

문제1. 다음 요구사항을 만족하는 자바 프로그램을 작성하시오.

· 요구사항

- 1) StudentScore 클래스는 학생의 성적 정보를 관리하는 역할을 수행한다.
- 2) StudentScore 클래스를 아래표를 참고하여 정의하시오.
- 3) Main 클래스를 작성하여 StudentScore 객체를 생성하고 기능을 테스트하시오.

· StudentScore 클래스 정보

| 클래스 | StudentScore | 패키지 | test01 |
|-----|---|--------|-------------------------|
| 속성 | 속성명 | 타입 | 설명 |
| | studentName | String | 학생 이름 |
| | studentId | String | 학생 아이디 |
| | subject | String | 과목명 |
| | score | double | 과목 점수 |
| 생성자 | studentName, studentId, subject, score를 매개변수로 속성을 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | updateScore(double) | void | 새로운 점수를 받아 점수를 수정 |
| | getScore() | double | 속성 score의 Getter |
| | printStudentInfo() | void | 학생의 이름, ID, 과목명, 점수를 출력 |

```

package test01;
/**
 * 날짜 : 2024/00/00
 * 이름 : 김철학
 * 내용 : 자바 총정리 문제
 */
public class Main {
    public static void main(String[] args) {
        // 학생 성적 객체 생성
        StudentScore student = new StudentScore("홍길동", "S001", "자바", 90.0);

        // 학생 정보 출력
        student.printStudentInfo();
        System.out.println("-----");

        // 성적 수정 및 조회 테스트
        student.updateScore(98.0);
        System.out.println("-----");

        // 유효하지 않은 성적 입력
        student.updateScore(102.0);
        System.out.println("-----");

        // 학생 정보 출력
        student.printStudentInfo();
        System.out.println("-----");
    }
}

```

| 결과 |
|---|
| 학생이름 : 홍길동 학생 ID : S001 과목 : 자바 점수 : 90.0 ----- 점수 수정 완료 ----- 잘못된 점수 입력 ----- 학생이름 : 홍길동 학생 ID : S001 과목 : 자바 점수 : 98.0 ----- |

문제2. 다음 요구사항을 만족하는 자바 프로그램을 작성하시오.

· 요구사항

- 1) Product 클래스는 쇼핑물의 상품 정보를 관리하는 역할을 수행한다.
- 2) Product 클래스를 아래표를 참고하여 정의하시오.
- 3) Main 클래스를 작성하여 Product 객체를 생성하고 기능을 테스트하시오.

· Product 클래스 정보

| 클래스 | Product | 패키지 | test02 |
|-----|--|--------|-----------------------------|
| 속성 | 속성명 | 타입 | 설명 |
| | productId | String | 상품 아이디 |
| | productName | String | 상품 이름 |
| | price | int | 상품 가격 |
| | quantity | int | 재고 수량 |
| 생성자 | productId, productName, price, quantity를 매개변수로 속성을 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | updatePrice(int) | void | 새로운 가격을 받아 상품 가격을 수정 |
| | addStock(int) | void | 추가할 재고 수량을 받아 재고를 증가 |
| | printProductInfo() | void | 상품 ID, 상품 이름, 가격, 재고 수량을 출력 |

```

package test02;
/**
 * 날짜 : 2024/00/00
 * 이름 : 김철학
 * 내용 : 자바 총정리 문제
 */
public class Main {
    public static void main(String[] args) {
        // 상품 객체 생성
        Product product = new Product("P001", "키보드", 35000, 10);

        // 상품 정보 출력
        product.printProductInfo();
        System.out.println("-----");

        // 가격 수정 및 재고 관리 테스트
        product.updatePrice(30000);
        System.out.println("-----");

        // 상품 정보 출력
        product.printProductInfo();
        System.out.println("-----");

        // 재고 추가
        product.addStock(5);
        System.out.println("-----");

        // 상품 정보 출력
        product.printProductInfo();
        System.out.println("-----");
    }
}

```

결과

```

상품 ID : P001
상품명 : 키보드
상품가격 : 35000
재고수량 : 10
-----
키보드 가격 수정 됨
-----
상품 ID : P001
상품명 : 키보드
상품가격 : 30000
재고수량 : 10
-----
키보드 15개 재고 추가 됨
-----
상품 ID : P001
상품명 : 키보드
상품가격 : 30000
재고수량 : 15
-----

```

문제3. 다음 요구사항을 만족하는 자바 프로그램을 작성하시오.

· 요구사항

- 1) MovieTicket 클래스는 영화 티켓 정보를 관리하는 역할을 수행한다.
- 2) MovieTicket 클래스를 아래표를 참고하여 정의하시오.
- 3) Main 클래스를 작성하여 MovieTicket 객체를 생성하고 기능을 테스트하시오.

· MovieTicket 클래스 정보

| 클래스 | MovieTicket | 패키지 | test03 |
|-----|--|---------|--|
| 속성 | 속성명 | 타입 | 설명 |
| | movieTitle | String | 영화제목 |
| | screenTime | String | 상영시간 |
| | seatNumber | String | 좌석번호 |
| | isBooked | boolean | 예매여부 |
| 생성자 | - movieTitle, screenTime, seatNumber를 매개변수로 속성을 초기화 - isBooked은 false 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | bookTicket() | void | - 티켓이 예약 가능한 경우 예약 처리 - isBooked을 true로 설정 - 티켓이 이미 예약된 경우 적절한 메시지를 출력 |
| | cancelBook() | void | - 티켓이 예약된 경우 예약을 취소 - isBooked을 false로 설정 - 티켓이 예약되지 않은 경우 적절한 메시지를 출력 |
| | printTicketInfo() | void | 영화제목, 상영시간, 좌석번호, 예약여부 출력 |

```

package test03;
/**
 * 날짜 : 2024/00/00
 * 이름 : 김철학
 * 내용 : 자바 총정리 문제
 */
public class Main {
    public static void main(String[] args) {
        // 티켓 객체 생성
        MovieTicket ticket = new MovieTicket("타이타닉", "19:00", "A1");

        // 티켓 정보 출력
        ticket.printTicketInfo();
        System.out.println("-----");

        // 예약 테스트
        ticket.bookTicket();
        System.out.println("-----");

        // 티켓 정보 출력
        ticket.printTicketInfo();
        System.out.println("-----");

        // 이미 예약된 경우
        ticket.bookTicket();
        System.out.println("-----");

        // 취소 테스트
        ticket.cancelBooking();
        System.out.println("-----");

        // 티켓 정보 출력
        ticket.printTicketInfo();
        System.out.println("-----");

        // 이미 취소된 경우
        ticket.cancelBooking();
        System.out.println("-----");
    }
}

```

결과

```

영화제목 : 타이타닉
상영시간 : 19:00
좌석번호 : A1
예매여부 : No
-----
타이타닉, 19:00, A1 예매 완료
-----
영화제목 : 타이타닉
상영시간 : 19:00
좌석번호 : A1
예매여부 : Yes
-----
이미 예매됨
-----
타이타닉, 19:00, A1 취소 완료
-----
영화제목 : 타이타닉
상영시간 : 19:00
좌석번호 : A1
예매여부 : No
-----
아직 예매 안됨
-----

```

문제4. 다음 요구사항을 만족하는 자바 프로그램을 작성하시오.

· 요구사항

- 1) Book과 Member 클래스를 아래표를 참고하여 정의하시오
- 2) Main 클래스를 작성하여 Book, Member 객체를 생성하고, 기능을 테스트하시오.

· Book 클래스 정보

| 클래스 | Book | 패키지 | test04 |
|-----|--|---------|----------------------------------|
| 속성 | 속성명 | 타입 | 설명 |
| | title | String | 도서제목 |
| | author | String | 저자 |
| | isbn | String | 도서번호 |
| | isBorrowd | boolean | 도서 대출 여부 |
| 생성자 | - title, author, isbn을 매개변수로 속성을 초기화 - isBorrowd는 false 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | borrowBook() | void | 도서가 대출되지 않은 상태라면 대출 |
| | returnBook() | void | 도서가 대출된 상태라면 반납 |
| | getBookInfo() | void | 도서의 정보를 출력 (제목, 저자, ISBN, 대출 여부) |
| | getTitle() | String | 속성 title의 Getter |

· Member 클래스 정보

| 클래스 | Member | 패키지 | test04 |
|-----|---|--------|--------------------------------|
| 속성 | 속성명 | 타입 | 설명 |
| | name | String | 회원 이름 |
| | memberId | String | 회원 아이디 |
| | borrowedBook | Book | 회원이 대출한 도서 |
| 생성자 | - name, memberId를 매개변수로 속성을 초기화 - borrowedBook은 null 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | borrowBook(Book) | void | 도서를 대출하고, 회원이 갖는 대출 도서에서 추가 |
| | returnBook(Book) | void | 도서를 반납하고, 회원이 갖는 대출 도서에서 제거 |
| | getMemberInfo() | void | 회원의 정보를 출력 (이름, 회원 ID, 대출한 도서) |

```

package test04;
/**
 * 날짜 : 2024/00/00
 * 이름 : 김철학
 * 내용 : 자바 총정리 문제
 */
public class Main {
    public static void main(String[] args) {
        // 도서 생성
        Book book = new Book("이것이 자바다", "신용권", "101-1234-1001");

        // 회원 생성
        Member member = new Member("홍길동", "A001");

        // 도서 정보 출력
        book.getBookInfo();
        System.out.println("-----");

        // 도서 대출
        member.borrowBook(book);
        System.out.println("-----");

        // 도서 정보 출력
        book.getBookInfo();
        System.out.println("-----");

        // 회원 정보 출력
        member.getMemberInfo();
        System.out.println("-----");

        // 도서 대출
        member.borrowBook(book);
        System.out.println("-----");

        // 도서 반납
        member.returnBook(book);
        System.out.println("-----");

        // 회원 정보 출력
        member.getMemberInfo();
        System.out.println("-----");
    }
}

```

결과

```

도서명 : 이것이 자바다
저자 : 신용권
ISBN : 101-1234-1001
대출여부 : 가능
-----
도서대출 : 이것이 자바다
-----
도서명 : 이것이 자바다
저자 : 신용권
ISBN : 101-1234-1001
대출여부 : 불가능
-----
회원명 : 홍길동
아이디 : A001
대출한 도서 : 이것이 자바다
-----
이것이 자바다 이미 대출됨
-----
도서반납 : 이것이 자바다
-----
회원명 : 홍길동
아이디 : A001
대출한 도서 : 없음
-----

```


문제5. 다음 요구사항을 만족하는 자바 프로그램을 작성하시오.

· 요구사항

- 1) Subject, Student 클래스를 아래표를 참고하여 정의하시오.
- 2) Main 클래스를 작성하여 Subject, Student 객체를 생성하고, 기능을 테스트하시오.

· Subject 클래스 정보

| 클래스 | Subject | 패키지 | test05 |
|-----|--|-----------|------------------------|
| 속성 | 속성명 | 타입 | 설명 |
| | subName | String | 과목 이름 |
| | students | Student[] | 과목을 수강하는 학생 배열 |
| | studentCount | int | 현재 수강 중인 학생 수 |
| 생성자 | <ul style="list-style-type: none"> - subName을 매개변수로 속성을 초기화 - students는 크기 10 배열로 초기화 - studentCount 초기값은 0 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | addStudent(Student) | void | 학생을 과목에 추가 |
| | printSubjectInfo() | void | 과목 이름과 수강하는 학생들의 이름 출력 |
| | getName() | String | 속성 name의 Getter |

· Student 클래스 정보

| 클래스 | Student | 패키지 | test05 |
|-----|---|-----------|---------------------------|
| 속성 | 속성명 | 타입 | 설명 |
| | name | String | 학생 이름 |
| | studentId | String | 학생 아이디 |
| | subjects | Subject[] | 학생이 수강하는 과목 배열 |
| | scores | int[] | 학생의 각 과목 성적 배열 |
| | subjectCount | int | 현재 수강 중인 과목 수 |
| 생성자 | <ul style="list-style-type: none"> - name, studentId를 매개변수로 속성을 초기화 - subjects는 크기 10 배열로 초기화 - scores는 크기 10 배열로 초기화 - subjectCount 초기값은 0 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | enrollSubject(Subject) | void | 과목을 수강 신청 |
| | setScore(Subject, int) | void | 특정 과목의 성적 입력 |
| | printStudentInfo() | void | 학생의 이름, ID, 수강 과목 및 성적 출력 |
| | getName() | String | 속성 name의 Getter |

```

package test05;
/**
 * 날짜 : 2024/00/00
 * 이름 : 김철학
 * 내용 : 자바 총정리 문제
 */
public class Main {
    public static void main(String[] args) {
        // 과목 생성
        Subject java = new Subject("자바");
        Subject database = new Subject("데이터베이스");

        // 학생 생성
        Student student1 = new Student("김유신", "S001");
        Student student2 = new Student("김춘추", "S002");

        // 학생 과목 수강 신청
        student1.enrollSubject(java);
        student1.enrollSubject(database);
        System.out.println("-----");

        // 학생 과목 수강 신청
        student2.enrollSubject(java);
        student2.enrollSubject(database);
        System.out.println("-----");

        // 성적 입력
        student1.setScore(java, 96);
        student1.setScore(database, 90);
        System.out.println("-----");

        // 성적 입력
        student2.setScore(java, 86);
        student2.setScore(database, 76);
        System.out.println("-----");

        // Subject 정보 출력
        java.printSubjectInfo();
        System.out.println("-----");

        // Subject 정보 출력
        database.printSubjectInfo();
        System.out.println("-----");

        // 학생 정보 출력
        student1.printStudentInfo();
        System.out.println("-----");

        // 학생 정보 출력
        student2.printStudentInfo();
        System.out.println("-----");
    }
}

```

| 결과 |
|--|
| 김유신 - 자바 과목 신청완료 김유신 - 데이터베이스 과목 신청완료 ----- |
| 김춘추 - 자바 과목 신청완료 김춘추 - 데이터베이스 과목 신청완료 ----- |
| 김유신 - 자바 점수 입력완료 김유신 - 데이터베이스 점수 입력완료 ----- |
| 김춘추 - 자바 점수 입력완료 김춘추 - 데이터베이스 점수 입력완료 ----- |
| 과목명 : 자바 수강생 : 김유신, 김춘추, ----- |
| 과목명 : 데이터베이스 수강생 : 김유신, 김춘추, ----- |
| 학생명 : 김유신 아이디 : S001 성적 - 자바:96 - 데이터베이스:90 ----- |
| 학생명 : 김춘추 아이디 : S002 성적 - 자바:86 - 데이터베이스:76 ----- |

문제6. 다음 요구사항을 만족하는 자바 프로그램을 작성하시오.

· 요구사항

- 1) Patient와 Doctor 클래스를 아래표를 참고하여 정의하시오
- 2) Main 클래스를 작성하여 Doctor와 Patient 객체를 생성하고, 기능을 테스트하시오.

· Patient 클래스 정보

| 클래스 | Patient | 패키지 | test06 |
|-----|---|--------|----------|
| 속성 | 속성명 | 타입 | 설명 |
| | name | String | 환자 이름 |
| | id | String | 환자 아이디 |
| | age | int | 환자 나이 |
| | diagnosis | String | 진단명 |
| 생성자 | name, id, age, diagnosis를 매개변수로 속성을 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | toString() | String | 속성 정보 출력 |

· Doctor 클래스 정보

| 클래스 | Doctor | 패키지 | test06 |
|-----|--|---------------|----------|
| 속성 | 속성명 | 타입 | 설명 |
| | name | String | 의사 이름 |
| | id | String | 의사 아이디 |
| | special | String | 전문분야 |
| | patients | List<Patient> | 담당 환자 |
| 생성자 | - name, id, special을 매개변수로 속성을 초기화 - patients는 ArrayList 생성 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | addPatient(Patient) | void | 환자 추가 |
| | toString() | String | 속성 정보 출력 |

```

package test06;

/**
 * 날짜 : 2024/00/00
 * 이름 : 김철학
 * 내용 : 자바 총정리 문제
 */
public class Main {
    public static void main(String[] args) {
        // 의사 생성
        Doctor doctor1 = new Doctor("김의사", "D001", "내과");
        Doctor doctor2 = new Doctor("박의사", "D002", "외과");

        // 환자 생성 및 의사에게 할당
        Patient patient1 = new Patient("이환자", "P001", 30, "감기");
        Patient patient2 = new Patient("박환자", "P002", 45, "고혈압");
        Patient patient3 = new Patient("최환자", "P003", 25, "알레르기");
        Patient patient4 = new Patient("김환자", "P004", 36, "비염");
        Patient patient5 = new Patient("정환자", "P005", 52, "허리디스크");

        doctor1.addPatient(patient1);
        doctor1.addPatient(patient2);
        doctor1.addPatient(patient3);

        doctor2.addPatient(patient4);
        doctor2.addPatient(patient5);

        // 의사 정보 및 담당 환자 정보 출력
        System.out.println(doctor1);
        System.out.println(doctor2);
    }
}

```

| 결과 |
|--|
| <p>의사: 김의사(ID: D001, 전문분야: 내과)</p> <p>담당 환자 목록:</p> <ul style="list-style-type: none"> - Patient [name=이환자, id=P001, age=30, diagnosis=감기] - Patient [name=박환자, id=P002, age=45, diagnosis=고혈압] - Patient [name=최환자, id=P003, age=25, diagnosis=알레르기] <p>의사: 박의사(ID: D002, 전문분야: 외과)</p> <p>담당 환자 목록:</p> <ul style="list-style-type: none"> - Patient [name=김환자, id=P004, age=36, diagnosis=비염] - Patient [name=정환자, id=P005, age=52, diagnosis=허리디스크] |

문제7. 다음 요구사항을 만족하는 자바 프로그램을 작성하시오.

· 요구사항

- 1) Product 클래스는 쇼핑물의 일반 상품 정보를 관리하는 역할을 수행한다.
- 2) Computer와 Clothes 클래스는 Product 클래스를 상속받아 각각 상품 정보를 관리한다.
- 3) Main 클래스를 작성하여 Computer, Clothes 객체를 생성하고, 기능을 테스트하시오.

· Product 클래스 정보

| | | | |
|-----|--|--------|----------|
| 클래스 | Product | 패키지 | test07 |
| 속성 | 속성명 | 타입 | 설명 |
| | prodId | String | 상품 아이디 |
| | prodName | String | 상품 이름 |
| | price | int | 상품 가격 |
| | quantity | int | 재고 수량 |
| 생성자 | prodId, prodName, price, quantity를 매개변수로 속성을 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | printProductInfo() | void | 상품 정보 출력 |

· Computer 클래스 정보

| | | | |
|-----|--|--------|----------|
| 클래스 | Computer | 패키지 | test07 |
| 속성 | 속성명 | 타입 | 설명 |
| | manufactor | String | 제조사 |
| 생성자 | prodId, prodName, price, quantity, manufactor를 매개변수로 속성을 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | printProductInfo() | void | 상품 정보 출력 |

· Clothes 클래스 정보

| | | | |
|-----|--|--------|---------------------|
| 클래스 | Clothes | 패키지 | test07 |
| 속성 | 속성명 | 타입 | 설명 |
| | size | String | 의류 사이즈(XL, L, M, S) |
| 생성자 | prodId, prodName, price, quantity, size를 매개변수로 속성을 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | printProductInfo() | void | 상품 정보 출력 |

```
package test07;

/**
 * 날짜 : 2024/00/00
 * 이름 : 김철학
 * 내용 : 자바 총정리 문제
 */
public class Main {
    public static void main(String[] args) {
        // 컴퓨터 객체 생성
        Computer computer = new Computer("E001", "데스크탑", 150, 10, "삼성전자");
        computer.printProductInfo();
        System.out.println("-----");

        // 의류 객체 생성
        Clothes tshirt = new Clothes("C001", "T-Shirt", 20, 50, "L");
        tshirt.printProductInfo();
        System.out.println("-----");
    }
}
```

결과

```
상품코드 : E001
상품명 : 데스크탑
상품가격 : 150
재고수량 : 10
제조사 : 삼성전자
-----
상품코드 : C001
상품명 : T-Shirt
상품가격 : 20
재고수량 : 50
사이즈 : L
-----
```

문제8. 다음 요구사항을 만족하는 자바 프로그램을 작성하시오.

· 요구사항

- 1) Person 클래스는 사람의 이름과 생년월일을 관리합니다.
- 2) Employee 클래스는 Person 클래스를 상속받아 직원 정보를 관리합니다.
- 3) Developer 클래스는 Employee 클래스를 상속받아 개발자 정보를 관리합니다.
- 4) Main 클래스를 작성하여 Person, Employee, Developer 객체를 생성하고, 기능을 테스트하시오.

· Person 클래스 정보

| | | | |
|-----|----------------------------|--------|----------|
| 클래스 | Person | 패키지 | test08 |
| 속성 | 속성명 | 타입 | 설명 |
| | name | String | 이름 |
| | birth | String | 생년월일 |
| 생성자 | name, birth를 매개변수로 속성을 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | printPersonInfo() | void | 사람 정보 출력 |

· Employee 클래스 정보

| | | | |
|-----|---|--------|-----------|
| 클래스 | Employee | 패키지 | test08 |
| 속성 | 속성명 | 타입 | 설명 |
| | employeeId | String | 직원 아이디 |
| | dept | String | 부서 |
| | salary | int | 급여 |
| 생성자 | employeeId, dept, salary를 매개변수로 속성을 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | raiseSalary(int) | void | 급여 인상 메서드 |
| | changeDept(String) | void | 부서 변경 메서드 |
| | printEmployeeInfo() | void | 직원 정보 출력 |

· Developer 클래스 정보

| | | | |
|-----|----------------------|--------|--------------|
| 클래스 | Developer | 패키지 | test08 |
| 속성 | 속성명 | 타입 | 설명 |
| | field | String | 개발 분야 |
| 생성자 | field를 매개변수로 속성을 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | changeField(String) | void | 개발 분야 변경 메서드 |
| | printDeveloperInfo() | void | 개발자 정보 출력 |

```

package test08;
/**
 * 날짜 : 2024/00/00
 * 이름 : 김철학
 * 내용 : 자바 총정리 문제
 */
public class Main {
    public static void main(String[] args) {
        Person person = new Person("김유신", "1990-05-07");
        person.printPersonInfo();
        System.out.println("-----");

        Employee emp = new Employee("김춘추", "1992-07-14", "E001", "영업부", 200);
        emp.printEmployeeInfo();
        System.out.println("-----");

        Developer dev = new Developer("이순신", "1995-10-21", "D001", "개발부", 300, "프론트");
        dev.printDeveloperInfo();
        System.out.println("-----");

        // 부서 변경 테스트
        emp.changeDepartment("인사부");
        emp.printEmployeeInfo();
        System.out.println("-----");

        // 급여 인상 테스트
        dev.changeField("백엔드");
        dev.raiseSalary(50);
        dev.printDeveloperInfo();
        System.out.println("-----");
    }
}

```

| 결과 |
|--|
| 이름 : 김유신 생년월일 : 1990-05-07 ----- |
| 이름 : 김춘추 생년월일 : 1992-07-14 직원 아이디 : E001 부서 : 영업부 급여 : 200 ----- |
| 이름 : 이순신 생년월일 : 1995-10-21 직원 아이디 : D001 부서 : 개발부 급여 : 300 개발분야 : 프론트 ----- |
| 부서 변경 이름 : 김춘추 생년월일 : 1992-07-14 직원 아이디 : E001 부서 : 인사부 급여 : 200 ----- |
| 개발 분야 변경 급여 인상 이름 : 이순신 생년월일 : 1995-10-21 직원 아이디 : D001 부서 : 개발부 급여 : 350 개발분야 : 백엔드 ----- |

문제9. 다음 요구사항을 만족하는 자바 프로그램을 작성하시오.

· 요구사항

- 1) Commentable 인터페이스는 댓글 등록과 댓글 목록을 반환하는 메서드를 정의합니다.
- 2) Article 클래스는 Commentable 인터페이스를 구현하고 게시물의 기본 정보를 관리합니다.
- 3) Comment 클래스는 게시물의 댓글 기본 정보를 관리합니다.
- 4) User 클래스는 사용자의 기본 정보를 관리합니다.
- 5) Board 클래스는 게시판의 게시물과 사용자를 관리합니다.
- 6) Main 클래스를 작성하여 Article, Comment, User, Board 객체를 생성하고, 기능을 테스트하시오.

· Commentable 인터페이스 정보

| 인터페이스 | Commentable | 패키지 | test09 |
|-----------|---------------------|---------------|----------|
| 추상 메서드 | 메서드명 | 반환타입 | 설명 |
| | addComment(Comment) | void | 댓글 등록 |
| | getComments() | List<Comment> | 댓글 목록 반환 |

· Comment 클래스 정보

| 클래스 | Comment | 패키지 | test09 |
|-----|----------------------------|--------|--------------------|
| 속성 | 속성명 | 타입 | 설명 |
| | no | int | 댓글 번호 |
| | content | String | 댓글 내용 |
| 생성자 | no, content를 매개변수로 속성을 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | getNo() | int | 속성 no의 Getter |
| | getContent() | String | 속성 content의 Getter |

· User 클래스 정보

| 클래스 | User | 패키지 | test09 |
|-----|--------------------------|--------|-----------------|
| 속성 | 속성명 | 타입 | 설명 |
| | uid | String | 사용자 아이디 |
| | name | String | 이름 |
| 생성자 | uid, name을 매개변수로 속성을 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | getUid() | String | 속성 uid의 Getter |
| | getName() | String | 속성 name의 Getter |

· Article 클래스 정보

| 클래스 | Article | 패키지 | test09 |
|-----|---|---------------|---------------------|
| 속성 | 속성명 | 타입 | 설명 |
| | no | String | 글 번호 |
| | title | String | 글 제목 |
| | content | String | 글 내용 |
| | user | User | 사용자 객체 |
| | comments | List<Comment> | 댓글 목록 |
| 생성자 | - no, title, content, user를 매개변수로 속성을 초기화 - comments는 ArrayList 생성 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | getNo() | String | 속성 no의 Getter |
| | getTitle() | String | 속성 title의 Getter |
| | getContent() | String | 속성 content의 Getter |
| | getUser() | User | 속성 user의 Getter |
| | addComment(Comment) | void | Commentable 메서드 재정의 |
| | getComments() | List<Comment> | Commentable 메서드 재정의 |

· Board 클래스 정보

| 클래스 | Board | 패키지 | test09 |
|-----|---|---------------|---------------------|
| 속성 | 속성명 | 타입 | 설명 |
| | articles | List<Article> | 게시물 목록 |
| | users | List<User> | 사용자 목록 |
| 생성자 | - articles는 ArrayList 생성 초기화 - users는 ArrayList 생성 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | addArticle(Article) | void | 게시물 추가 |
| | addUser(User) | void | 사용자 추가 |
| | getArticles() | List<Article> | 속성 articles의 Getter |
| | getUsers() | List<User> | 속성 users의 Getter |

```

package test09;
/**
 * 날짜 : 2024/00/00
 * 이름 : 김철학
 * 내용 : 자바 총정리 문제
 */
public class Main {
    public static void main(String[] args) {
        // 게시판 생성
        Board board = new Board();

        // 사용자 생성
        User user1 = new User("A101", "김유신");
        User user2 = new User("A102", "김춘추");
        User user3 = new User("A103", "장보고");

        // 게시물 생성
        Article article1 = new Article(1, "안녕하세요.", "내용1", user1);
        Article article2 = new Article(2, "반갑습니다.", "내용2", user2);
        Article article3 = new Article(3, "감사합니다.", "내용3", user3);

        // 댓글 생성
        Comment comment1 = new Comment(101, "1번글의 101번 댓글 입니다.");
        Comment comment2 = new Comment(102, "2번글의 102번 댓글 입니다.");
        Comment comment3 = new Comment(103, "1번글의 103번 댓글 입니다.");
        Comment comment4 = new Comment(104, "3번글의 104번 댓글 입니다.");

        // 게시물에 댓글 추가
        article1.addComment(comment1);
        article1.addComment(comment3);
        article2.addComment(comment2);
        article3.addComment(comment4);

        // 게시판에 게시물 추가
        board.addArticle(article1);
        board.addArticle(article2);
        board.addArticle(article3);

        // 게시판의 모든 게시물 출력
        List<Article> articles = board.getArticles();
    }
}

```

코드 완성하기

결과

```

-----
| 번호 |   제목   |   내용   | 작성자 |
-----
|  1   | 안녕하세요. | 내용1 | 김유신 |
-----
↳ 1번글의 101번 댓글 입니다.
↳ 1번글의 103번 댓글 입니다.
-----
|  2   | 반갑습니다. | 내용2 | 김춘추 |
-----
↳ 2번글의 102번 댓글 입니다.
-----
|  3   | 감사합니다. | 내용3 | 장보고 |
-----
↳ 3번글의 104번 댓글 입니다.
-----

```

문제10. 다음 요구사항을 만족하는 자바 프로그램을 작성하시오.

· 요구사항

- 1) Product 인터페이스는 Clothes, Electronics, Food 클래스 구조 설계를 위해 정의합니다.
- 2) Clothes 클래스는 Product 인터페이스를 구현하고 의류 상품의 정보를 관리합니다.
- 3) Electronics 클래스는 Product 인터페이스를 구현하고 전자기기 상품의 정보를 관리합니다.
- 4) Food 클래스는 Product 인터페이스를 구현하고 식품 상품의 정보를 관리합니다.
- 5) Customer 클래스는 고객의 기본 정보를 관리합니다.
- 6) Order 클래스는 상품 주문 정보를 관리합니다.
- 7) ShoppingMall 클래스는 고객 목록, 주문 내역 목록을 관리합니다.
- 8) Main 클래스를 작성하고 ShoppingMall, Customer, Product, Order 생성하고, 기능을 테스트하시오.

· Product 인터페이스 정보

| 인터페이스 | Product | 패키지 | test10 |
|-----------|------------|--------|---------|
| 추상 메서드 | 메서드명 | 반환타입 | 설명 |
| | getName() | String | 상품명 반환 |
| | getPrice() | int | 상품가격 반환 |

· Clothes 클래스 정보

| 클래스 | Clothes | 패키지 | test10 |
|-----|----------------------------|--------|-----------------------|
| 속성 | 속성명 | 타입 | 설명 |
| | name | String | 의류 상품명 |
| | price | int | 의류 상품가격 |
| 생성자 | name, price를 매개변수로 속성을 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | getName() | String | Product 인터페이스 메서드 재정의 |
| | getPrice() | int | Product 인터페이스 메서드 재정의 |

· Food 클래스 정보

| 클래스 | Food | 패키지 | test10 |
|-----|----------------------------|--------|-----------------------|
| 속성 | 속성명 | 타입 | 설명 |
| | name | String | 식품 상품명 |
| | price | int | 식품 상품가격 |
| 생성자 | name, price를 매개변수로 속성을 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | getName() | String | Product 인터페이스 메서드 재정의 |
| | getPrice() | int | Product 인터페이스 메서드 재정의 |

· Customer 클래스 정보

| 클래스 | Customer | 패키지 | test10 |
|-----|--------------------------|--------|----------------|
| 속성 | 속성명 | 타입 | 설명 |
| | uid | String | 고객 아이디 |
| | name | String | 고객 이름 |
| 생성자 | uid, name을 매개변수로 속성을 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | getUid() | String | 속성 uid Getter |
| | getName() | String | 속성 name Getter |

· Order 클래스 정보

| 클래스 | Order | 패키지 | test10 |
|-----|---|---------------|----------------------|
| 속성 | 속성명 | 타입 | 설명 |
| | orderNo | int | 주문번호 |
| | customer | Customer | 주문 고객 정보 |
| | products | List<Product> | 주문 상품 내역 |
| | totalPrice | int | 전체 주문 가격 |
| 생성자 | - orderNo, customer를 매개변수로 속성을 초기화 - products는 ArrayList 생성 초기화 - totalPrice는 0 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | addProduct(Product) | void | 상품 추가 주문 |
| | getOrderNo() | int | 속성 orderNo Getter |
| | getCustomer() | Customer | 속성 customer Getter |
| | getProducts() | List<Product> | 속성 products Getter |
| | getTotalPrice() | int | 속성 totalPrice Getter |

· ShoppingMall 클래스 정보

| 클래스 | ShoppingMall | 패키지 | test10 |
|-----|---|------------------------------|----------|
| 속성 | 속성명 | 타입 | 설명 |
| | customers | List<Customer> | 주문번호 |
| | orders | List<Order> | 주문 고객 정보 |
| | customerOrders | HashMap<String, List<Order>> | 주문 상품 내역 |
| 생성자 | - customers, orders는 ArrayList 생성 초기화 - customerOrders는 HashMap 생성 초기화 | | |
| 메서드 | 메서드명 | 반환타입 | 설명 |
| | addCustomer(Customer) | void | 고객 정보 추가 |
| | addOrder(Order) | void | 주문 정보 추가 |
| | getOrders() | List<Order> | 주문 내역 반환 |

```

package test10;
/**
 * 날짜 : 2024/00/00
 * 이름 : 김철학
 * 내용 : 자바 총정리 문제
 */
public class Main {
    public static void main(String[] args) {

        // 쇼핑몰 생성
        ShoppingMall shoppingMall = new ShoppingMall();

        // 고객 생성
        Customer customer1 = new Customer("A101", "김유신");
        Customer customer2 = new Customer("A102", "김춘추");

        // 고객 추가
        shoppingMall.addCustomer(customer1);
        shoppingMall.addCustomer(customer2);

        // 제품 생성
        Product tshirt = new Clothes("티셔츠", 15000);
        Product orange = new Food("오렌지", 3000);
        Product banana = new Food("바나나", 3500);

        // 주문 생성
        Order order1 = new Order(1, customer1);
        order1.addProduct(orange);
        order1.addProduct(tshirt);

        Order order2 = new Order(2, customer2);
        order2.addProduct(tshirt);
        order2.addProduct(banana);

        // 주문 추가
        shoppingMall.addOrder(order1);
        shoppingMall.addOrder(order2);

        // 모든 주문 출력
        List<Order> orders = shoppingMall.getOrders();

        코드를 완성하기
    }
}

```

| 결과 |
|--|
| 주문번호 : 1 고객이름 : 김유신 주문금액 : ₩18000 주문목록 - 오렌지 ₩3000 - 티셔츠 ₩15000 주문번호 : 2 고객이름 : 김춘추 주문금액 : ₩18500 주문목록 - 티셔츠 ₩15000 - 바나나 ₩3500 |