayNha, double dienTich, int soBongDen) {

        super();

        this.maPhong = maPhong;

        this.dayNha = dayNha;

        this.dienTich = dienTich;

        this.soBongDen = soBongDen;

    }

    public PhongHoc() {

        super();

    }

    public boolean duAnhSang()

    {

        if(dienTich/soBongDen > 10)

            return false;

        return true;

    }

    @Override

    public String toString() {

        return String.format("|"+"%-15s"+"|"+"%-10s"+"|"+"%15.0f"+"|"+"%15d"+"|", maPhong, dayNha, dienTich, soBongDen);

    }

}

class LyThuyet extends PhongHoc {

    private boolean mayChieu;

    @Override

    boolean xetDatChuan() {

        if(duAnhSang() && mayChieu == true)

            return true;

        return false;

    }

    public boolean isMayChieu() {

        return mayChieu;

    }

    public void setMayChieu(boolean mayChieu) {

        this.mayChieu = mayChieu;

    }

    public LyThuyet(String maPhong, String dayNha, double dienTich, int soBongDen, boolean mayChieu) {

        super(maPhong, dayNha, dienTich, soBongDen);

        this.mayChieu = mayChieu;

    }

    public LyThuyet() {

        super();

    }

    public LyThuyet(String maPhong, String dayNha, double dienTich, int soBongDen) {

        super(maPhong, dayNha, dienTich, soBongDen);

    }

    @Override

    public String toString() {

        String danhGia, mayChieu;

        if(xetDatChuan())

            danhGia = "Dat chuan";

        else

            danhGia = "Khong Dat chuan ";

        if(this.mayChieu)

            mayChieu = "Co";

        else

            mayChieu = "Khong";

        return super.toString()+String.format("%-15s"+"|"+"%-15s"+"|", mayChieu, danhGia);

    }

}

class MayTinh extends PhongHoc {

    private int soMayTinh;

    @Override

    boolean xetDatChuan() {

        if(duAnhSang() && dienTich/soMayTinh <= 1.5)

            return true;

        return false;

    }

    public int getSoMayTinh() {

        return soMayTinh;

    }

    public void setSoMayTinh(int soMayTinh) throws Exception {

        if(soMayTinh > 0)

            this.soMayTinh = soMayTinh;

        else

            throw new Exception("So may tinh khong duoc la 0!");

    }

    public MayTinh(String maPhong, String dayNha, double dienTich, int soBongDen, int soMayTinh) {

        super(maPhong, dayNha, dienTich, soBongDen);

        this.soMayTinh = soMayTinh;

    }

    public MayTinh() {

        super();

    }

    public MayTinh(String maPhong, String dayNha, double dienTich, int soBongDen) {

        super(maPhong, dayNha, dienTich, soBongDen);

    }

    @Override

    public String toString() {

        String danhGia;

        if(xetDatChuan())

            danhGia = "Dat chuan";

        else

            danhGia = "Khong Dat chuan";

        return super.toString()+String.format("%15d"+"|"+"%-15s"+"|", soMayTinh, danhGia);

    }

}

class ThiNghiem extends PhongHoc {

    private int sucChua;

    private boolean bonRua;

    @Override

    boolean xetDatChuan() {

        if(duAnhSang() && bonRua == true)

            return true;

        return false;

    }

    public int getSucChua() {

        return sucChua;

    }

    public void setSucChua(int sucChua) throws Exception {

        if(sucChua > 50)

            this.sucChua = sucChua;

        else

            throw new Exception("Suc chua khong duoc be hon 50!");

    }

    public boolean isBonRua() {

        return bonRua;

    }

    public void setBonRua(boolean bonRua) {

        this.bonRua = bonRua;

    }

    public ThiNghiem(String maPhong, String dayNha, double dienTich, int soBongDen, int sucChua, boolean bonRua) {

        super(maPhong, dayNha, dienTich, soBongDen);

        this.sucChua = sucChua;

        this.bonRua = bonRua;

    }

    public ThiNghiem() {

        super();

    }

    public ThiNghiem(String maPhong, String dayNha, double dienTich, int soBongDen) {

        super(maPhong, dayNha, dienTich, soBongDen);

    }

    @Override

    public String toString() {

        String danhGia, bonRua;

        if(xetDatChuan())

            danhGia = "Dat chuan";

        else

            danhGia = "Khong Dat chuan";

        if(this.bonRua)

            bonRua = "Co";

        else

            bonRua = "Khong";

        return super.toString()+String.format("%15d"+"|"+"%-15s"+"|"+"%-15s"+"|", sucChua, bonRua, danhGia);

    }

}

class DanhSachPhongHoc {

    ArrayList<PhongHoc> listPH;

    public DanhSachPhongHoc()

    {

        listPH = new ArrayList<PhongHoc>();

    }

    public ArrayList<PhongHoc> getAllPH()

    {

        return listPH;

    }

    public boolean themPhongHoc(PhongHoc ph) throws Exception

    {

        if(listPH.contains(ph))

            throw new Exception("Trung ma phong");

        else

            listPH.add(ph);

        return true;

    }

    public void xoaPH(PhongHoc ph)

    {

        listPH.remove(ph);

    }

    public void suaThongTinPH(PhongHoc ph)

    {

        int vt = listPH.indexOf(ph);

        if(vt>=0)

            listPH.set(vt, ph);

    }

    public PhongHoc timPH(String ma)

    {

        for (PhongHoc ph : listPH) {

            if(ph.maPhong.equalsIgnoreCase(ma))

                return ph;

        }

        return null;

    }

    public ArrayList<PhongHoc> timPHDatChuan()

    {

        int dem = 0;

        ArrayList<PhongHoc> list = new ArrayList<PhongHoc>();

        for (PhongHoc ph : listPH) {

            if(ph.xetDatChuan())

            {

                list.add(ph);

                dem++;

            }

        }

        if(dem==0)

            return null;

        return list;

    }

    public ArrayList<PhongHoc> timPHTheoSoMayTinh(int soBongDen)

    {

        int dem = 0;

        ArrayList<PhongHoc> list = new ArrayList<PhongHoc>();

        for (PhongHoc ph : listPH) {

            if(ph instanceof MayTinh)

            {

                if(ph.getSoBongDen() >= soBongDen)

                {

                    list.add(ph);

                    dem++;

                }

            }

        }

        if(dem==0)

            return null;

        return list;

    }

    public void sapXepDayNha()

    {

        Collections.sort(listPH, new Comparator<PhongHoc>(){

            public int compare(PhongHoc ph1, PhongHoc ph2)

            {

                return ph1.getDayNha().compareToIgnoreCase(ph2.getDayNha());

            }

        });

    }

    public void sapXepDienTichGiamDan()

    {

        Collections.sort(listPH, new Comparator<PhongHoc>() {

            public int compare(PhongHoc ph1, PhongHoc ph2)

            {

                if(ph1.getDienTich() == ph2.getDienTich())

                    return 0;

                else if(ph1.getDienTich() < ph2.getDienTich())

                    return 1;

                else

                    return -1;

            }

        });

    }

    public void sapXepBongDenTangDan()

    {

        Collections.sort(listPH, new Comparator<PhongHoc>() {

            public int compare(PhongHoc ph1, PhongHoc ph2)

            {

                Integer i1 = new Integer(ph1.getSoBongDen());

                Integer i2 = new Integer(ph2.getSoBongDen());

                return i1.compareTo(i2);

            }

        });

    }

    public int tongPhongHoc()

    {

        int sum = 0;

        for (PhongHoc ph : listPH)

        {

            sum++;

        }

        return sum;

    }

    public int tongPhongHocLT()

    {

        int sum = 0;

        for (PhongHoc ph : listPH)

        {

            if(ph instanceof LyThuyet)

                sum++;

        }

        return sum;

    }

    public int tongPhongHocMT()

    {

        int sum = 0;

        for (PhongHoc ph : listPH)

        {

            if(ph instanceof MayTinh)

                sum++;

        }

        return sum;

    }

    public int tongPhongHocTN()

    {

        int sum = 0;

        for (PhongHoc ph : listPH)

        {

            if(ph instanceof ThiNghiem)

                sum++;

        }

        return sum;

    }

}

public class quanlytruonghoc {

    static DanhSachPhongHoc listPH = new DanhSachPhongHoc();

    public static void main(String[] args) throws Exception {

        System.out.println("QUAN LY PHONG HOC");

        nhapCung();

        int chon = -1;

        do

        {

            chon = menu();

            switch (chon)

            {

            case 1:

                try {

                    nhapDuLieu();

                    System.out.println("Them phong thanh cong!");

                    xuatDanhSach(listPH.getAllPH());

                } catch (Exception e) {

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

                }

                break;

            case 2:

                xuatDanhSach(listPH.getAllPH());

                break;

            case 3:

                System.out.println("Danh sach phong dat chuan:");

                xuatDanhSach(listPH.timPHDatChuan());

                break;

            case 4:

                xuatDSTheoSoMayTinh();

                break;

            case 5:

                listPH.sapXepDayNha();

                xuatDanhSach(listPH.getAllPH());

                break;

            case 6:

                listPH.sapXepDienTichGiamDan();

                xuatDanhSach(listPH.getAllPH());

                break;

            case 7:

                listPH.sapXepBongDenTangDan();

                xuatDanhSach(listPH.getAllPH());

                break;

            case 8:

                System.out.println("Tong so phong hien co la: "+listPH.tongPhongHoc());

                break;

            case 9:

                System.out.println("Tong so phong ly thuyet: "+listPH.tongPhongHocLT());

                break;

            case 10:

                System.out.println("Tong so phong ly thuyet may tinh: "+listPH.tongPhongHocMT());

                break;

            case 11:

                System.out.println("Tong so phong ly thuyet thi nghiem: "+listPH.tongPhongHocTN());

                break;

            case 12:

                timKiemPH();

                break;

            case 13:

                xoaPH();

                break;

            case 14:

                capNhatSoMayTinh();

                break;

            }

        }while(chon != 0);

        System.out.println("Xin chon lai!");

    }

    public static int menu()

    {

        Scanner nhap = new Scanner(System.in);

        System.out.println("");

        System.out.println("\_\_\_\_\_\_\_\_\_\_\_ MENU\_\_\_\_\_\_\_\_\_\_\_");

        System.out.println("1.  Them phong hoc                          ");

        System.out.println("2.  In thong tin tat ca phong hoc          ");

        System.out.println("3.  In thong tin tat ca phong hoc dat chuan ");

        System.out.println("4.  In thong tin tat ca phong hoc co may tinh   ");

        System.out.println("5.  Sap xep theo day nha tang dan        ");

        System.out.println("6.  Sap xep theo dien tich giam dan        ");

        System.out.println("7.  Sap xep theo bong den tang dan     ");

        System.out.println("8.  Tong so phong hoc                    ");

        System.out.println("9.  Tong so phong hoc  ly thuyet            ");

        System.out.println("10. Tong so phong hoc  may tinh           ");

        System.out.println("11. Tong so phong hoc  thi nghiem          ");

        System.out.println("12. Tim kiem phong hoc                    ");

        System.out.println("13. Xoa phong hoc                      ");

        System.out.println("14. Cap nhat so may tinh                   ");

        System.out.println("0.  Thoat                   ");

        System.out.println("Vui long chon chuc nang:");

        int chon = nhap.nextInt();

        return chon;

    }

    public static void nhapCung() throws Exception

    {

        PhongHoc ph1 = new LyThuyet("PH001", "Dãy A", 150, 10, true);

        listPH.themPhongHoc(ph1);

        PhongHoc ph2 = new LyThuyet("PH002", "Dãy A", 200, 20, true);

        listPH.themPhongHoc(ph2);

        PhongHoc ph3 = new LyThuyet("PH003", "Dãy B", 150, 17, false);

        listPH.themPhongHoc(ph3);

        PhongHoc ph4 = new LyThuyet("PH004", "Dãy C", 200, 25, true);

        listPH.themPhongHoc(ph4);

        PhongHoc ph5 = new MayTinh("PH005", "Dãy B", 50, 10, 34);

        listPH.themPhongHoc(ph5);

        PhongHoc ph6 = new MayTinh("PH006", "Dãy B", 100, 10, 70);

        listPH.themPhongHoc(ph6);

        PhongHoc ph7 = new MayTinh("PH007", "Dãy C", 100, 10, 30);

        listPH.themPhongHoc(ph7);

        PhongHoc ph8 = new MayTinh("PH008", "Dãy C", 50, 10, 35);

        listPH.themPhongHoc(ph8);

        PhongHoc ph9 = new ThiNghiem("PH009", "Dãy D", 200, 20, 150, true);

        listPH.themPhongHoc(ph9);

        PhongHoc ph10 = new ThiNghiem("PH010", "Dãy D", 200, 20, 150, false);

        listPH.themPhongHoc(ph10);

        PhongHoc ph11 = new ThiNghiem("PH011", "Dãy D", 200, 20, 150, true);

        listPH.themPhongHoc(ph11);

        PhongHoc ph12 = new ThiNghiem("PH012", "Dãy D", 200, 15, 150, true);

        listPH.themPhongHoc(ph12);

    }

    public static void nhapDuLieu() throws Exception

    {

        String maPhong, dayNha;

        int soBongDen, soMayTinh, sucChua, loaiPH, chon;

        double dienTich;

        boolean mayChieu, bonRua;

        PhongHoc ph1 = new LyThuyet();

        PhongHoc ph2 = new MayTinh();

        PhongHoc ph3 = new ThiNghiem();

        Scanner nhap = new Scanner(System.in);

        try {

            System.out.println("Nhập mã phòng học:");

            maPhong = nhap.nextLine();

            if(listPH.timPH(maPhong)==null)

            {

                System.out.println("Nhập dãy nhà:");

                dayNha = nhap.nextLine();

                System.out.println("Nhập diện tích:");

                dienTich = nhap.nextDouble();

                System.out.println("Nhập số bóng đèn:");

                soBongDen = nhap.nextInt();

                System.out.println("Chọn loại phòng học:");

                System.out.println("    1.  Phòng lý thuyết");

                System.out.println("    2.  Phòng máy tính");

                System.out.println("    3.  Phòng thí nghiệm");

                loaiPH = nhap.nextInt();

                switch (loaiPH)

                {

                case 1:

                    System.out.println("Phòng có trang bị máy chiếu không?");

                    System.out.println("    1.  Có");

                    System.out.println("    2.  Không");

                    chon = nhap.nextInt();

                    if(chon == 1)

                        mayChieu = true;

                    else

                        mayChieu = false;

                    try {

                        ph1.setMaPhong(maPhong);

                        ph1.setDayNha(dayNha);

                        ph1.setDienTich(dienTich);

                        ph1.setSoBongDen(soBongDen);

                    } catch (Exception e) {

                        throw new Exception(e.getMessage());

                    }

                    ph1 = new LyThuyet(maPhong, dayNha, dienTich, soBongDen, mayChieu);

                    listPH.themPhongHoc(ph1);

                    break;

                case 2:

                    System.out.println("Nhập số máy tính:");

                    soMayTinh = nhap.nextInt();

                    try {

                        ph2.setMaPhong(maPhong);

                        ph2.setDayNha(dayNha);

                        ph2.setDienTich(dienTich);

                        ph2.setSoBongDen(soBongDen);

                    } catch (Exception e) {

                        throw new Exception(e.getMessage());

                    }

                    ph2 = new MayTinh(maPhong, dayNha, dienTich, soBongDen, soMayTinh);

                    listPH.themPhongHoc(ph2);

                    break;

                case 3:

                    System.out.println("Nhập sức chứa:");

                    sucChua = nhap.nextInt();

                    System.out.println("Phòng có bồn rửa không?");

                    System.out.println("    1.  Có");

                    System.out.println("    2.  Không");

                    chon = nhap.nextInt();

                    if(chon == 1)

                        bonRua = true;

                    else

                        bonRua = false;

                    try {

                        ph3.setMaPhong(maPhong);

                        ph3.setDayNha(dayNha);

                        ph3.setDienTich(dienTich);

                        ph3.setSoBongDen(soBongDen);

                    } catch (Exception e) {

                        throw new Exception(e.getMessage());

                    }

                    ph3 = new ThiNghiem(maPhong, dayNha, dienTich, soBongDen, sucChua, bonRua);

                    listPH.themPhongHoc(ph3);

                    break;

                }

            }

            else

                throw new Exception("Mã phòng học đã tồn tại!");

        } catch (Exception e) {

            System.out.println("Lỗi: "+e.getMessage());

            throw new Exception("Thêm không thành công!");

        }

    }

    public static void xuatTieuDePhongLT()

    {

        inGachNgang1();

        System.out.printf("|%-15s|%-10s|%-15s|%-15s|%-15s|%-15s|\n", "Mã phòng học", "Dãy nhà", "Diện tích(m^2)", "Số bóng đèn", "Máy chiếu", "Đánh giá");

        inGachNgang1();

    }

    public static void xuatTieuDePhongMT()

    {

        inGachNgang1();

        System.out.printf("|%-15s|%-10s|%-15s|%-15s|%-15s|%-15s|\n", "Mã phòng học", "Dãy nhà", "Diện tích(m^2)", "Số bóng đèn", "Số máy tính", "Đánh giá");

        inGachNgang1();

    }

    public static void xuatTieuDePhongTN()

    {

        inGachNgang2();

        System.out.printf("|%-15s|%-10s|%-15s|%-15s|%-15s|%-15s|%-15s|\n", "Mã phòng học", "Dãy nhà", "Diện tích(m^2)", "Số bóng đèn", "Sức chứa", "Bồn rửa", "Đánh giá");

        inGachNgang2();

    }

    public static void inGachNgang1()

    {

        for(int i = 0; i < 92; i++)

        {

            System.out.print("-");

        }

        System.out.print("\n");

    }

    public static void inGachNgang2()

    {

        for(int i = 0; i < 108; i++)

        {

            System.out.print("-");

        }

        System.out.print("\n");

    }

    public static void xuatDanhSach(ArrayList<PhongHoc> list)

    {

        System.out.println("Danh sách tất cả các phòng học:");

        System.out.println("Phòng lý thuyết:");

        xuatTieuDePhongLT();

        for (PhongHoc ph : list)

        {

            if(ph instanceof LyThuyet)

            {

                System.out.println(ph);

                inGachNgang1();

            }

        }

        System.out.println("Phòng máy tính:");

        xuatTieuDePhongMT();

        for (PhongHoc ph : list)

        {

            if(ph instanceof MayTinh)

            {

                System.out.println(ph);

                inGachNgang1();

            }

        }

        System.out.println("Phòng thí nghiệm:");

        xuatTieuDePhongTN();

        for (PhongHoc ph : list)

        {

            if(ph instanceof ThiNghiem)

            {

                System.out.println(ph);

                inGachNgang2();

            }

        }

    }

    public static void timKiemPH()

    {

        Scanner nhap = new Scanner(System.in);

        String ma;

        System.out.println("Nhập mã phòng học cần tìm:");

        ma = nhap.nextLine();

        PhongHoc phTim = listPH.timPH(ma);

        if(phTim==null)

            System.out.println("Mã phòng học không tồn tại!");

        else

        {

            if(phTim instanceof LyThuyet)

            {

                xuatTieuDePhongLT();

                System.out.println(phTim);

                inGachNgang1();

            }

            else if(phTim instanceof MayTinh)

            {

                xuatTieuDePhongMT();

                System.out.println(phTim);

                inGachNgang1();

            }

            else

            {

                xuatTieuDePhongTN();

                System.out.println(phTim);

                inGachNgang2();

            }

        }

    }

    public static void xoaPH()

    {

        int chon;

        Scanner nhap = new Scanner(System.in);

        System.out.println("Nhập mã phòng học cần xóa:");

        String ma = nhap.nextLine();

        PhongHoc phXoa = listPH.timPH(ma);

        if(phXoa!=null)

        {

            System.out.println("Bạn có chắc chắn muốn xóa?");

            System.out.println("    1.  Có");

            System.out.println("    2.  Không");

            chon = nhap.nextInt();

            if(chon == 1)

            {

                listPH.xoaPH(phXoa);

                System.out.println("Xóa thành công!");

                System.out.println("Danh Sách phòng học sau khi xóa:");

                xuatDanhSach(listPH.getAllPH());

            }

            else

                System.out.println("Bạn đã hủy thao tác xóa!");

        }

        else

            System.out.println("Mã phòng học không tồn tại!");

    }

    public static void capNhatSoMayTinh() throws Exception

    {

        String ma;

        int soMay;

        Scanner nhap = new Scanner(System.in);

        System.out.println("Nhập mã phòng học cần cập nhật:");

        ma = nhap.nextLine();

        PhongHoc phTim = listPH.timPH(ma);

        if(phTim==null)

            System.out.println("Mã phòng học không tồn tại!");

        else if(phTim instanceof MayTinh)

        {

            System.out.println("Thông tin phòng học trước khi cập nhật:");

            xuatTieuDePhongMT();

            System.out.println(phTim);

            inGachNgang1();

            System.out.println("Nhập số lượng máy tính:");

            soMay = nhap.nextInt();

            ((MayTinh) phTim).setSoMayTinh(soMay);

            System.out.println("Cập nhật thành công!");

            System.out.println("Thông tin phòng học sau khi đã cập nhật:");

            xuatTieuDePhongMT();

            System.out.println(phTim);

            inGachNgang1();

        }

        else

            System.out.println("Không phải mã phòng máy tính!");

    }

    public static void xuatDSTheoSoMayTinh()

    {

        int soMay;

        Scanner nhap = new Scanner(System.in);

        System.out.println("Nhập số lượng máy tính:");

        soMay = nhap.nextInt();

        ArrayList<PhongHoc> list = listPH.timPHTheoSoMayTinh(soMay);

        System.out.println("Danh sách các phòng máy tính có số máy từ "+soMay+" trở lên:");

        xuatTieuDePhongMT();

        for (PhongHoc ph : list) {

            {

                System.out.println(ph);

                inGachNgang1();

            }

        }

    }

}