

2. Một thư viện cần quản lý các tài liệu bao gồm Sách, Tạp chí, Báo. Mỗi tài liệu gồm có các thuộc tính sau: Mã tài liệu(Mã tài liệu là duy nhất), Tên nhà xuất bản, số bản phát hành.

- Các loại sách cần quản lý thêm các thuộc tính: tên tác giả, số trang.

- Các tạp chí cần quản lý thêm: Số phát hành, tháng phát hành.

- Các báo cần quản lý thêm: Ngày phát hành.

Yêu cầu 1: Xây dựng các lớp để quản lý tài liệu cho thư viện một cách hiệu quả.

Yêu cầu 2: Xây dựng lớp QuanLySach có các chức năng sau:

Thêm mới tài liệu: Sách, tạp chí, báo.

Xoá tài liệu theo mã tài liệu.

Hiện thị thông tin về tài liệu.

Tìm kiếm tài liệu theo loại: Sách, tạp chí, báo.

Thoát khỏi chương trình.

// Document

```
package b2;
```

```
public class Document {
    protected String id;
    protected String nxb;
    protected int number;

    public Document(String id, String nxb, int number)
    {
        this.id = id;
        this.nxb = nxb;
        this.number = number;
    }

    public String getId() {
        return id;
    }
}
```

```

    public void setId(String id) {
        this.id = id;
    }

    public String getNxb() {
        return nxb;
    }

    public void setNxb(String nxb) {
        this.nxb = nxb;
    }

    public int getNumber() {
        return number;
    }

    public void setNumber(int number) {
        this.number = number;
    }
}

```

// Book

```

package b2;

public class Book extends Document {
    private String author;
    private int numerPage;

    public Book(String id, String nxb, int number,
String author, int numerPage) {
        super(id, nxb, number);
        this.author = author;
        this.numerPage = numerPage;
    }

    public String getAuthor() {
        return author;
    }
}

```

```

    public void setAuthor(String author) {
        this.author = author;
    }

    public int getNumerPage() {
        return numerPage;
    }

    public void setNumerPage(int numerPage) {
        this.numerPage = numerPage;
    }

    @Override
    public String toString() {
        return "Book{" +
            "author='" + author + '\'' +
            ", numerPage=" + numerPage +
            ", id='" + id + '\'' +
            ", nxb='" + nxb + '\'' +
            ", number='" + number + '\'' +
            '}';
    }
}

```

// Journal

```

package b2;

public class Journal extends Document {

    private int issueNumber;
    private int monthIssue;

    public Journal(String id, String nxb, int number,
int issueNumber, int monthIssue) {
        super(id, nxb, number);
        this.issueNumber = issueNumber;
        this.monthIssue = monthIssue;
    }
}

```

```

public int getIssueNumber() {
    return issueNumber;
}

public void setIssueNumber(int issueNumber) {
    this.issueNumber = issueNumber;
}

public int getMonthIssue() {
    return monthIssue;
}

public void setMonthIssue(int monthIssue) {
    this.monthIssue = monthIssue;
}

@Override
public String toString() {
    return "Journal{" +
        "issueNumber=" + issueNumber +
        ", monthIssue=" + monthIssue +
        ", id='" + id + '\'' +
        ", nxb='" + nxb + '\'' +
        ", number='" + number + '\'' +
        '}';
}
}

```

// Newspaper

```

package b2;

public class Newspaper extends Document {
    private int dayIssue;

    public Newspaper(String id, String nxb, int number,
int dayIssue) {
        super(id, nxb, number);
        this.dayIssue = dayIssue;
    }
}

```

```

    }

    public int getDayIssue() {
        return dayIssue;
    }

    public void setDayIssue(int dayIssue) {
        this.dayIssue = dayIssue;
    }

    @Override
    public String toString() {
        return "Newspaper{" +
            "dayIssue=" + dayIssue +
            ", id='" + id + '\'' +
            ", nxb='" + nxb + '\'' +
            ", number='" + number + '\'' +
            '}';
    }
}

```

// Manager Document

```

package b2;

import java.util.ArrayList;
import java.util.List;

public class ManagerDocument {
    List<Document> documents;

    public ManagerDocument() {
        this.documents = new ArrayList<>();
    }

    public void addDocument(Document document) {
        this.documents.add(document);
    }

    public boolean deleteDocument(String id) {
        Document doc = this.documents.stream()
            .filter(document -> document.getId().equals(id))
            .findFirst().orElse(null);
        if (doc == null) {
            return false;
        }
        this.documents.remove(doc);
    }
}

```



```

        System.out.print("Enter nxb:");
        String nxb = scanner.nextLine();
        System.out.print("Enter number: ");
        int number = scanner.nextInt();
        System.out.print("Enter author: ");
        scanner.nextLine();
        String author = scanner.nextLine();
        System.out.print("Enter page number: ");
        int pageNumber = scanner.nextInt();
        Document book = new Book(id, nxb, number, author,
pageNumber);

        managerDocument.addDocument(book);
        System.out.println(book.toString());
        scanner.nextLine();
        break;
    }
    case "b": {
        System.out.print("Enter ID: ");
        String id = scanner.nextLine();
        System.out.print("Enter nxb:");
        String nxb = scanner.nextLine();
        System.out.print("Enter number: ");
        int number = scanner.nextInt();
        System.out.print("Enter Day issue: ");
        int dayIssue = scanner.nextInt();
        Document newspaper = new Newspaper(id, nxb,
number, dayIssue);

        managerDocument.addDocument(newspaper);
        System.out.println(newspaper.toString());
        scanner.nextLine();
        break;
    }
    case "c": {
        System.out.print("Enter ID: ");
        String id = scanner.nextLine();
        System.out.print("Enter nxb:");
        String nxb = scanner.nextLine();
        System.out.print("Enter number: ");
        int number = scanner.nextInt();
        System.out.print("Enter issue number : ");
        int issueNumber = scanner.nextInt();
        System.out.print("Enter issue month : ");
        int issueMonth = scanner.nextInt();
        Document journal = new Journal(id, nxb, number,
issueNumber, issueMonth);

        managerDocument.addDocument(journal);
        System.out.println(journal.toString());
        scanner.nextLine();
        break;
    }
    default:
        break;
    }
    break;
}
case "2": {

```

```

        System.out.println("Enter a to search book");
        System.out.println("Enter b to search newspaper");
        System.out.println("Enter c to search journal");
        String choice = scanner.nextLine();
        switch (choice) {
            case "a": {
                managerDocument.searchByBook();
                break;
            }
            case "b": {
                managerDocument.searchByNewspaper();
                break;
            }
            case "c": {
                managerDocument.searchByJournal();
                break;
            }
            default:
                System.out.println("Invalid");
                break;
        }
        break;
    }
    case "3": {
        managerDocument.showInfor();
        break;
    }
    case "4": {
        System.out.print("Enter id to remove: ");
        String id = scanner.nextLine();
        System.out.println(managerDocument.deleteDocument(id) ?
"Success" : "Fail");
    }
    break;
    case "5": {
        return;
    }
    default:
        System.out.println("Invalid");
        continue;
}

}

}
}

```

3. Các thí sinh dự thi đại học bao gồm các thí sinh thi khối A, B, và khối C. Các thí sinh cần quản lý các thông tin sau: Số báo danh, họ tên, địa chỉ, mức ưu tiên.

- Thí sinh thi khối A thi các môn: Toán, Lý, Hoá.
- Thí sinh thi khối B thi các môn: Toán, Hoá, Sinh.
- Thí sinh thi khối C thi các môn: Văn, Sử, Địa.

Yêu cầu 1: Xây dựng các lớp để quản lý các thí sinh dự thi đại học.

Yêu cầu 2: Xây dựng lớp TuyenSinh có các chức năng:

Thêm mới thí sinh.

Hiện thị thông tin của thí sinh và khối thi của thí sinh.

Tìm kiếm theo số báo danh.

Thoát khỏi chương trình.

// Candidate

```
package b3;

public class Candidate {
    protected String id;
    protected String name;
    protected String address;
    protected int priority;

    public Candidate(String id, String name, String
address, int priority) {
        this.id = id;
        this.name = name;
        this.address = address;
        this.priority = priority;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }
}
```

```

    public void setName(String name) {
        this.name = name;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public int getPriority() {
        return priority;
    }

    public void setPriority(int priority) {
        this.priority = priority;
    }
}

```

// Candidate A

```

package b3;

public class CandidateA extends Candidate {
    public static final String MON_TOAN = "Toan";
    public static final String MON_LY = "Ly";
    public static final String MON_HOA = "Hoa";
    public CandidateA(String id, String name, String
address, int priority) {
        super(id, name, address, priority);
    }

    @Override
    public String toString() {
        return "CandidateA{" +
            "id='" + id + '\'' +
            ", name='" + name + '\'' +
            ", address='" + address + '\'' +

```

```

        ", priority=" + priority + ", Subject:
" + MON_TOAN + " - " + MON_LY + " - " + MON_HOA +
        '}';
    }
}

```

// Candidate B

```

package b3;

public class CandidateB extends Candidate {

    public static final String MON_TOAN = "Toan";
    public static final String MON_HOA = "Hoa";
    public static final String MON_SINH = "Sinh";

    public CandidateB(String id, String name, String
address, int priority) {
        super(id, name, address, priority);
    }

    @Override
    public String toString() {
        return "CandidateB{" +
            "id='" + id + '\'' +
            ", name='" + name + '\'' +
            ", address='" + address + '\'' +
            ", priority=" + priority + ", Subject:
" + MON_TOAN + " - " + MON_SINH + " - " + MON_HOA +
            '}';
    }
}

```

// Candidate C

```

package b3;

public class CandidateC extends Candidate {

    public static final String MON_VAN = "Van";

```

```

        public static final String MON_SU = "Su";
        public static final String MON_DIA = "Dia";

        public CandidateC(String id, String name, String
address, int priority) {
            super(id, name, address, priority);
        }

        @Override
        public String toString() {
            return "CandidateC{" +
                "id='" + id + '\'' +
                ", name='" + name + '\'' +
                ", address='" + address + '\'' +
                ", priority=" + priority + ", Subject:
" + MON_VAN + " - " + MON_SU + " - " + MON_DIA +
                '\'';
        }
    }
}

```

// Manager Candidate

```

package b3;

import java.util.ArrayList;
import java.util.List;

public class ManagerCandidate {
    List<Candidate> candidates;
    public ManagerCandidate() {
        this.candidates = new ArrayList<>();
    }

    public void add(Candidate candidate) {
        this.candidates.add(candidate);
    }

    public void showInfor() {
        this.candidates.forEach(candidate ->
System.out.println(candidate.toString()));
    }
}

```

```

        public Candidate searchById(String id) {
            return
this.candidates.stream().filter(candidate ->
candidate.getId().equals(id)).findFirst().orElse(null);
        }

    }
}

```

// Main

```

package b3;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ManagerCandidate managerCandidate = new ManagerCandidate();
        while (true) {
            System.out.println("Application Manager Candidate");
            System.out.println("Enter 1: To insert candidate");
            System.out.println("Enter 2: To show information of candidate: ");
            System.out.println("Enter 3: To search candidate by id");
            System.out.println("Enter 4: To exit:");
            String line = scanner.nextLine();
            switch (line) {
                case "1": {
                    System.out.println("Enter a: to insert Candidate A");
                    System.out.println("Enter b: to insert Candidate B");
                    System.out.println("Enter c: to insert Candidate C");
                    String type = scanner.nextLine();
                    switch (type) {
                        case "a": {
                            managerCandidate.add(createCadidate(scanner, "a"));
                            break;
                        }
                        case "b": {
                            managerCandidate.add(createCadidate(scanner, "b"));
                            break;
                        }
                        case "c": {
                            managerCandidate.add(createCadidate(scanner, "c"));
                            break;
                        }
                        default:
                            System.out.println("Invalid");
                            break;
                    }
                    break;
                }
                case "2": {
                    managerCandidate.showInfor();
                    break;
                }
            }
        }
    }
}

```

```

    }
    case "3": {
        System.out.print("Enter ID: ");
        String id = scanner.nextLine();
        Candidate candidate = managerCandidate.searchById(id);
        if (candidate == null) {
            System.out.println("Not found");
        } else {
            System.out.println(candidate.toString());
        }
        break;
    }
    case "4": {
        return;
    }
    default:
        System.out.println("Invalid");
        continue;
}

}

}

public static Candidate createCadidate(Scanner scanner, String cate) {
    System.out.print("Enter ID: ");
    String id = scanner.nextLine();
    System.out.print("Enter name: ");
    String name = scanner.nextLine();
    System.out.print("Enter address: ");
    String address = scanner.nextLine();
    System.out.print("Enter Priotity: ");
    int priority = scanner.nextInt();
    scanner.nextLine();
    if (cate.equals("a")) {
        return new CandidateA(id, name, address, priority);
    } else if (cate.equals("b")) {
        return new CandidateB(id, name, address, priority);
    } else {
        return new CandidateC(id, name, address, priority);
    }
}

}

```

4. Để quản lý các hộ dân cư trong một khu phố, người ta cần các thông tin sau: Số thành viên trong gia đình, Số nhà, thông tin mỗi cá nhân trong gia đình. Với mỗi cá nhân, người ta quản lý các thông tin sau: Họ tên, Tuổi, Nghề nghiệp, số chứng minh nhân dân(duy nhất cho mỗi người).

Yêu cầu 1: Hãy xây dựng lớp Ngươi để quản lý thông tin của mỗi cá nhân.

Yêu cầu 2: Xây dựng lớp HoGiaDinh để quản lý thông tin của từng hộ gia đình.

Yêu cầu 3: Xây dựng lớp KhuPho để quản lý các thông tin của từng hộ gia đình.

Yêu cầu 4: Nhập n hộ dân. (n nhập từ bàn phím), hiển thị thông tin của các hộ trong khu phố.

// Person

```
package b4;
```

```
public class Person {
    private String name;
    private int age;
    private String job;
    private String passport;

    public Person(String name, int age, String job,
String passport) {
        this.name = name;
        this.age = age;
        this.job = job;
        this.passport = passport;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getJob() {
        return job;
    }

    public void setJob(String job) {
        this.job = job;
    }
}
```

```

    }

    public String getPassport() {
        return passport;
    }

    public void setPassport(String passport) {
        this.passport = passport;
    }

    @Override
    public String toString() {
        return "Person{" +
            "name='" + name + '\'' +
            ", age=" + age +
            ", job='" + job + '\'' +
            ", passport='" + passport + '\'' +
            '}';
    }
}

```

// Family

```

package b4;

import java.util.List;

public class Family {
    private List<Person> presons;
    private String address;

    public Family(List<Person> presons, String address)
    {
        this.presons = presons;
        this.address = address;
    }

    public List<Person> getPresons() {
        return presons;
    }
}

```



```

    public void setPresons(List<Person> presons) {
        this.presons = presons;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    @Override
    public String toString() {
        return "Family{" +
            "presons=" + presons +
            ", address='" + address + '\'' +
            '}';
    }
}

```

// Town

```

package b4;

import java.util.ArrayList;
import java.util.List;

public class Town {
    List<Family> families;

    public Town() {
        this.families = new ArrayList<>();
    }

    public List<Family> getFamilies() {
        return families;
    }
}

```

```

    public void setFamilies(List<Family> families) {
        this.families = families;
    }

    public void addFamily(Family family) {
        this.families.add(family);
    }
}

```

// Main

```

package b4;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

// This is sudo code. To overview how to do it.
// You can finish by fill you code

public class Main {
    public static void main(String[] args) {
        Town town = new Town();
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter n:");
        int n = scanner.nextInt();
        for(int i = 0; i < n; i++) {
            // Input foreach famili

            System.out.println("Enter address:");
            scanner.nextLine();
            String address = scanner.nextLine();

            // Enter person number in family
            List<Person> persons = new ArrayList<>();
            System.out.println("Enter number person");
            int number = scanner.nextInt();

            for (int j = 0; j < number; j++) {
                // Inout person for family
            }
        }
    }
}

```

```

        // After done input
        town.addFamily(new Family(persons,address ));

    }

}

```

5. Để quản lý khách hàng đến thuê phòng của một khách sạn, người ta cần các thông tin sau: Số ngày thuê, loại phòng, thông tin cá nhân của những người thuê phòng.

Biết rằng phòng loại A có giá 500\$, phòng loại B có giá 300\$ và loại C có giá 100\$.

Với mỗi cá nhân cần quản lý các thông tin sau: Họ tên, tuổi, số chứng minh nhân dân.

Yêu cầu 1: Hãy xây dựng lớp *Ngnoi* để quản lý thông tin cá nhân của những người thuê phòng.

Yêu cầu 2: Xây dựng lớp *KhachSan* để quản lý các thông tin về khách trọ.

Yêu cầu 3: Xây dựng các phương thức thêm mới, xoá theo số chứng minh nhân dân. Tính tiền thuê phòng cho khách(xác định khách bằng số chứng minh nhân dân) dựa vào công thức: (số ngày thuê * giá của từng loại phòng)

// Person

```
package b5;
```

```

public class Person {
    private String name;
    private int age;
    private String passport;
    private Room room;
    private int numberRent;
}

```

```
    public Person(String name, int age, String
passport, Room room, int numberRent) {
        this.name = name;
        this.age = age;
        this.passport = passport;
        this.numberRent = numberRent;
        this.room = room;
    }

    public int getNumberRent() {
        return numberRent;
    }

    public void setNumberRent(int numberRent) {
        this.numberRent = numberRent;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getPassport() {
        return passport;
    }

    public void setPassport(String passport) {
        this.passport = passport;
    }
}
```

```

    public Room getRoom() {
        return room;
    }

    public void setRoom(Room room) {
        this.room = room;
    }

    @Override
    public String toString() {
        return "Person{" +
            "name='" + name + '\'' +
            ", age=" + age +
            ", passport='" + passport + '\'' +
room.toString() +
            '}';
    }
}

```

// Hotel

```

package b5;

import java.util.ArrayList;
import java.util.List;

public class Hotel {
    private List<Person> persons;

    public Hotel() {
        persons = new ArrayList<>();
    }

    public void add(Person person) {
        this.persons.add(person);
    }

    public boolean delete(String passport) {
        Person person = this.persons.stream().filter(p ->
p.getPassport().equals(passport)).findFirst().orElse(null);

```

```

        if (person == null) {
            return false;
        } else {
            this.persons.remove(person);
            return true;
        }
    }

    public int calculator(String passport) {
        Person person = this.persons.stream().filter(p ->
p.getPassport().equals(passport)).findFirst().orElse(null);
        if (person == null) {
            return 0;
        }
        return person.getRoom().getPrice() *
person.getNumberRent();
    }

    public void show() {
        this.persons.forEach(p ->
System.out.println(p.toString()));
    }
}

```

// Hotel

```

package b5;

public class Room {
    protected String category;
    protected int price;

    public Room(String category, int price) {
        this.category = category;
        this.price = price;
    }

    public String getCategory() {
        return category;
    }

    public void setCategory(String category) {

```

```
        this.category = category;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }
}
```

// Room A

```
package b5;

public class RoomA extends Room {

    public RoomA() {
        super("A", 500);
    }

    @Override
    public String toString() {
        return "RoomA{" +
            "category='" + category + '\'' +
            ", price=" + price +
            '}';
    }
}
```

// Room B

```
package b5;

public class RoomB extends Room {
    public RoomB() {
        super("B", 300);
    }
}
```

```

@Override
public String toString() {
    return "RoomB{" +
        "category='" + category + '\'' +
        ", price=" + price +
        '}';
}
}

```

// Room C

```

package b5;

public class RoomC extends Room {
    public RoomC() {
        super("C", 100);
    }

    @Override
    public String toString() {
        return "RoomC{" +
            "category='" + category + '\'' +
            ", price=" + price +
            '}';
    }
}

```

// Main

```

package b5;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Hotel hotel = new Hotel();
        while (true) {
            System.out.println("Application Manager Candidate");
            System.out.println("Enter 1: To insert person for rent");
            System.out.println("Enter 2: To remove person by passport");
            System.out.println("Enter 3: To calculator price by passport");
            System.out.println("Enter 4: To show infor");
            System.out.println("Enter 5: To exit:");
            String line = scanner.nextLine();
            switch(line) {

```



```

        case "1": {
            System.out.print("Enter name: ");
            String name = scanner.nextLine();
            System.out.print("Enter age: ");
            int age = scanner.nextInt();
            System.out.print("Enter passport: ");
            scanner.nextLine();
            String passport = scanner.nextLine();
            System.out.println("Choise a to rent room type A");
            System.out.println("Choise b to rent room type B");
            System.out.println("Choise c to rent room type C");
            String choise = scanner.nextLine();
            Room room;
            if (choise.equals("a")) {
                room = new RoomA();
            } else if (choise.equals("b")) {
                room = new RoomB();
            } else if (choise.equals("c")) {
                room = new RoomC();
            } else {
                continue;
            }
            System.out.print("Enter number day for rent: ");
            int numberRent = scanner.nextInt();
            Person person = new Person(name, age, passport, room, numberRent);
            hotel.add(person);
            scanner.nextLine();
            break;
        }
        case "2": {
            System.out.print("Enter passport: ");
            String passport = scanner.nextLine();
            hotel.delete(passport);
            break;
        }
        case "3": {
            System.out.print("Enter passport: ");
            String passport = scanner.nextLine();
            System.out.println("Price: " + hotel.calculator(passport));
            break;
        }
        case "4": {
            hotel.show();
            break;
        }
        case "5": {
            return;
        }
        default:
            System.out.println("Invalid");
            continue;
    }
}
}
}

```

6. Để quản lý hồ sơ học sinh của trường THPT nhà trường cần các thông tin sau: Lớp, và các thông tin về cá nhân của mỗi học sinh.

Mỗi học sinh có các thông tin sau: Họ tên, tuổi, quê quán.

Yêu cầu 1: Xây dựng HocSinh để quản lý thông tin của mỗi học sinh.

Yêu cầu 2: Xây dựng các phương thức thêm, hiển thị thông tin của mỗi học sinh.

Yêu cầu 3: Cài đặt chương trình có các chức năng sau:

Thêm học sinh mới.

Hiển thị các học sinh 20 tuổi.

Cho biết số lượng các học sinh có tuổi là 23 và quê ở DN.

// Student

```
package b6;
```

```
public class Student {
    private String name;
    private int age;
    private String hometown;
    private int classStudent;

    public Student(String name, int age, String
hometown, int classStudent) {
        this.name = name;
        this.age = age;
        this.hometown = hometown;
        this.classStudent = classStudent;
    }

    public int getClassStudent() {
        return classStudent;
    }

    public void setClassStudent(int classStudent) {
        this.classStudent = classStudent;
    }
}
```

```

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getHometown() {
        return hometown;
    }

    public void setHometown(String hometown) {
        this.hometown = hometown;
    }

    @Override
    public String toString() {
        return "Student{" +
            "name='" + name + '\'' +
            ", age=" + age +
            ", hometown='" + hometown + '\'' +
            '}';
    }
}

```

// Student

```
package b6;
```

```
import java.util.ArrayList;
```

```

import java.util.List;
import java.util.stream.Collectors;

public class School {
    private List<Student> students;

    public School() {
        this.students = new ArrayList<>();
    }

    public void add(Student student) {
        this.students.add(student);
    }

    public List<Student> getStudent20YearOld() {
        return this.students.stream().filter(student ->
student.getAge() == 20).collect(Collectors.toList());
    }

    public long countStudent23YearOldInDN() {
        return this.students.stream().filter(student ->
student.getAge() == 23 &&
student.getHometown().equals("DN")).count();
    }
}

```

// Main

```

package b6;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        School school = new School();

        // add Student to school by func add(). for ex:
school.add(new Student("// infor");

```

```

        school.getStudent20YearOld();

        school.countStudent23YearOldInDN();
    }
}

```

7. Khoa CNTT – DHKHTN cần quản lý việc thanh toán tiền lương cho các cán bộ giáo viên trong khoa. Để quản lý được, khoa cần các thông tin sau:

Với mỗi cán bộ giáo viên có các thông tin sau: lương cứng, lương thưởng, tiền phạt, lương thực lĩnh, và các thông tin cá nhân: Họ tên, tuổi, quê quán, mã số giáo viên.

Yêu cầu 1: Xây dựng lớp *Nguo*i để quản lý các thông tin cá nhân của mỗi giáo viên.

Yêu cầu 2: Xây dựng lớp *CBGV* để quản lý các thông tin của các cán bộ giáo viên.

Yêu cầu 3: Xây dựng các phương thức thêm, xoá các cán bộ giáo viên theo mã số giáo viên. Yêu cầu 4: Tính lương thực lĩnh cho giáo viên: Lương thực = Lương cứng + lương thưởng – lương phạt.

// Teacher

```

package b7;

public class Teacher {
    private double salary;
    private double bonus;
    private double penaty;
    private double realSalary;
    private String name;
    private int age;
    private String hometown;
    private String id;
}

```

```
    public Teacher(double salary, double bonus, double
penaty, double realSalary, String name, int age, String
hometown, String id) {
        this.salary = salary;
        this.bonus = bonus;
        this.penaty = penaty;
        this.realSalary = realSalary;
        this.name = name;
        this.age = age;
        this.hometown = hometown;
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getHometown() {
        return hometown;
    }

    public void setHometown(String hometown) {
        this.hometown = hometown;
    }

    public String getId() {
        return id;
    }
}
```

```
}

public void setId(String id) {
    this.id = id;
}

public double getSalary() {
    return salary;
}

public void setSalary(double salary) {
    this.salary = salary;
}

public double getBonus() {
    return bonus;
}

public void setBonus(double bonus) {
    this.bonus = bonus;
}

public double getPenaty() {
    return penaty;
}

public void setPenaty(double penaty) {
    this.penaty = penaty;
}

public double getRealSalary() {
    return realSalary;
}

public void setRealSalary(double realSalary) {
    this.realSalary = realSalary;
}
}
```

// Manager Teacher

```
package b7;

import java.util.ArrayList;
import java.util.List;

public class ManagerTeacher {
    private List<Teacher> teachers;

    public ManagerTeacher() {
        this.teachers = new ArrayList<>();
    }

    public void add(Teacher teacher) {
        this.teachers.add(teacher);
    }

    public boolean deleteById(String id) {
        Teacher teacher =
this.teachers.stream().filter(t ->
t.getId().equals(id)).findFirst().orElse(null);
        if (teacher == null) {
            return false;
        }
        this.teachers.remove(teacher);
        return true;
    }

    public double getSalary(String id) {
        Teacher teacher =
this.teachers.stream().filter(t ->
t.getId().equals(id)).findFirst().orElse(null);
        if (teacher == null) {
            return 0;
        }
        return teacher.getSalary() + teacher.getBonus()
+ teacher.getPenaty();
    }
}
```


// Main

```
package b7;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ManagerTeacher managerTeacher = new
ManagerTeacher();
        while (true) {
            // show input for user choise
            // 1 to insert
            // 2 to remove => input id. output boolean
            // 4 get salary => input id. output double
            // 5 exit => return

        }
    }
}
```

8. Thư viện trung tâm đại học quốc gia có nhu cầu quản lý việc mượn, trả sách. Sinh viên đăng ký tham gia mượn sách thông qua thẻ mà thư viện cung cấp.

Với mỗi thẻ sẽ lưu các thông tin sau: Mã phiếu mượn, ngày mượn, hạn trả, số hiệu sách, và các thông tin cá nhân của sinh viên mượn sách. Các thông tin của sinh viên mượn sách bao gồm: Họ tên, tuổi, lớp.

Để đơn giản cho ứng dụng console. Chúng ta mặc định ngày mượn, ngày trả là số nguyên dương.

Yêu cầu 1: Xây dựng lớp *SinhVien* để quản lý thông tin của mỗi sinh viên.

Yêu cầu 2: Xây dựng lớp *TheMuon* để quản lý việc mượn trả sách của các sinh viên.

Yêu cầu 3: Xây dựng các phương thức: Thêm, xoá theo mã phiếu mượn và hiển thị thông tin các thẻ mượn.

// Card

```
package b8;

public class Card {
    private Student student;
    private String id;
    private int borrowDate;
    private int paymentDate;
    private int bookId;

    public Card(Student student, String id, int
borrowDate, int paymentDate, int bookId) {
        this.student = student;
        this.id = id;
        this.borrowDate = borrowDate;
        this.paymentDate = paymentDate;
        this.bookId = bookId;
    }

    public Student getStudent() {
        return student;
    }

    public void setStudent(Student student) {
        this.student = student;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }
}
```

```
public int getBorrowDate() {
    return borrowDate;
}

public void setBorrowDate(int borrowDate) {
    this.borrowDate = borrowDate;
}

public int getPaymentDate() {
    return paymentDate;
}

public void setPaymentDate(int paymentDate) {
    this.paymentDate = paymentDate;
}

public int getBookId() {
    return bookId;
}

public void setBookId(int bookId) {
    this.bookId = bookId;
}
}
```

// Student

```
package b8;

public class Student {
    private String name;
    private int age;
    private String school;

    public Student(String name, int age, String school)
    {
        this.name = name;
        this.age = age;
        this.school = school;
    }
}
```

```

    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getSchool() {
        return school;
    }

    public void setSchool(String school) {
        this.school = school;
    }
}

```

// Manager Card

```

package b8;

import java.util.ArrayList;
import java.util.List;

public class ManagerCard {

    private List<Card> cards;

    public ManagerCard() {
        this.cards = new ArrayList<>();
    }
}

```

```

    }

    public void add(Card card) {
        this.cards.add(card);
    }

    public boolean delete(String id) {
        Card card = this.cards.stream().filter(t ->
t.getId().equals(id)).findFirst().orElse(null);
        if (card == null) {
            return false;
        }
        this.cards.remove(card);
        return true;
    }
}

```

// Main

```

package b8;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ManagerCard managerCard = new ManagerCard();
        while (true) {
            // show input for user choise
            // 1 to insert
            // 2 to remove => input id. output boolean
            // 4 get salary => input id. output double
            // 5 exit => return

        }
    }
}

```