

## Exercise 5

- 아래의 Unit class를 상속받아서 1~3번 문제를 푼다.
- ArrayList<Unit> 을 사용해서 각 문제의 Unit 들을 저장한다.
- 모든 문제에서 입력으로 list 를 받으면 지금까지 만들어진 Unit 들을 toString() 메소드를 통해 출력한다.
  - 자세한 예는 1~3의 입출력에서 보여준다.
- 모든 문제에서 입력의 갯수는 정해지지 않고, 대신 한줄에 QUIT 가 입력되면 프로그램을 멈춘다.
- 모든 정수연산의 결과는 int 의 범위를 넘지 않는다.

```
class Unit {
    protected String operator;
    protected int left;
    protected int right;

    public Unit(int left, int right, String operator) {
        // TODO: implement me

        calculate(); // Dynamic binding
    }

    protected void calculate() {
        // TODO: implement me
    }
}
```

### 메소드 설명

- 생성자에서는 파라미터로 받은 값을 저장하고, 연산한 결과를 아래의 출력 예시처럼 출력한다.
- calculate() 에서는 생성자를 통해 입력받은 값으로 결과값을 연산하고 result 에 저장한다.
- toString() 에서는 calculate() 에서 연산한 결과를 아래의 출력 예시처럼 출력한다.

## 1. Arithmetic Unit

### 1-1. 설명

```
class ArithmeticUnit extends Unit {
    protected int result;

    public ArithmeticUnit(int left, int right, String operator) {
        super(left, right, operator);
    }

    @Override
    protected void calculate() {
        // TODO: implement me
    }

    @Override
    public String toString() {
        // TODO: implement me
    }
}
```

- 정수 연산(+, -, /, %, \*, ^ (power))을 하는 Unit
- 정수 2개와 하나의 연산자를 받아서 계산하는 Unit 을 구현하라.

### 1-2. 입력 예시

```
2 4 +
3 5 *
list
6 4 /
list
16 5 %
3 10 -
3 6 ^
QUIT
```

- int 범위의 숫자 2개와 연산자가 (스페이스)를 간격으로 입력된다.

### 1-3. 출력 예시

```
2+4=6
3*5=15
2+4=6
3*5=15
6/4=1
2+4=6
3*5=15
6/4=1
16%5=1
3-10=-7
3^6=729
```

## 2. Compare Unit

### 2-1. 설명

```
class CompareUnit extends Unit {
    protected boolean result;

    public CompareUnit(int left, int right, String operator) {
        // TODO: implement me
    }

    @Override
    protected void calculate() {
        // TODO: implement me
    }

    @Override
    public String toString() {
        // TODO: implement me
    }
}
```

- 비교 연산(==, !=, <=, <, >, >=)을 하는 Unit
- 정수 2개와 하나의 연산자를 받아서 계산하는 Unit을 구현하라.

### 2-2. 입력 예시

```
2 4 <=
list
3 5 ==
6 4 >
6 15 >=
3 10 !=
QUIT
```

- int 범위의 숫자 2개와 연산자가 (스페이스)를 간격으로 입력된다.

### 2-3. 출력 예시

```
2<=4 : true
2<=4 : true
3==5 : false
6>4 : true
6>=15 : true
3!=10 : true
```

## 3. Bit Unit

### 3-1. 설명

```

class BitUnit extends Unit {
    protected int result;

    public BitUnit(int left, int right, String operator) {
        // TODO: implement me
    }

    @Override
    protected void calculate() {
        // TODO: implement me
    }

    @Override
    public String toString() {
        // TODO: implement me
    }
}

```

- 비트 연산(|, &, >>, <<)을 하는 Unit
- 정수 2개와 하나의 연산자를 받아서 계산하는 Unit을 구현하라.

### 3-2. 입력 예시

```

1 1 &
2 1 |
16 2 >>
8 2 <<
list
QUIT

```

- int 범위의 숫자 2개와 연산자가 (스페이스)를 간격으로 입력된다.

### 3-3. 출력 예시

```

1&1=1
2|1=3
16>>2=4
8<<2=32
1&1=1
2|1=3
16>>2=4
8<<2=32

```