

2022년 1학기 자바프로그래밍 중간고사 금요일반

제출파일: Calculator.java, midexam2.zip

- 기능 구현과 상관없이 프로그램 동작 중 예외가 발생하는 경우, 각 항목별 -2점 감점함
- 각 기능별 부분 점수는 없음 (기능에 대한 코드만 있고 동작이 안되면 점수 없음)
- 그 외 과제의 감점 요인과 동일: 컴파일 에러, 주석, UTF-8 등

1. 상속 및 오버라이딩을 활용한 사칙연산 프로그램 (10점)

제출파일: Calculator.java

아래에 제공된 Calc 클래스를 상속 받고 overriding 기법을 적용하여서 사칙 연산을 수행하는 클래스를 각각 구현하시오.

- 제출파일: **Calculator.java** (하나의 파일에 모두 구현)
- Calc 클래스
 - ✓ 부모 클래스
- Add 클래스: 덧셈 연산 수행 (1점)
 - ✓ Calc 클래스를 상속 받고 calculate(int a, int b) 메소드 오버라이딩
 - ✓ 객체 출력을 위한 toString() 메소드 오버라이딩, 예) "Add: 2 + 3 = "
- Sub 클래스: 뺄셈 연산 수행 (1점)
 - ✓ Calc 클래스를 상속 받고 calculate(int a, int b) 메소드 오버라이딩
 - ✓ 객체 출력을 위한 toString() 메소드 오버라이딩, 예) "Sub: 4 - 3 = "
- Mul 클래스: 곱셈 연산 수행 (1점)
 - ✓ Calc 클래스를 상속 받고 calculate(int a, int b) 메소드 오버라이딩
 - ✓ 객체 출력을 위한 toString() 메소드 오버라이딩, 예) "Mul: 1 * 2 = "
- Div 클래스: 나눗셈 연산 수행 (1점)
 - ✓ Calc 클래스를 상속 받고 calculate(int a, int b) 메소드 오버라이딩
 - a/b에서 b가 0인 경우 "0으로 나눌 수 없습니다." 출력
 - ✓ 객체 출력을 위한 toString() 메소드 오버라이딩, 예) "Div: 10 / 5 = "
- Calculator 클래스 (6점)
 - ✓ main()함수가 있는 클래스
 - 두 정수 및 연산자(+, -, *, /) 입력 받음 (1점)
 - 프로그램 종료는 입력 받은 두 정수가 모두 -1인 경우, 프로그램 종료 (1점)
 - 입력된 연산자에 따라 해당 객체 생성 및 calculate() 호출 (1점)
 - 잘못된 입력에 대한 예외 처리 기능 구현 (2점)
 - 각 클래스에 오버라이딩된 toString() 을 이용한 객체 출력 및 연산 결과 출력 (1점)

```
class Calc {
    protected int a, b;
    public Calc() {
        a = b = 0;
    }
    public void setValue(int a, int b) {
        this.a = a;
        this.b = b;
    }
    public int calculate() { // 서브 클래스에서 재정의
        return 0;
    }
}
```

```
}  
}
```

실행결과: 정상 동작

- 주황색 부분: toString() 오버라이딩 구현 예

```
두 정수와 연산자를 입력하세요(종료: -1 -1): 1 2 +  
덧셈 연산  
Add:1 + 2 = 3  
두 정수와 연산자를 입력하세요(종료: -1 -1): 2 3 *  
곱셈 연산  
Mul:2 * 3 = 6  
두 정수와 연산자를 입력하세요(종료: -1 -1): 3 4 -  
뺄셈 연산  
Sub:3 - 4 = -1  
두 정수와 연산자를 입력하세요(종료: -1 -1): 4 5 /  
나눗셈 연산  
Div:4 / 5 = 0  
두 정수와 연산자를 입력하세요(종료: -1 -1): 2 0 /  
나눗셈 연산  
0으로 나눌 수 없습니다.  
두 정수와 연산자를 입력하세요(종료: -1 -1): -1 -1  
종료합니다.
```

실행 결과: 예외 처리

- 사칙 연산 기호 이외의 기호 처리
- 입력 순서 오류 처리
- 숫자가 아닌 문자 입력 예외 처리

```
두 정수와 연산자를 입력하세요(종료: -1 -1): 1 2 %  
잘못된 연산자입니다. 다시 입력하세요.  
두 정수와 연산자를 입력하세요(종료: -1 -1): 1 + 2  
잘못된 입력입니다. 다시 입력하세요.  
두 정수와 연산자를 입력하세요(종료: -1 -1): / 30 6  
잘못된 입력입니다. 다시 입력하세요.  
두 정수와 연산자를 입력하세요(종료: -1 -1): 20 - 10 +  
잘못된 입력입니다. 다시 입력하세요.  
두 정수와 연산자를 입력하세요(종료: -1 -1): 20 10 ^  
잘못된 연산자입니다. 다시 입력하세요.  
두 정수와 연산자를 입력하세요(종료: -1 -1): a b +  
잘못된 입력입니다. 다시 입력하세요.  
두 정수와 연산자를 입력하세요(종료: -1 -1): -1 -1  
종료합니다.
```

2. 인터페이스 구현을 활용한 주사위 게임 프로그램 (15점)

제출파일: midexam2.zip

2명의 선수가 각각 2개의 주사위를 가지고 게임을 한다. 사용자가 초기에 입력한 회수만큼 2개의 주사위를 던져서 나온 숫자의 합이 큰 경우, 각 게임을 이기는 것으로 결정하였다. 제공된

RollingDiceInterface (인터페이스)를 활용하여 주사위 게임을 구현하고, 주사위 합의 빈도수 및 두 주사위의 숫자별 빈도수를 계산하고 화면에 출력하시오.

- 제공된 인터페이스를 사용하지 않거나 임의로 변경할 경우, 0점 처리함


■ 제출파일: Gamer.java, RollingDiceApp.java

■ Gamer.java (5점)

- ✓ RollingDiceInterface 4개 기능 구현 (각 1점)
- ✓ toString() 메소드 오버라이딩: "사람이름(주사위1 값, 주사위2 값) = 주사위의 합" (1점)
- ✓ 생성자에 gamer 이름 및 주사위의 최대 숫자 입력 (14면체 2개 사용)

■ RollingDiceApp.java (10점)

- ✓ main()가 있는 클래스
- ✓ 두 명의 Gamer 객체 생성 (Alice, Bob)
- ✓ 사용자가 입력한 숫자만큼 주사위를 던져서 각 gamer의 승패를 비교하고 화면에 출력 (3점)
 - "Win", "Even game" 출력
 - 무한 반복 없음
- ✓ 각 gamer의 두 주사위의 합의 누적된 빈도수를 계산하고 화면 출력 (3점)
- ✓ 각 gamer의 두 주사위 숫자의 누적된 빈도수를 계산하고 화면 출력 (4점)
 - (1, 3)인 경우, 2차원 배열의 해당 index에 표시하고 화면에 출력

| Gamer1 (Alice) | Gamer2 (Bob) |
|---|--|
|  |  |

<RollingDiceInterface 내용>

```
public interface RollingDiceInterface {
    /**
     * 2개의 랜덤 숫자 생성 (1 ~ 주사위의 최대 숫자 14)
     */
    public abstract void roll();
    /**
     * 2개 주사위의 값을 배열로 리턴
     * @return
     */
    public abstract int[] getFaceNumbers();
    /**
     * 2개 주사위 값의 합을 리턴
     * @return
     */
}
```

```

    */
    public abstract int getDiceSum();
    /**
     * 자신의 2개 주사위의 합과 다른 선수의 주사위의 합을 비교
     * - Upcasting, downcasting 사용
     * @param Object
     * @return
     *      . player1 > player2, return 1
     *      . player1 == player2, return 0
     *      . player1 < player2, return -1
     */
    public abstract int compareTo(Object obj);
}

```

실행 결과

주사위를 던질 회수를 입력하세요: 10

```

-----
Alice( 3,12)= 15   =      Bob( 9, 6)= 15 : Even game
Alice( 6, 7)= 13   <      Bob( 2,14)= 16 : Bob Win
Alice( 4,11)= 15   >      Bob(10, 4)= 14 : Alice Win
Alice(14, 4)= 18   >      Bob( 1, 3)=  4 : Alice Win
Alice( 7,13)= 20   >      Bob( 8, 1)=  9 : Alice Win
Alice( 6,10)= 16   >      Bob( 1,12)= 13 : Alice Win
Alice( 5, 4)=  9   <      Bob( 5, 8)= 13 : Bob Win
Alice(12,12)= 24   >      Bob( 7, 9)= 16 : Alice Win
Alice(14, 8)= 22   >      Bob( 4, 4)=  8 : Alice Win
Alice(14, 9)= 23   >      Bob( 8,11)= 19 : Alice Win

```

[두 주사위 합의 빈도수]

```

-----
 2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
-----
 0  0  1  0  0  0  1  2  0  0  0  3  1  3  0  1  1  1  0  1  1  1  0  0  0  0
-----

```

[주사위 숫자별 빈도수]

```

-----
  1  2  3  4  5  6  7  8  9 10 11 12 13 14
1 0 0 1 0 0 0 0 0 0 0 0 1 0 0
2 0 0 0 0 0 0 0 0 0 0 0 0 0 1
3 0 0 0 0 0 0 0 0 0 0 0 1 0 0
4 0 0 0 1 0 0 0 0 0 0 1 0 0 0
5 0 0 0 1 0 0 0 1 0 0 0 0 0 0
6 0 0 0 0 0 0 1 0 0 1 0 0 0 0
7 0 0 0 0 0 0 0 0 1 0 0 0 1 0
8 1 0 0 0 0 0 0 0 0 0 1 0 0 0
9 0 0 0 0 0 1 0 0 0 0 0 0 0 0
10 0 0 0 1 0 0 0 0 0 0 0 0 0 0
11 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12 0 0 0 0 0 0 0 0 0 0 0 1 0 0
13 0 0 0 0 0 0 0 0 0 0 0 0 0 0
14 0 0 0 1 0 0 0 1 1 0 0 0 0 0
-----

```