❑ 응용 프로그래밍

(1) **[생성형 AI 활용]** 객체의 집합 관계와 사용관계, 정적메소드를 모두 활용하는 예를 제시하세요

생성형 AI 플랫폼	
질문	
답변	
실행결과	

(2) **[생성형 AI 활용]** 단 한 개의 게임아이템을 생성하는 클래스를 작성하고테스트 하세요

생성형 AI 플랫폼	
질문	
답변	
실행결과	

(3) 신발 매장에서 상품관리를 위한 클래스를 작성하고 테스트하시오

```
public static void main(String[] args) {
        Shoes shoes1=new Shoes(255, 3, "샌들" );
        Shoes shoes2=new Shoes(235, 5, "운동화" );

        System.out.printf("shoes1 : %s₩n",shoes1);
        System.out.printf("shoes2 : %s₩n₩n",shoes2);

        System.out.println("샌들 다섯개를 판매합니다");
        if(shoes1.sell(5))
                System.out.println("현재 샌들의 개수는 " + shoes1.getCnt() + " 입니다");
        else
                System.out.println("수량부족입니다");

        System.out.println("운동화 2 개를 추가합니다");
        shoes2.add(2);
        System.out.println("현재 운동화 개수는 " + shoes2.getCnt() + " 입니다");
}
```

```
shoes1 : 신발 [종류=샌들, 사이즈=255, 재고=3]
shoes2 : 신발 [종류=운동화, 사이즈=235, 재고=5]

샌들 다섯개를 판매합니다
수량부족입니다
운동화 2개를 추가합니다
현재 운동화 개수는 7 입니다
```

[프로그램 소스]

```
class Shoes {
    private int size, cnt;
    private String type;

    public Shoes(int size, int cnt, String type) {
```

```
        this.size = size;
        this.cnt = Math.max(cnt, 0);
        this.type = (type != null) ? type : "알 수 없음";
    }

    public int getSize() {
        return size;
    }

    public int getCnt() {
        return cnt;
    }

    public String getType() {
        return type;
    }

    public void add(int cnt) {
        if (cnt > 0) {
            this.cnt += cnt;
        }
    }

    public boolean sell(int cnt) {
        if (cnt <= 0 || this.cnt < cnt) {
            return false; // 판매 실패
        }
        this.cnt -= cnt;
        return true; // 판매 성공
    }

    @Override
    public String toString() {
        return "신발 [종류=" + type + ", 사이즈=" + size + ", 재고=" + cnt + "]";
    }
}

public class week03_app03 {
        public static void main(String[] args) {
                Shoes shoes1=new Shoes(255, 3, "샌들" );
                Shoes shoes2=new Shoes(235, 5, "운동화" );

                System.out.printf("shoes1 : %s₩n",shoes1);
                System.out.printf("shoes2 : %s₩n₩n",shoes2);
```

```
                System.out.println("샌들 다섯개를 판매합니다");
                if(shoes1.sell(5))
                        System.out.println("현재 샌들의 개수는 " + shoes1.getCnt() + " 입니다");
                else
                        System.out.println("수량부족입니다");

                System.out.println("운동화 2 개를 추가합니다");
                shoes2.add(2);
                System.out.println("현재 운동화 개수는 " + shoes2.getCnt() + " 입니다");
        }
}
```

[실행 결과]

(4) 날짜를 나타내는 클래스 Date를 만들어보자. Date는 연도, 월, 일 등의 속성을 가지며, 날짜를 "2012.7.12"과 같이 출력하는 메소드 print1(), 날짜를 "July 12, 2012"와 같이 출력하는 print2() 등의 메소드를 갖는다. 회원 정보를 나타내는 Member 클래스를 만들어 보자. 회원 번호(int, 두 자리 난수)와 이름(String), 생년월일(Date 클래스) 속성을 가지며 객체 정보를 문자열로 반환하는 toString() 메소드를 갖는다. 테스트 하고 실행 하시오

```
public class Test {
    public static void main(String[] args) {
        // 회원 객체 생성 (이름과 생년월일 입력)
        Member member1 = new Member("김철수", 1995, 5, 20);
        Member member2 = new Member("이영희", 2002, 11, 15);

        // 회원 정보 출력 (toString() 메서드 자동 호출)
        System.out.println(member1);
        System.out.println();
        System.out.println(member2);

        // 날짜 출력 테스트
        System.out.println("₩n 날짜 출력 테스트:");
        Date date = new Date(2024, 3, 18);
        date.print1(); // 2024.3.18
        date.print2(); // March 18, 2024
    }
}
```

```
회원 번호: 53
이름: 김철수
생년월일: 1995/5/20

회원 번호: 96
이름: 이영희
생년월일: 2002/11/15

날짜 출력 테스트:
2024.3.18
March 18, 2024
```

[프로그램 소스]

```
class Date{
        private int year, month, date;
```

```
	Date(int year,int month,int date){
		this.year=year;
		this.month=month;
		this.date=date;
	}

	void print1() {
		System.out.println(year+"."+month+"."+date);
	}

	void print2() {
		String[] strMonth= {"January", "February", "March", "April", "May", "June",
"July","August","September","October","November","December"};
		System.out.println(strMonth[month-1]+" "+date+", "+year);
	}
	public String toString() {
		return year+"/"+month+"/"+date;
	}
}
class Member {
    private int memberId; // 회원 번호 (두 자리 난수)
    private String name;  // 회원 이름
    private Date birthDate; // 생년월일 (Date 클래스 사용)

    public Member(String name, int year, int month, int day) {
        this.memberId = (int)(Math.random()*90)+10; // 10~99 사이 난수 생성
        this.name = name;
        this.birthDate = new Date(year, month, day);
    }

    // 객체 정보를 문자열로 반환
    @Override
    public String toString() {
        return "회원 번호: " + memberId + "₩n 이름: " + name + "₩n 생년월일: " + birthDate;
    }
}
public class Test {
    public static void main(String[] args) {
        // 회원 객체 생성 (이름과 생년월일 입력)
        Member member1 = new Member("김철수", 1995, 5, 20);
        Member member2 = new Member("이영희", 2002, 11, 15);

        // 회원 정보 출력 (toString() 메서드 자동 호출)
        System.out.println(member1);
```

```
            System.out.println();
            System.out.println(member2);

            // 날짜 출력 테스트
            System.out.println("₩n 날짜 출력 테스트:");
            Date date = new Date(2024, 3, 18);
            date.print1(); // 2024.3.18
            date.print2(); // March 18, 2024
        }
    }
```

[실행 결과]

(5) 학생을 나타내는 클래스 Student를 만들어보자. 학생은 이름(private name)과 점수(private score), 등급(private grade)을 가진다. 객체 내용(이름, 점수, 등급)을 문자열로 반환하는 메소드 toString()를 가진다. 이름과 점수를 전달하는 생성자도 필요하다. 점수를 변경하는 메소드와 등급과 점수를 반환하는 메소드도 필요하다. 제시한 점수 이상의 학생을 검색하는 정적 메소드와 특정 등급의 학생만 검색하는 정적 메소드를 멤버로 갖는 클래스를 작성한다. 필드는 외부에 공개하지 않는다

```
public class Test {
    public static void main(String[] args) {
        // 1. 이름 없는 학생 객체 생성 후 점수 설정
        Student student1 = new Student("둘리");
        student1.setScore(85);

        // 2. 이름과 점수를 전달한 학생 객체 생성
        Student student2 = new Student("김철수", 92);

        // 3. 배열에 5 명의 학생 저장
        Student[] students = { student1, student2, new Student("이영희", 76), new Student("박지훈", 89),
                               new Student("최민수", 95) };

        // 4. 모든 학생 정보 출력
        System.out.println("₩n=== 학생 목록 ===");
        for (Student student : students) {
            System.out.println(student);
        }

        // 5. 등급과 성적으로 학생 정보 출력
        StudentControl.searchGrade(students, 'B');
        StudentControl.searchScore(students, 86);
    }
}
```

```
=== 학생 목록 ===
이름: 둘리, 점수: 85, 등급: B
이름: 김철수, 점수: 92, 등급: A
이름: 이영희, 점수: 76, 등급: C
이름: 박지훈, 점수: 89, 등급: B
이름: 최민수, 점수: 95, 등급: A

B 등급 학생 목록:
이름: 둘리, 점수: 85, 등급: B
이름: 박지훈, 점수: 89, 등급: B

86 이상 학생 목록:
이름: 김철수, 점수: 92, 등급: A
이름: 박지훈, 점수: 89, 등급: B
이름: 최민수, 점수: 95, 등급: A
```

[프로그램 소스]

```
//학생을 나타내는 클래스
class Student {
	private String name;
	private int score;
	private char grade;

	// 이름 없는 학생 생성자 (점수는 나중에 설정)
	public Student(String name) {
		this(name, 0);
	}

	// 이름과 점수를 받는 생성자
	public Student(String name, int score) {
		this.name = name;
		setScore(score); // 점수 설정 시 등급도 자동 계산
	}

	// 점수 설정 및 등급 자동 계산
	public void setScore(int score) {
		this.score = score;
		this.grade = calculateGrade(score);
	}

	public char getGrade() {
		return grade;
	}

	public int getScore() {
		return score;
	}

	// 등급 계산 메서드
	private char calculateGrade(int score) {
		if (score >= 90)
			return 'A';
		if (score >= 80)
			return 'B';
		if (score >= 70)
			return 'C';
		if (score >= 60)
			return 'D';
		return 'F';
	}
```

```
	@Override
	public String toString() {
		return "이름: " + name + ", 점수: " + score + ", 등급: " + grade;
	}
}

//등급이 B 이상인 학생을 출력하는 유틸리티 클래스
class StudentControl {
	public static void searchGrade(Student[] students, char grade) {
		System.out.printf("₩n%c 등급 학생 목록:₩n", grade);
		for (Student student : students) {
			if (student.getGrade() == grade) {
				System.out.println(student);
			}
		}
	}

	public static void searchScore(Student[] students, int score) {
		System.out.printf("₩n%d 이상 학생 목록:₩n",score);
		for (Student student : students) {
			if (student.getScore() >= score) {
				System.out.println(student);
			}
		}
	}
}

//테스트 클래스
public class Test {
	public static void main(String[] args) {
		// 1. 이름 없는 학생 객체 생성 후 점수 설정
		Student student1 = new Student("둘리");
		student1.setScore(85);

		// 2. 이름과 점수를 전달한 학생 객체 생성
		Student student2 = new Student("김철수", 92);

		// 3. 배열에 5 명의 학생 저장
		Student[] students = { student1, student2, new Student("이영희", 76), new Student("박지훈", 89),
				new Student("최민수", 95) };

		// 4. 모든 학생 정보 출력
		System.out.println("₩n=== 학생 목록 ===");
		for (Student student : students) {
```

```
                System.out.println(student);
            }

            StudentControl.searchGrade(students, 'B');
            StudentControl.searchScore(students, 86);

        }
}
```

[실행 결과]

(6) 아이디는 키보드로 입력 받으며, 비밀번호는 아이디+객체생성순서+2자리수 정수형 난수를 연결하여 초기화하는 Info 클래스를 제시된 조건대로 작성하고 테스트 하세요.

▪ 필드 구성 – 외부에 공개하지 않는다
- id : String, 아이디 저장
- pass : String, 비밀번호 저장

▪ 생성자 : 아이디 필드는 매개변수로 받은 값으로 초기화하고, 비밀번호는 아이디 +객체 생성순서 + 두자리 난수로 초기화

▪ 메소드 구성
- 객체 내용을 문자열로 반환하는 toString()
- 필드 값을 변경하고 반환하는 getter, setter()

Info 객체를 생성하고 테스트하는 InfoTest 클래스를 작성 하시오.

▪ main() 메소드
- Info 객체 두개를 선언하고 아이디는 입력 받아서 생성자 매개변수로 전달
- 두 개의 객체 내용 출력
- 첫번째 객체의 비밀번호를 난수를 사용하여 변경한 후 출력

```
아이디를 입력 하세요 >>> hallym
첫번째 객체 생성 완료
아이디를 입력 하세요 >>> software
두번째 객체 생성 완료

첫번째 객체의 아이디와 비밀번호 출력
Info [아이디=hallym, 비밀번호=hallym113]

두번째 객체의 아이디와 비밀번호 출력
Info [아이디=software, 비밀번호=software210]
₩n첫번째 객체의 비밀번호 변경 후 출력
Info [아이디=hallym, 비밀번호=hallym131]
```

[소스]

```
import java.util.*;

class Info {
```

```
	private String id;
	private String pass;
	private static int num=0;
	private int beon;

	Info(String id) {
		this.id = id;
		int rnd= (int) (Math.random() * 90) + 10;
		beon=++num;
		pass =id+beon+rnd;
	}

	public String getId() {
		return id;
	}

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

	public String getPass() {
		return pass;
	}

	public void setPass(int rnd) {
		this.pass = id+beon+rnd;
	}

	@Override
	public String toString() {
		return "Info [아이디=" + id + ", 비밀번호=" + pass + "]";
	}
}

public class sol3 {
	public static void main(String[] args) {
		Scanner key = new Scanner(System.in);
		System.out.print("아이디를 입력 하세요 >>> ");
		Info obj1 = new Info(key.next());
		System.out.println("첫번째 객체 생성 완료");
		System.out.print("아이디를 입력 하세요 >>> ");
		Info obj2 = new Info(key.next());
		System.out.println("두번째 객체 생성 완료");
```

```
                System.out.println("₩n 첫번째 객체의 아이디와 비밀번호 출력");
                System.out.println(obj1);

                System.out.println("₩n 두번째 객체의 아이디와 비밀번호 출력");
                System.out.println(obj2);


                obj1.setPass((int) (Math.random() * 90) + 10);
                System.out.println("₩₩n 첫번째 객체의 비밀번호 변경 후 출력");
                System.out.println(obj1);
        }
}
```

[실행 결과]

(7) **[생성형 AI 활용]** AI가 제시하는 객체 배열 활용 문제를 프로그램하고 결과를 제시하세요. 문제 해결을 위한 코드는 답변에서 제외 되어야 합니다

생성형 AI 플랫폼	
질문	
답변	

[프로그램 소스]

[실행 결과]