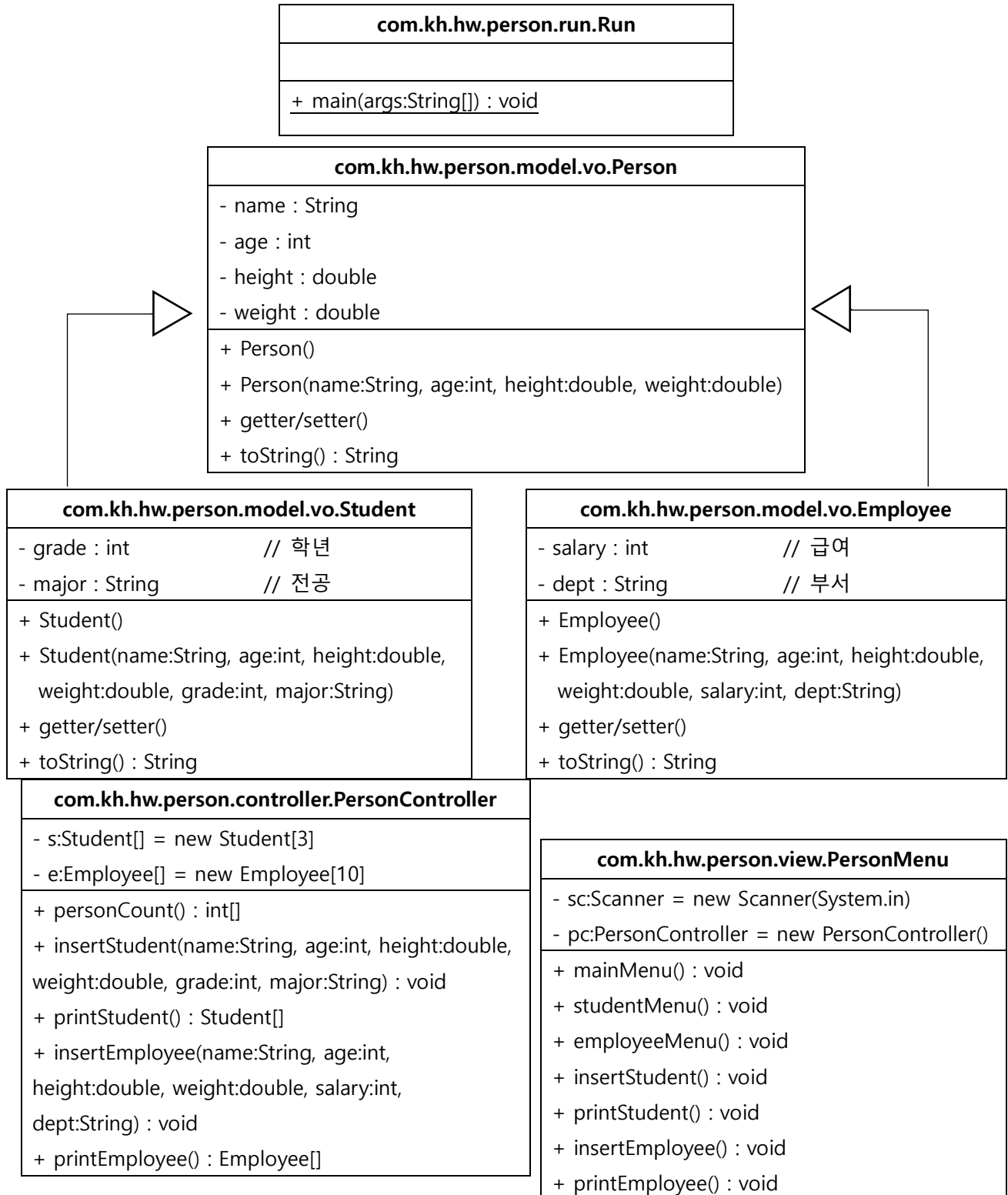


* 프로젝트 명 : 08_InheritPractice

[문제 1] 다음과 같은 조건을 만족하는 프로그램을 작성 하시오

1. 구현 클래스 다이어그램 (Class Diagram)



* 각 클래스의 setter/getter 메소드는 직접 구현한다.

2. 구현 클래스 설명

| Package명 | Class명 | Method | 설명 |
|-----------------------------------|------------------|--|---|
| com.kh.hw. person.model. vo | Person | +Person() | 기본 생성자 |
| | | +Person(name:String, age:int, height:double, weight:double) | 매개변수 생성자 |
| | | +getter/setter() | 필드 값을 반환하고 대입하는 메소드 |
| | | +toString():String | 각 필드 반환 메소드 |
| | Student | +Student() | 기본 생성자 |
| | | +Student(name:String, age:int, height:double, weight:double, grade:int, major:String) | 매개변수 생성자 |
| | | +getter/setter() | 필드 값을 반환하고 대입하는 메소드 |
| | | +toString():String | 각 필드 반환 메소드 |
| | Employee | +Employee() | 기본 생성자 |
| | | +Employee(name:String, age:int, height:double, weight:double, salary:int, dept:String) | 매개변수 생성자 |
| | | +getter/setter() | 필드 값을 반환하고 대입하는 메소드 |
| | | +toString():String | 각 필드 반환 메소드 |
| com.kh.hw. person.view | PersonMenu | +mainMenu():void | 메인 메뉴를 출력하는 메소드 |
| | | +studentMenu():void | 학생 메뉴를 출력하는 메소드 |
| | | +employeeMenu():void | 사원 메뉴를 출력하는 메소드 |
| | | +insertStudent():void | 사용자에게 객체배열에 저장할 학생 데이터를 받는 메소드 |
| | | +printStudent():void | 객체배열에 저장된 학생 데이터를 출력하는 메소드 |
| | | +insertEmployee():void | 사용자에게 객체배열에 저장할 사원 데이터를 받는 메소드 |
| | | +printEmployee():void | 객체배열에 저장된 사원 데이터를 출력하는 메소드 |
| com.kh.hw.person.controller | PersonController | +personCount():int[] | 각 객체배열에 저장된 객체의 수를 정수배열에 담아 반환하는 메소드 |
| | | +insertStudent(name:String, age:int, height:double, weight:double, grade:int, major:String):void | 매개변수로 받아온 데이터를 학생 객체 배열 중 빈 곳에 저장하는 메소드 |
| | | +printStudent():Student[] | 학생 객체 배열의 주소를 반환하는 메소드 |

| | | | |
|--------------------------|-----|---|---|
| | | +insertEmployee(name:String, age:int, height:double, weight:double, salary:int, dept:String):void | 매개변수로 받아온 데이터를 사원 객체 배열 중 빈 곳에 저장되는 메소드 |
| | | +printEmployee():Employee[] | 사원 객체 배열의 주소를 반환하는 메소드 |
| com.kh.hw. person.run | Run | <u>+main(args:String[]) : void</u> | PersonMenu클래스의 mainMenu()메소드를 호출 |

3. 사용 데이터

| 이름 | 나이 | 신장 | 몸무게 | 학년 | 전공 |
|-----|----|-------|------|-----------|----------|
| 강건강 | 20 | 178.2 | 70.0 | 1 | 정보시스템공학과 |
| 남나눔 | 21 | 187.3 | 80.0 | 2 | 경영학과 |
| 류라라 | 23 | 167.0 | 45.0 | 4 | 정보통신공학과 |
| 이름 | 나이 | 신장 | 몸무게 | 급여 | 부서 |
| 박보배 | 26 | 180.3 | 72.0 | 100000000 | 영업부 |
| 송성실 | 38 | 182.0 | 76.0 | 200000000 | 기획부 |

4. class 구조

```

public class Run{
    public static void main(String[] args) {
        new PersonMenu().mainMenu();
    }
}

```

```

public class PersonMenu{
    // 필드

    public void mainMenu() {
        학생은 최대 3명까지 저장할 수 있습니다.
        현재 저장된 학생은 M명입니다.
        사원은 최대 10명까지 저장할 수 있습니다.
        현재 저장된 사원은 N명입니다.
        // M과 N에 들어가는 숫자는 PersonController(pc)클래스에 있는
        // personCount()메소드의 반환 값을 이용하여 출력

        1. 학생 메뉴      ➔ studentMenu()
        2. 사원 메뉴      ➔ employeeMenu()
        9. 끝내기        ➔ "종료합니다." 출력 후 종료
        메뉴 번호 :
        // 잘못 입력했을 경우 "잘못 입력하셨습니다. 다시 입력해주세요." 출력 후 반복
    }

    public void studentMenu(){
        1. 학생 추가      ➔ insertStudent()
        2. 학생 보기      ➔ printStudent()
        9. 메인으로      ➔ "메인으로 돌아갑니다." 출력 후 메인으로
        // 만일 학생 객체 배열에 담긴 데이터의 수가 3개 일 때
        // "학생을 담을 수 있는 공간이 꽉 찼기 때문에 학생 추가 메뉴는 더 이상
        // 활성화 되지 않습니다."를 출력 후 메뉴 번호를 다음
        // 실제로 1번을 누르면 "잘못 입력하셨습니다. 다시 입력해주세요."를 출력

        // 학생 객체 배열에 담긴 데이터의 수가 3개가 아닐 때는
        // 위에 출력한 메뉴 모두 받을 수 있게 함
        메뉴 번호 :
        // 잘못 입력했을 경우, "잘못 입력하셨습니다. 다시 입력해주세요." 출력 후 반복
    }
}

```

```

public void employeeMenu(){
    1. 사원 추가    ➔ insertEmployee()
    2. 사원 보기    ➔ printEmployee()
    9. 메인으로    ➔ "메인으로 돌아갑니다." 출력 후 메인으로
    // 만일 사원 객체 배열에 담긴 데이터의 수가 10개 일 때
    // "사원을 담을 수 있는 공간이 꽉 찼기 때문에 사원 추가 메뉴는 더 이상
    // 활성화 되지 않습니다."를 출력 후 메뉴 번호를 다음
    // 실제로 1번을 누르면 "잘못 입력하셨습니다. 다시 입력해주세요."를 출력

    // 사원 객체 배열에 담긴 데이터의 수가 10개가 아닐 때는
    // 위에 출력한 메뉴 모두 받을 수 있게 함
    메뉴 번호 :
    // 잘못 입력했을 경우, "잘못 입력하셨습니다. 다시 입력해주세요." 출력 후 반복
}

public void insertStudent(){
    학생 이름 :
    학생 나이 :
    학생 키 :
    학생 몸무게 :
    학생 학년 :
    학생 전공 :
    // 위의 데이터를 모두 사용자에게 입력 받아 pc의 insertStudent()메소드의
    // 매개변수로 넘겨 줌
    // 이 때 학생 객체 배열에 담긴 데이터의 수가 3개가 되지 않았을 때는
    // "그만하시려면 N(또는 n), 이어하시려면 아무 키나 누르세요 : "를 출력해
    // N 또는 n이 아니면 위의 데이터를 받는 것을 반복하고
    // N 또는 n이라면 반복을 멈춤
    // 만일 학생 객체 배열에 담긴 데이터의 수가 3이 되었을 때는
    // "학생을 담을 수 있는 공간이 꽉 찼기 때문에 학생 추가를 종료하고
    // 학생 메뉴로 돌아갑니다."를 출력 후 반복 종료
}

```

```

public void printStudent(){
    // pc의 printStudent() 메소드의 반환 값을 이용하여 학생 객체 배열에 저장된
    // 모든 데이터 출력
}

public void insertEmployee(){
    사원 이름 :
    사원 나이 :
    사원 키 :
    사원 몸무게 :
    사원 급여 :
    사원 부서 :
    // 위의 데이터를 모두 사용자에게 입력 받아 pc의 insertEmployee()메소드의
    // 매개변수로 넘겨 줌
    // 이 때 사원 객체 배열에 담긴 데이터의 수가 10개가 되지 않았을 때는
    // "그만하시려면 N(또는 n), 이어하시려면 아무 키나 누르세요 :"를 출력해
    // N 또는 n이 아니면 위의 데이터를 받는 것을 반복하고
    // N 또는 n이라면 반복을 멈춤
    // 만일 사원 객체 배열에 담긴 데이터의 수가 10이 되었을 때는
    // "사원을 담을 수 있는 공간이 꽉 찼기 때문에 사원 추가를 종료하고
    // 사원 메뉴로 돌아갑니다."를 출력 후 반복 종료
}

public void printEmployee(){
    // pc의 printEmployee() 메소드의 반환 값을 이용하여 사원 객체 배열에 저장된
    // 모든 데이터 출력
}
}

```

5. 실행 결과

학생은 최대 3명까지 저장할 수 있습니다.

현재 저장된 학생은 0명입니다.

사원은 최대 10명까지 저장할 수 있습니다.

현재 저장된 사원은 0명입니다.

1. 학생 메뉴

2. 사원 메뉴

9. 끝내기

메뉴 번호 : 1

1. 학생 추가

2. 학생 보기

9. 메인으로

메뉴 번호 : 1

학생 이름 : 강건강

학생 나이 : 20

학생 키 : 178.2

학생 몸무게 : 70.0

학생 학년 : 1

학생 전공 : 정보시스템공학과

그만 하시려면 N(또는 n), 이어하시려면 아무 키나 누르세요 : o

학생 이름 : 남나눔

학생 나이 : 21

학생 키 : 187.3

학생 몸무게 : 80.0

학생 학년 : 2

학생 전공 : 경영학과

그만 하시려면 N(또는 n), 이어하시려면 아무 키나 누르세요 : y

학생 이름 : 류라라

학생 나이 : 23

학생 키 : 167

학생 몸무게 : 45

학생 학년 : 4

학생 전공 : 정보통신공학과

학생을 담을 수 있는 공간이 꽉 찼기 때문에 학생 추가를 종료하고 학생메뉴로 돌아갑니다.

1. 학생 추가

2. 학생 보기

9. 메인으로

학생을 담을 수 있는 공간이 꽉 찼기 때문에 학생 추가 메뉴는 더 이상 활성화 되지 않습니다.

메뉴 번호 : 1

잘못 입력하셨습니다. 다시 입력해주세요.

- 1. 학생 추가
- 2. 학생 보기
- 9. 메인으로

학생을 담을 수 있는 공간이 꽉 찼기 때문에 학생 추가 메뉴는 더 이상 활성화 되지 않습니다.

메뉴 번호 : 2

강건강, 20, 178.2, 70.0, 1, 정보시스템공학과
남나눔, 21, 187.3, 80.0, 2, 경영학과
류라라, 23, 167.0, 45.0, 4, 정보통신공학과

- 1. 학생 추가
- 2. 학생 보기
- 9. 메인으로

학생을 담을 수 있는 공간이 꽉 찼기 때문에 학생 추가 메뉴는 더 이상 활성화 되지 않습니다.

메뉴 번호 : 9

메인으로 돌아갑니다.

학생은 최대 3명까지 저장할 수 있습니다.
현재 저장된 학생은 3명입니다.
사원은 최대 10명까지 저장할 수 있습니다.
현재 저장된 사원은 0명입니다.

- 1. 학생 메뉴
- 2. 사원 메뉴
- 9. 끝내기

메뉴 번호 : 2

- 1. 사원 추가
- 2. 사원 보기
- 9. 메인으로

메뉴 번호 : 1

사원 이름 : 박보배

사원 나이 : 26

사원 키 : 180.3

사원 몸무게 : 72

사원 급여 : 100000000

사원 부서 : 영업부

그만 하시려면 N(또는 n), 이어하시려면 아무 키나 누르세요 : ↵

사원 이름 : 송성실

사원 나이 : 28

사원 키 : 182

사원 몸무게 : 76

사원 급여 : 200000000

사원 부서 : 기획부

그만 하시려면 N(또는 n), 이어하시려면 아무 키나 누르세요 : n

- 1. 사원 추가
- 2. 사원 보기
- 9. 메인으로

메뉴 번호 : 2

박보배, 26, 180.3, 72.0

송성실, 28, 182.0, 76.0

- 1. 사원 추가
- 2. 사원 보기
- 9. 메인으로

메뉴 번호 : 9

메인으로 돌아갑니다.

학생은 최대 3명까지 저장할 수 있습니다.

현재 저장된 학생은 3명입니다.

사원은 최대 10명까지 저장할 수 있습니다.

현재 저장된 사원은 2명입니다.

- 1. 학생 메뉴
- 2. 사원 메뉴
- 9. 끝내기

메뉴 번호 : 9

종료합니다.