

과목 : 객체지향 프로그래밍(가반)

교수 : 최지웅 교수

이름 : 김병준

학번 : 20162448

과제 #3

2	4	6	8	10	12
0	0	0	0	0	0

1. (2번) Grade.java – 3과목 점수의 평균 출력

● Source

```
import java.util.InputMismatchException;
import java.util.Scanner;
public class Grade {
    private static int math;
    private static int science;
    private static int english;
    private Grade(int math, int science, int english) {
        Grade.math = math;
        Grade.science = science;
        Grade.english = english;
    }
    private int average() {
        return (math + science + english) / 3;
    }
    public static void main(String args[]) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("수학, 과학, 영어 순으로 3 개의 점수 입력 >> ");
        try {
            Grade.math = scanner.nextInt();
            Grade.science = scanner.nextInt();
            Grade.english = scanner.nextInt();
        } catch (InputMismatchException e) {
            System.out.println("숫자만 입력하여 주세요.");
            System.exit(-1);
        }
        Grade me = new Grade(math, science, english);
        System.out.println("평균은 " + me.average());
        scanner.close();
    }
}
```

● Result

1) INPUT = 정상적인 입력

```
수학, 과학, 영어 순으로 3개의 점수 입력 >> 90 88 95
평균은 91
```

2) INPUT = 비정상적인 입력

수학, 과학, 영어 순으로 3개의 점수 입력 >> 33 99 aa
숫자만 입력하여 주세요.

2. (4번) Rectangle.java – 두 직사각형의 서로의 포함 유무 출력

● Source

```
public class Rectangle {
    private int x;
    private int y;
    private int width;
    private int height;
    private Rectangle(int x, int y, int width, int height) {
        this.x = x;
        this.y = y;
        this.width = width;
        this.height = height;
    }
    private int square() {
        return width * height;
    }
    private boolean contains(Rectangle r) {
        if(r.x > this.x && r.y > this.y) {
            if((r.x + r.width) < (this.x + width) && (r.y + r.height) < (this.y + height)) {
                return true;
            }
        }
        return false;
    }
    private void show() {
        System.out.println("(" + x + ", " + y + ") 에서 크기가 " + width + "*" + height + "인 사각형 ");
    }
    public static void main(String args[]) {
        Rectangle r = new Rectangle(2, 2, 8, 7);
        Rectangle s = new Rectangle(5, 5, 6, 6);
        Rectangle t = new Rectangle(1, 1, 10, 10);
        r.show();
        System.out.println("s 의 면적은 " + s.square());
        if(t.contains(r)) System.out.println("t 는 r 을 포함합니다");
        if(t.contains(s)) System.out.println("t 는 s 을 포함합니다");
    }
}
```

● Result

```
(2, 2) 에서 크기가 8*7인 사각형
s의 면적은 36
t는 r을 포함합니다
```

3. (6번) CircleManager.java – 3개의 원 정보를 입력받고 가장 면적이 큰 원 출력

● Source

```
import java.util.InputMismatchException;
import java.util.Scanner;
class Circle {
    private double x, y;
    private int radius;
    Circle(double x, double y, int radius) {
        this.x = x;
        this.y = y;
        this.radius = radius;
    }
    int area() {
        return this.radius * this.radius;
    }
    void show() {
        System.out.println("(" + x + ", " + y + ") " + radius);
    }
}
public class CircleManager {
    public static void main(String args[]) {
        Scanner scanner = new Scanner(System.in);
        double x;
        double y;
        int radius;
        Circle[] c = new Circle[3];
        for(int i = 0; i < c.length; i++) {
            System.out.print("x, y, radius >> ");
            try {
                x = scanner.nextDouble();
                y = scanner.nextDouble();
                radius = scanner.nextInt();
                c[i] = new Circle(x, y, radius);
            } catch (InputMismatchException e) {
                System.out.println("x, y 는 실수로, radius 는 정수로 입력하여
주세요.");
                scanner.nextLine();
                i--;
            }
        }
        CompareArea(c[0], c[1], c[2]);
    }
    private static void CompareArea(Circle a, Circle b, Circle c) {
        if (a.area() == b.area() && b.area() == c.area()) {
            System.out.println("세 원의 면적이 모두 같습니다.");
        } else if (a.area() == b.area() || b.area() == c.area() || a.area() ==
c.area()) {
            System.out.println("같은 면적을 가진 두개의 원이 있습니다");
        } else {
            if (a.area() > b.area()) {
                System.out.print("가장 면적이 큰 원은 ");
                a.show();
            } else if (b.area() > c.area()) {
                System.out.print("가장 면적이 큰 원은 ");
                b.show();
            } else if (c.area() > a.area()) {
                System.out.print("가장 면적이 큰 원은 ");
                c.show();
            }
        }
    }
}
```

```

        System.out.print("가장 면적이 큰 원은 ");
        c.show();
    }
}
}

```

- Result

```

x, y, radius >> 3.0 3.0 aa
x, y는 실수로, radius는 정수로 입력하여 주세요.
x, y, radius >> 3.0 3.0 11
x, y, radius >> 3.2 4.4 12
x, y, radius >> 12 13 3.3
x, y는 실수로, radius는 정수로 입력하여 주세요.
x, y, radius >> 12 13 3
가장 면적이 큰 원은 (3.2, 4.4)12

```

4. (8번) PhoneBook.java – 전화번호부 프로그램

- Source

```
import java.util.InputMismatchException;
import java.util.Scanner;
class Phone {
    private String name;
    private String phone_num;
    Phone(String name, String phone_num) {
        this.name = name;
        this.phone_num = phone_num;
    }
    String getName() {
        return name;
    }
    String getPhoneNum() {
        return phone_num;
    }
}
public class PhoneBook {
    public static void main(String args[]) {
        int num;
        Scanner scanner = new Scanner(System.in);
        System.out.print("인원수 >> ");
        try {
            num = scanner.nextInt();
            Phone[] phone = new Phone[num];
            for(int i = 0; i < num; i++) {
                System.out.print("이름과 전화번호(이름과 번호는 빈 칸없이 입력) >> ");
                String name = scanner.next();
                String phone_num = scanner.next();
                phone[i] = new Phone(name, phone_num);
            }
            while(true) {
                boolean check = false;
                System.out.print("검색할 이름 >> ");
                String input = scanner.next();
                for (int i = 0; i < phone.length; i++) {
                    if (phone[i].getName().equals(input)) {
                        System.out.println(input + "의 번호는 " +
phone[i].getPhoneNum());
                        check = true;
                        break;
                    }
                }
                if(input.equals("그만"))
                    break;
                else if(!check)
                    System.out.println(input + " 이 없습니다.");
            }
        } catch (InputMismatchException e) {
            System.out.println("숫자만 입력해주세요");
            System.exit(-1);
        }
    }
}
```

- Result

```
인원수 >> 3
이름과 전화번호(이름과 번호는 빈 칸없이 입력) >> 김병준 010-4199-4157
이름과 전화번호(이름과 번호는 빈 칸없이 입력) >> 정미경 010-4129-4157
이름과 전화번호(이름과 번호는 빈 칸없이 입력) >> 김남홍 010-5280-1003
검색할 이름 >> 김병준
김병준의 번호는 010-4199-4157
검색할 이름 >> 함정완
함정완 이 없습니다.
검색할 이름 >> 김남홍
김남홍의 번호는 010-5280-1003
검색할 이름 >> 그만
```

5. (10번) DicApp.java – 한영 변환 사전 프로그램

● Source

```
import java.util.Scanner;
class Dictionary {
    private static String[] kor = {"사랑", "아기", "돈", "미래", "희망"};
    private static String[] eng = {"Love", "Baby", "Money", "Future", "Hope"};
    static String kor2Eng(String word) {
        for(int i = 0; i < kor.length; i++) {
            if(word.equals(kor[i])) return eng[i];
        }
        return null;
    }
}
public class DicApp {
    public static void main(String args[]) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("한영 단어 검색 프로그램입니다.");
        while(true) {
            System.out.print("한글 단어? ");
            String input = scanner.next();
            if (input.equals("그만")) break;
            String answer = Dictionary.kor2Eng(input);
            if (answer == null) System.out.println(input + "은 저의 사전에 없습니다.");
            else System.out.println(input + "은 영어로 " + answer);
        }
    }
}
```

● Result

```
한영 단어 검색 프로그램입니다.
한글 단어? 희망
희망은 영어로 Hope
한글 단어? 아빠
아빠은 저의 사전에 없습니다.
한글 단어? 엄마
엄마은 저의 사전에 없습니다.
한글 단어? 사랑
사랑은 영어로 Love
한글 단어? 아기
아기은 영어로 Baby
한글 단어? 그만
```


6. (12번) ReservationSystem.java – 콘서트 좌석 조회 및 등록 삭제 프로그램

- Source – Solve2to12A: if-else문

```
import java.util.InputMismatchException;
import java.util.Scanner;
class Person {
    private String name;
    Person() { this(null); }
    Person(String name) { this.name = name; }
    String getName() { return this.name; }
    void setName(String name) { this.name = name; }
}

public class ReservationProgram {
    private static Person[][] seat = new Person[3][10];
    private static Scanner scanner = new Scanner(System.in);
    private static int reservation_count = 0;
    public static void main(String args[]) {
        int menu_input;
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 10; j++) {
                seat[i][j] = new Person();
            }
        }
        while (true) {
            System.out.println("명품콘서트홀 예약 시스템입니다.");
            System.out.print("예약:1, 조회:2, 취소:3, 끝내기:4 >> ");
            try {
                menu_input = scanner.nextInt();
            } catch (InputMismatchException e) {
                System.out.println("잘못된 메뉴 선택입니다. 다시입력하여 주세요.");
                scanner.nextLine();
                continue;
            }
            switch (menu_input) {
                default:
                    System.out.println("잘못된 메뉴 선택입니다. 다시입력하여 주세요.");
                    break;
                case 1:
                    ReservationProgram.Reservation();
                    break;
                case 2:
                    ReservationProgram.PrintSeat();
                    break;
                case 3:
                    ReservationProgram.RemoveReservation();
                    break;
                case 4:
                    System.exit(-1);
            }
        }
    }

    private static void Reservation() {
        int layer;
        int seat_num;
        String name;
    }
}
```

```

        boolean condition = false;
        while (!condition) {
            layer = ReservationProgram.selectLayer();
            System.out.print("이름 >> ");
            name = scanner.next();
            seat_num = ReservationProgram.selectSeat();
            if (ReservationProgram.isExist(seat[layer - 1][seat_num - 1])) {
                System.out.println("이미 예약된 좌석입니다. 다른 좌석을
선택하여주세요.");
                condition = false;
            } else {
                condition = true;
                seat[layer - 1][seat_num - 1] = new Person(name);
                reservation_count++;
            }
        }
    }

    private static void RemoveReservation() {
        int layer;
        String name;
        boolean condition = false;
        if (reservation_count == 0) return;
        layer = ReservationProgram.selectLayer();
        System.out.print("이름 >> ");
        name = scanner.next();
        for (int i = 0; i < 10; i++) {
            if (name.equals(seat[layer - 1][i].getName())) {
                seat[layer - 1][i].setName(null);
                reservation_count--;
                condition = true;
            }
        }
        if (!condition) System.out.println("해당 이름으로 등록된 좌석을 찾을 수
없습니다.");
    }

    private static boolean isExist(Person seat) { return (seat.getName() !=
null); }
    private static int selectLayer() {
        int layer;
        while (true) {
            System.out.print("좌석구분 S(1), A(2), B(3) >> ");
            try {
                layer = scanner.nextInt();
            } catch (InputMismatchException e) {
                System.out.println("잘못된 메뉴 선택입니다. 다시입력하여 주세요.");
                scanner.nextLine();
                continue;
            }
            if (layer < 1 || layer > 3) {
                System.out.println("잘못된 메뉴 선택입니다. 다시입력하여 주세요.");
            } else {
                break;
            }
        }
        return layer;
    }

```

```

    }

    private static int selectSeat() {
        int seat_num;
        while (true) {
            System.out.print("번호 >> ");
            try {
                seat_num = scanner.nextInt();
            } catch (InputMismatchException e) {
                System.out.println("잘못된 메뉴 선택입니다. 다시입력하여 주세요.");
                scanner.nextLine();
                continue;
            }
            if (seat_num < 1 || seat_num > 10) System.out.println("잘못된 좌석
선택입니다. 다시입력하여 주세요.");
            else break;
        }
        return seat_num;
    }

    private static void PrintSeat() {
        for (int i = 0; i < 3; i++) {
            switch (i) {
                case 0:
                    System.out.print("S >> ");
                    break;
                case 1:
                    System.out.print("A >> ");
                    break;
                case 2:
                    System.out.print("B >> ");
                    break;
            }
            for (int j = 0; j < 10; j++) {
                if (seat[i][j].getName() == null) System.out.print("___ ");
                else System.out.print(seat[i][j].getName() + " ");
            }
            System.out.println();
        }
        System.out.println("<<<조회를 완료하였습니다.>>>");
    }
}

```

- Result

- 1) 예약(예약자가 없을 경우)

```
명품콘서트홀 예약 시스템입니다.  
예약:1, 조회:2, 취소:3, 끝내기:4 >> 1  
좌석구분 S(1), A(2), B(3) >> 1  
이름 >> 김병준  
번호 >> 5  
명품콘서트홀 예약 시스템입니다.  
예약:1, 조회:2, 취소:3, 끝내기:4 >> 1  
좌석구분 S(1), A(2), B(3) >> 2  
이름 >> 김상현  
번호 >> 3  
명품콘서트홀 예약 시스템입니다.  
예약:1, 조회:2, 취소:3, 끝내기:4 >> 1  
좌석구분 S(1), A(2), B(3) >> 1  
이름 >> 하정완  
번호 >> 3  
명품콘서트홀 예약 시스템입니다.  
예약:1, 조회:2, 취소:3, 끝내기:4 >> 1  
좌석구분 S(1), A(2), B(3) >> 3  
이름 >> 유찬양  
번호 >> 10  
명품콘서트홀 예약 시스템입니다.  
예약:1, 조회:2, 취소:3, 끝내기:4 >>
```

- 2) 예약(예약자가 존재할 경우)

```
명품콘서트홀 예약 시스템입니다.  
예약:1, 조회:2, 취소:3, 끝내기:4 >> 1  
좌석구분 S(1), A(2), B(3) >> 1  
이름 >> 임대호  
번호 >> 3  
이미 예약된 좌석입니다. 다른 좌석을 선택해주세요.
```

3) 조회

```
명품콘서트홀 예약 시스템입니다.  
예약:1, 조회:2, 취소:3, 끝내기:4 >> 2  
S >> _ _ _ _ _ 하정완 _ _ _ _ _ 김병준 _ _ _ _ _  
A >> _ _ _ _ _ 김상현 _ _ _ _ _  
B >> _ _ _ _ _ 유찬양  
<<<조회를 완료하였습니다.>>>
```

4) 취소(이름이 존재할 경우)

```
명품콘서트홀 예약 시스템입니다.  
예약:1, 조회:2, 취소:3, 끝내기:4 >> 3  
좌석구분 S(1), A(2), B(3) >> 1  
이름 >> 김병준  
명품콘서트홀 예약 시스템입니다.  
예약:1, 조회:2, 취소:3, 끝내기:4 >> 2  
S >> _ _ _ _ _ 하정완 _ _ _ _ _  
A >> _ _ _ _ _ 김상현 _ _ _ _ _ 임대호 _ _  
B >> _ _ _ _ _ 유찬양  
<<<조회를 완료하였습니다.>>>
```

5) 취소(이름이 존재하지 않을 경우)

```
명품콘서트홀 예약 시스템입니다.  
예약:1, 조회:2, 취소:3, 끝내기:4 >> 3  
좌석구분 S(1), A(2), B(3) >> 1  
이름 >> 함정완  
해당 이름으로 등록된 좌석을 찾을 수 없습니다.
```

6) 종료

```
명품콘서트홀 예약 시스템입니다.  
예약:1, 조회:2, 취소:3, 끝내기:4 >> 4  
  
Process finished with exit code 255
```

7) 잘못된 입력

명품콘서트홀 예약 시스템입니다.

예약:1, 조회:2, 취소:3, 끝내기:4 >> aa

잘못된 메뉴 선택입니다. 다시입력하여 주세요.

좌석구분 S(1), A(2), B(3) >> 4

잘못된 메뉴 선택입니다. 다시입력하여 주세요.

번호 >> 110

잘못된 좌석 선택입니다. 다시입력하여 주세요.