

□ 응용 프로그래밍

- (1) **[생성형 AI 활용]** 제시된 main()과 결과를 참고하여 Book 객체의 동일성 여부(내용)를 판단하는 Book 클래스를 작성하세요. 단, equals() 메소드를 재정의하지 않으며 필드는 전용 멤버이어야 하고, 객체 내부에서 비교할 수 있도록 합니다

```
book1 => 책 제목: 1984, 저자: George Orwell
book2 => 책 제목: 1984, 저자: George Orwell
book1과 book2 필드 멤버값이 동일합니다.

book1 => 책 제목: 1984, 저자: George Orwell
book2 => 책 제목: 훈공자, 저자: 신용권
book1과 book2 필드 멤버 값이 서로 다릅니다.
```

```
public static void main(String[] args) {
    Book book1 = new Book("1984", "George Orwell"); //책 제목과 지은이를 매개변수로 전달합니다.
    Book book2 = new Book("1984", "George Orwell");
    Book book3 = new Book("훈공자", "신용권");
    isCheck(book1, book2);
    isCheck(book2, book3);
}
```

생성형 AI 플랫폼	
질문	
답변	
실행결과	

- (2) **[생성형 AI 활용]** 상속관계가 적용된 간단한 예를 제시하고, 어떤 경우에 상속 관계를 활용하는지 설명하세요

생성형 AI 플랫폼	
질문	
답변	
실행결과	

- (3) 키보드로 영 단어를 입력하면 단어 뜻을 한글로 출력하는 프로그램을 작성하세요. 영 단어와 한글은 객체 배열 생성시 초기화 합니다

```
class Word{
    //영어 단어 저장을 위한 필드, private
    //영 단어에 대한 뜻을 저장하는 필드, private
    //생성자 - 매개변수로 받은 값을 필드로 초기화
    //모든 필드에 대한 getter 메소드
    //매개변수로 받은 문자열이 현재 객체가 가지는 영 단어와 일치하는지 검사하여 true, false를 반환하는
    wordComp() 메소드
```

```
1 : 영단어와 뜻을 입력하세요 >>>>> one 하나
2 : 영단어와 뜻을 입력하세요 >>>>> student 학생
3 : 영단어와 뜻을 입력하세요 >>>>> apple 사과
4 : 영단어와 뜻을 입력하세요 >>>>> school 학교
5 : 영단어와 뜻을 입력하세요 >>>>> flower 꽃
6 : 영단어와 뜻을 입력하세요 >>>>> may 5월
7 : 영단어와 뜻을 입력하세요 >>>>> tree 나무
찾고자 하는 단어를 입력하세요. 종료(f) >>
school
school : 학교
찾고자 하는 단어를 입력하세요. 종료(f) >>
sign
해당 단어가 없습니다
찾고자 하는 단어를 입력하세요. 종료(f) >>
f
프로그램 종료
```

```

        //객체 내용을 문자열로 반환하는 toString() 메소드
    }

    class WordTest{
        public static void main(String[] args){
            //객체 배열을 선언 & 생성, 배열 크기는 7- 본인 작성
            //입력한 값으로 배열 초기화 - 본인 작성
            //wordComp() 메소드를 사용하여 객체 배열에 저장된 영 단어와 입력된 영단어와 일치하면 그에 해
            당하는 뜻을 출력 - 본인 작성
        }
    }

```

[프로그램 소스]

```

import java.util.*;
class Word {
    private String eng;
    private String kor;
    public Word(String eng, String kor) {
        this.eng = eng;
        this.kor = kor;
    }

    public String getKor() {
        return kor;
    }

    public String getEng() {
        return eng;
    }

    @Override
    public String toString() {
        return "Word [eng=" + eng + ", kor=" + kor + "]";
    }

    public boolean wordComp(String eng) {
        return this.eng.equals(eng);
    }
}

public class WordTest {
    public static void main(String[] args) {
        Word[] dim = new Word[7];
        Scanner in = new Scanner(System.in);

```

```

for (int i = 0; i < dim.length; i++) {
    System.out.print((i + 1) + " : 영단어와 뜻을 입력하세요 >>>>> ");
    dim[i] = new Word(in.next(), in.next());
}

while(true) {
    boolean flag=true;
    System.out.println("찾고자 하는 단어를 입력하세요. 종료(f) >> ");
    String search = in.next();
    if(search.equalsIgnoreCase("f"))
        break;

    for (Word temp : dim) {
        if (temp.wordComp(search)) {
            System.out.println(search + " : " + temp.getKor());
            flag=false;
            break;
        }
    }
    if(flag)
        System.out.println("해당 단어가 없습니다");
}
System.out.println("프로그램 종료");
}
}

```

[실행결과]

- (4) Person 클래스를 설계합니다. Person 클래스는 이름, 주소, 전화번호를 필드로 가집니다. 하나 이상의 생성자를 정의하고 각 필드에 대하여 접근자/설정자 메소드를 작성하세요. Person 클래스를 상속 받아 Customer 클래스를 작성합니다. Customer는 고객 번호와 마일리지를 필드로 가집니다. 한 개 이상의 생성자를 작성하고 접근자/설정자 메소드를 작성하세요. 두개의 클래스 모두 객체 정보를 문자열로 반환하는 toString() 메소드를 정의합니다. 테스트 클래스를 작성하여 위 클래스 타입의 객체를 만들고 각 객체의 모든 정보를 출력하세요.

```

public class PersonTest {
    public static void main(String[] args) {
        public static void main(String[] args) {
            //이름, 주소, 고객 번호, 마일리지
            Customer customer1=new Customer("hallym", "춘천시", "cn120", 1000);

            //이름, 주소, 전화번호,고객 번호, 마일리지

```

```
Customer customer2=new Customer("software", "화천군", "248-2328", "cn845", 3000);
```

```
//이름, 주소, 전화번호
```

```
Person person= new Person("computer", "횡성군", "240-5000");
```

```
System.out.println(customer1);
```

```
System.out.println(customer2);
```

```
System.out.println(person);
```

```
}
```

```
}
```

```
}
```

```
name=hallym, address=춘천시, phone=null  
customerNumber=cn120, mileage=1000
```

```
name=software, address=화천군, phone=248-2328  
customerNumber=cn845, mileage=3000
```

```
name=computer, address=횡성군, phone=240-5000
```

[프로그램 소스]

```
class Person {
```

```
    String name; //이름
```

```
    String address; //주소
```

```
    String phone; //전화 번호
```

```
    public Person() {
```

```
        //this(null, null, null);
```

```
    }
```

```
    public Person(String name, String address) {
```

```
        this(name, address, null);
```

```
    }
```

```
    public Person(String name, String address, String phone) {
```

```
        this.name = name;
```

```
        this.address = address;
```

```
        this.phone = phone;
```

```
    }
```

```
    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 String getPhone() {
```

```
        return phone;
```

```
    }
```

```
    public void setPhone(String phone) {
```

```
        this.phone = phone;
    }
    @Override
    public String toString() {
        return "name=" + name + ", address=" + address + ", phone=" + phone + "\n";
    }
}
```

```
class Customer extends Person {
    String customerNumber; //고객번호
    int mileage; //마일리지

    public Customer() {
        super();
    }

    public Customer(String name, String address, String phone) {
        this(name, address, phone, null, 0);
    }

    public Customer(String name, String address, String customerNumber, int mileage) {
        this(name, address, null, customerNumber, mileage);
    }

    public Customer(String name, String address, String phone, String customerNumber, int mileage) {
        super(name, address, phone);
        this.customerNumber = customerNumber;
        this.mileage = mileage;
    }

    public String getCustomerNumber() {
        return customerNumber;
    }

    public void setCustomerNumber(String customerNumber) {
        this.customerNumber = customerNumber;
    }

    public int getMileage() {
        return mileage;
    }

    public void setMileage(int mileage) {
        this.mileage = mileage;
    }

    @Override
    public String toString() {
```

```

String str = super.toString();
return str+"customerNumber=" + customerNumber + ", mileage=" + mileage + "\n";
}
}

public class Exam_01 {
    public static void main(String[] args) {
        //이름, 주소, 고객번호, 마일리지
        Customer customer1=new Customer("hallym", "춘천시", "cn120", 1000);

        //이름, 주소, 전화번호,고객 번호, 마일리지
        Customer customer2=new Customer("software", "화천군", "248-2328", "cn845", 3000);

        //이름, 주소, 전화번호
        Person person= new Person("computer", "횡성군", "240-5000");
        System.out.println(customer1);
        System.out.println(customer2);
        System.out.println(person);
    }
}

```

[실행 결과]

- (5) 일반적인 책을 나타내는 Book 클래스를 상속받아서 잡지를 나타내는 Magazine 클래스를 작성합니다. Book 클래스는 제목, 페이지수, 저자 등의 정보를 가집니다. Magazine 클래스는 추가로 발매일 정보를 가집니다. 생성자, 접근자, 설정자를 포함하여서 각각의 클래스를 작성하세요. 이들 클래스들의 객체를 만들고 각 객체의 모든 정보를 출력하는 테스트 클래스를 작성하세요.

```

책 이름 : 잡지A
페이지 수 : 10
저자 : 기자A
발매일 :2010년 2월 25일

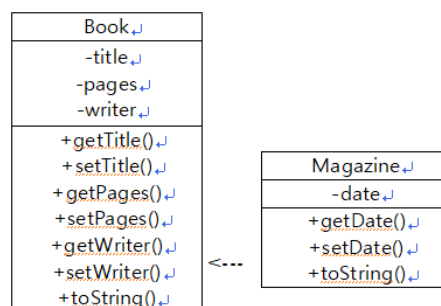
----bookA 객체 저자 변경 후---
책 이름 : 잡지A
페이지 수 : 10
저자 : 기자변경
발매일 :2010년 2월 25일

책 이름 : 잡지B
페이지 수 : 20
저자 : 기자B
발매일 :2010년 3월 8일

----bookB 객체 페이지 변경 후---
책 이름 : 잡지B
페이지 수 : 50
저자 : 기자B
발매일 :2010년 3월 8일

----bookB 객체 발매일 변경 후---
책 이름 : 잡지B
페이지 수 : 50
저자 : 기자B
발매일 :2020년 8월 31일

```



```

public class Test {

```

```
public static void main(String[] args) {  
    Magazine bookA = new Magazine("잡지A", 10, "기자A", "2010년 2월 25일");  
    Magazine bookB = new Magazine("잡지B", 20, "기자B", "2010년 3월 8일");  
    //bookB의 타입을 다음과 같이 Book으로 선언 하였을 때 발생하는 문제는?  
    //Book bookB = new Magazine("잡지B", 20, "기자B", "2010년 3월 8일");  
    //제시된 결과처럼 출력될 수 있도록 완성 – 본인작성  
}  
}
```

[프로그램 소스]

```
class Book {  
    private String title;  
    private int pages;  
    private String writer;  
  
    public Book(String title, int pages, String writer) {  
        this.title = title;  
        this.pages = pages;  
        this.writer = writer;  
    }  
  
    public String getTitle() {  
        return title;  
    }  
  
    public void setTitle(String title) {  
        this.title = title;  
    }  
  
    public int getPages() {  
        return pages;  
    }  
  
    public void setPages(int pages) {  
        this.pages = pages;  
    }  
  
    public String getWriter() {  
        return writer;  
    }  
  
    public void setWriter(String writer) {  
        this.writer = writer;  
    }  
}
```

```
    public String toString() {  
        return "책 이름 : " + title + "\n페이지 수 : " + pages + "\n저자 : " + writer;  
    }  
}
```

```
class Magazine extends Book {  
    private String date;  
  
    public Magazine(String title, int pages, String writer, String date) {  
        super(title, pages, writer);  
        this.date = date;  
    }  
  
    public String getDate() {  
        return date;  
    }  
  
    public void setDate(String date) {  
        this.date = date;  
    }  
  
    public String toString() {  
        return super.toString() + "\n발매일 : " + date;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        Magazine bookA = new Magazine("잡지A", 10, "기자A", "2010년 2월 25일");  
        Magazine bookB = new Magazine("잡지B", 20, "기자B", "2010년 3월 8일");  
  
        System.out.println(bookA);  
        System.out.println("\n----bookA 객체 저자 변경 후---");  
        bookA.setWriter("기자변경");  
        System.out.println(bookA);  
        System.out.println();  
  
        System.out.println(bookB);  
        System.out.println("\n----bookB 객체 페이지 변경 후---");  
        bookB.setPages(50);  
        System.out.println(bookB);  
        System.out.println("\n----bookB 객체 발매일 변경 후---");  
        bookB.setDate("2020년 8월 31일");  
    }  
}
```

```

        System.out.println(bookB);
    }
}

```

[실행 결과]

- (6) 일반적인 음식을 나타내는 Food 클래스를 상속받아서 멜론을 나타내는 Melon 클래스를 작성합니다. Food 클래스는 칼로리, 가격, 중량 등의 정보를 가집니다. Melon 클래스는 추가로 경작농원정보를 가집니다. 생성자, 접근자, 설정자를 포함하여서 각각의 클래스를 작성합니다. 이들 클래스들의 객체를 만들고 각 객체의 모든 정보를 출력하는 테스트 클래스도 작성합니다.

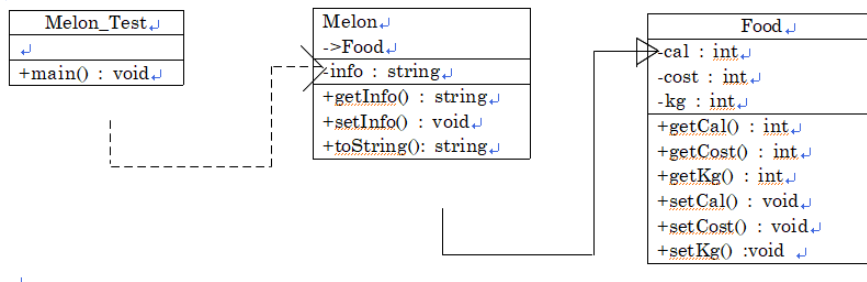
```

m1 객체 정보 [ cal = 1200   cost=2500   kg=21   경작 농원=Hallym_farm ]
m2 객체 정보 [ cal = 1000   cost=2000   kg=10   경작 농원=Software_farm ]

>> m2 객체의 cost와 경작농원 정보 변경 후
cal = 1000   cost=3000   kg=10   경작 농원=Java_farm

m1 : 중량 >> 21
프로그램 종료

```



[프로그램 소스]

```

class Food {
    private int cal; //칼로리
    private int cost; // 가격
    private int kg; // 중량

    public Food(int cal, int cost, int kg) { // 생성자 매개변수 존재
        this.cal= cal;
        this.cost = cost;
        this.kg = kg;
    }

    public Food() { // 생성자
        this(0, 0, 0);
    }

    public int getCal() {
        return cal;
    }
}

```

```
public void setCal(int cal) {
    this.cal = cal;
}

public int getCost() {
    return cost;
}

public void setCost(int cost) {
    this.cost = cost;
}

public int getKg() {
    return kg;
}

public void setKg(int kg) {
    this.kg = kg;
}

@Override
public String toString() {
    return "cal = " + cal + "    cost=" + cost + "    kg=" + kg;
}
}
```

```
class Melon extends Food { // melon 클래스 작성 Food 상속
    private String info; // 경작 농원 정보

    public Melon(int cal, int cost, int kg, String info) {
        super(cal, cost, kg); // Food 생성자 호출
        this.info = info;
    }

    public Melon() {
        this(0,0,0,null);
    }

    public void setInfo(String info) { // 설정자.
        this.info = info;
    }

    public String getInfo() { // 접근자.
        return info;
    }
}
```

```

    }

    public String toString() {
        String str = super.toString();
        str += "    경작 농원=" + this.info;
        return str;
    }
}

public class Test{
    public static void main(String[] args) {
        Melon m1 = new Melon(1200, 2500, 21, "Hallym_farm");
        Melon m2 = new Melon(1000, 2000, 10, "Software_farm");

        System.out.println("m1 객체 정보 [ " + m1 + " ]");
        System.out.println("m2 객체 정보 [ " + m2 + " ]\n");

        System.out.println(">> m2 객체의 cost와 경작농원 정보 변경 후");
        m2.setCost(3000);
        m2.setInfo("Java_farm");
        System.out.println(m2 + "\n");

        System.out.println("m1 : 중량 >> " + m1.getKg());
        System.out.println("프로그램 종료");
    }
}

```

[실행 결과]

- (7) 생성형 AI가 제시하는 상속을 활용하는 문제를 프로그램하고 결과를 제시 하세요. 문제 해결을 위한 코드는 답변에서 제외 되어야 합니다

생성형 AI 플랫폼	
질문	
답변	

[프로그램 소스]

[실행 결과]
