

□ 개념 확인

- (1) main() 메소드에서 Action 클래스를 상속받는 익명 객체를 구현하고 move메소드를 호출하는 프로그램을 작성하세요. 단, 구현 내용은 본인이 결정합니다.

```
abstract class Action{
    protected String animal;
    public Action(String animal) {
        this.animal = animal;
    }
    abstract void move();
}
```

[프로그램 소스]

```
package practice1;

abstract class Action{
    protected String animal;
    public Action(String animal) {
        this.animal = animal;
    }
    abstract void move(); // abstract 붙여야 한다.
}
```

```
package practice1;

public class Main {
    public static void main(String[] args) {
        Action action = new Action("Dog") { // 상속을 받는 객체를 생성
            void move() {
                System.out.println(animal + " 는/은 네발로 걸어서 이동합니다.");
            };
        };
        action.move();
    }
}
```

[실행 결과]

```
Dog 는/은 네발로 걸어서 이동합니다.
```

- (2) 정적 인터페이스 멤버를 갖는 클래스 Calculator를 제시된 결과처럼 실행될 수 있도록 구현하여 프로그램을 완성 하세요

```
class Calculator{
```

```
//1) 인터페이스 Control 타입을 갖는 필드 변수 cal 선언 - 본인 작성
```

```
//2) 인터페이스 Control 타입 매개변수 1개를 갖는 setControl() 메소드 구현 - 본인 작성
```

```
//반환값 없으며 매개변수를 필드 변수 값으로 대입
```

```
void compute(int a, int b) {
```

```
    cal.run(a, b);
```

```
}
```

```
static interface Control{
```

```
    void run(int a, int b);
```

```
}
```

```
}
```

```
//정적 인터페이스 멤버를 구현하는 Add 클래스 구현 - 합을 계산하여 출력 - 본인 작성
```

```
//정적 인터페이스 멤버를 구현하는 Exp 클래스 구현 -  $a^b$  를 계산하여 출력 - 본인 작성
```

```
public class Answer02 {
```

```
    public static void main(String[] args) {
```

```
        Calculator cal = new Calculator();
```

```
        cal.setControl(new Add());
```

```
        cal.compute((int) (Math.random() * 10) + 1, (int) (Math.random() * 10) + 1);
```

```
        cal.setControl(new Exp());
```

```
        cal.compute(5, 3);
```

```
    }
```

```
}
```

[프로그램 소스]

```
package practice2;
```

```
class Calculator{
```

```
//1) 인터페이스 Control 타입을 갖는 필드 변수 cal 선언 - 본인 작성
```

```
Control cal;
```

```
//2) 인터페이스 Control 타입 매개변수 1개를 갖는 setControl() 메소드 구현 - 본인 작성
```

```
//반환값 없으며 매개변수를 필드 변수 값으로 대입
```

```
void setControl(Control cal){
```

```
    this.cal = cal;
```

```
}
```

```
void compute(int a, int b){
```

```
    cal.run(a, b);
```

```
----- Add -----
```

```
8 + 9 = 17
```

```
----- Exp -----
```

```
5 ** 3 = 125
```

```

    }

    static interface Control{ // static 생략 가능
        void run(int a, int b);
    }
}

```

```

package practice2;

class Add implements Calculator.Control {
    public void run(int a, int b) { // public 반드시 기재해야함
        System.out.println(a + " + " + b + " = " + (a+b));
    }
}

```

```

package practice2;

class Exp implements Calculator.Control {
    public void run(int a, int b) {
        int result = 1;
        for(int i = 1; i <= b; i++){

            result *= a;
        }
        System.out.println(a + " ** " + b + " = " + result);
    }
}

```

```

package practice2;

public class Answer02 {
    public static void main(String[] args) {
        Calculator cal = new Calculator();
        System.out.println("----- Add -----");
        cal.setControl(new Add());
        cal.compute((int) (Math.random() * 10) + 1, (int) (Math.random() * 10) + 1);
        System.out.println("----- Exp -----");
        cal.setControl(new Exp());
        cal.compute(5, 3);
    }
}

```

[실행 결과]

```
----- Add -----  
8 + 7 = 15  
----- Exp -----  
5 ** 3 = 125
```

□ 응용 프로그래밍

- (3) 다음과 같은 인터페이스를 구현하는 익명 객체를 생성하고 테스트 하는 프로그램을 작성하세요. 제시된 결과를 참고하세요.

```
interface Media {  
    void play(String instru); //재생  
    void stop(String instru); //중지  
}
```

```
기기를 선택하세요  
1: MP3 2:CDPlay 3:WindowMediaPlayer >>2  
CDPlay를 재생합니다.  
CDPlay를 중지합니다.
```

```
public class Answer03 {  
    public static void main(String[] args) {  
        //Media 클래스를 상속받는 익명 자식 객체 구현 - 본인 작성  
        //메뉴를 선택하고 실행 - 본인 작성  
    }  
}
```

[프로그램 소스]

```
package practice3;  
  
interface Media {  
    void play(String instru); //재생  
    void stop(String instru); //중지  
}
```

```
package practice3;  
  
import java.util.Scanner;  
  
public class Answer03 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("기기를 선택하세요\n1: MP3 2: CDPlay 3: WindowMediaPlayer >>");  
        String instru = sc.next();  
        //Media 클래스를 상속받는 익명 자식 객체 구현 - 본인 작성  
        Media media = new Media(){  
            public void play(String instru) {  
                switch(instru){  
                    case "1":
```

```

        System.out.println("MP3 를 재생합니다.");
        break;
    case "2":
        System.out.println("CDPlay 를 재생합니다.");
        break;
    case "3":
        System.out.println("WindowMediaPlay 를 재생합니다.");
        break;
    default:
        System.out.println("잘못된 입력입니다.");
    }
}

public void stop(String instru){
    switch(instru){
        case "1":
            System.out.println("MP3 를 중지합니다.");
            break;
        case "2":
            System.out.println("CDPlay 를 중지합니다.");
            break;
        case "3":
            System.out.println("WindowMediaPlay 를 중지합니다.");
            break;
        default:
            System.out.println("잘못된 입력입니다.");
    }
}

};
//메뉴를 선택하고 실행 - 본인 작성
media.play(instru);
media.stop(instru);
}
}

```

[실행 결과]

기기를 선택하세요

1: MP3 2: CDPlay 3: WindowMediaPlay >>2

CDPlay를 재생합니다.

CDPlay를 중지합니다.

- (4) 제시된 인터페이스에 대한 익명 객체를 생성하여 매개변수로 전달합니다. 각 익명 객체는 <<, >>, &, ^ 연산을 수행합니다.

```

interface Calculator {
    int run(int a, int b);
}

public class Answer04 {
    public static void execute(Calculator cal, String ch) {
        int a = (int) (Math.random() * 20) + 1;
        int b = (int) (Math.random() * 5) + 1;
        System.out.print(a + " " + ch + " " + b + " => ");
        System.out.println(cal.run(a,b));
    }

    public static void main(String[] args) {
        //본인 작성
    }
}

```

11	>>	3	=>	1
13	<<	3	=>	104
4	&	5	=>	4
17	^	1	=>	16

[프로그램 소스]

```
package pratice4;
```

```

interface Calculator {
    int run(int a, int b);
}

```

```
package pratice4;
```

```

public class Answer04 {
    public static void execute(Calculator cal, String ch) {
        int a = (int) (Math.random() * 20) + 1;
        int b = (int) (Math.random() * 5) + 1;
        System.out.print(a + " " + ch + " " + b + " => ");
        System.out.println(cal.run(a,b));
    }

    public static void main(String[] args) {
        //본인 작성
        execute(new Calculator() {
            public int run(int a, int b) {
                return a << b;
            }
        }, ">>");
        execute(new Calculator() {
            public int run(int a, int b) {
                return a << b;
            }
        }, "<<");
    }
}

```

```

        execute(new Calculator() {
            public int run(int a, int b) {
                return a >> b;
            }
        }, ">>");
        execute(new Calculator() {
            public int run(int a, int b) {
                return a << b;
            }
        }, "<<");
        execute(new Calculator() {
            public int run(int a, int b) {
                return a & b;
            }
        }, "&");
        execute(new Calculator() {
            public int run(int a, int b) {
                return a ^ b;
            }
        }, "^");
    }
}

```

[실행 결과]

```

2   >>   1 => 4
18  <<   3 => 144
3   &    4 => 0
20  ^    3 => 23

```

(5) 4번 문제를 람다식을 사용하여 해결하는 프로그램으로 수정하세요

[프로그램 소스]

```

package practice5;

interface Calculator {
    int run(int a, int b);
}

```

```

package practice5;

public class Answer05 {
    public static void main(String[] args) {
        Calculator leftShift = (a,b) -> a << b;
        Calculator rightShift = (a,b) -> a >> b;
        Calculator and = (a,b) -> a & b;
        Calculator xor = (a,b) -> a ^ b;
        execute(leftShift, ">>");
        execute(rightShift, "<<");
        execute(and, "&");
        execute(xor, "^");
    }

    public static void execute(Calculator cal, String ch) {
        int a = (int) (Math.random() * 20) + 1;
        int b = (int) (Math.random() * 20) + 1;
        int result = cal.run(a, b);
        System.out.println(a + " " + ch + " " + b + " => " + result);
    }
}

```

```

    int b = (int) (Math.random() * 5) + 1;
    System.out.print(a + " " + ch + " " + b + " => ");
    System.out.println(cal.run(a,b));
}
}

```

[실행 결과]

```

11  >>  4 => 176
3    <<  4 => 0
3    &   4 => 0
14   ^   2 => 12

```

(6) 다음과 같은 인터페이스를 람다식으로 구현하세요.

@FunctionalInterface

```

interface Action{
    void play(String media);
}

```

피아노를 연주하였습니다
축구경기를 하였습니다
게임을 하였습니다

[프로그램 소스]

```

package practice6;

@FunctionalInterface
interface Action {
    void play(String media);
}

```

```

package practice6;

public class Answer06 {
    public static void main(String[] args) {
        Action action = (media) -> System.out.println(media + "하였습니다.");

        action.play("피아노 연주");
        action.play("축구경기를 ");
        action.play("게임을 ");
    }
}

```

[실행 결과]

피아노 연주하였습니다.
축구경기를 하였습니다.
게임을 하였습니다.

(7) 6번 문제를 정적 메소드 참조로 구현하세요. 단, 정적 메소드는 별도의 클래스로 구현해야 합니다

[프로그램 소스]

```
package practice7;

interface Action {
    void play(String media);
}
```

```
package practice7;

public class Ref{
    public static void piano(String media) {
        System.out.println(media + "를 연주하였습니다.");
    }
    public static void soccer(String media) {
        System.out.println(media + "경기를 하였습니다.");
    }
    public static void game(String media) {
        System.out.println(media + "을 하였습니다.");
    }
}
```

```
package practice7;

public class Answer07 {
    public static void move(Action action, String media) {
        action.play(media);
    }
    public static void main(String[] args) {
        move(Ref::piano, "피아노");
        move(Ref::soccer, "축구");
        move(Ref::game, "게임");
    }
}
```

[실행 결과]

피아노를 연주하였습니다.
축구경기를 하였습니다.
게임을 하였습니다.

- (8) 학생 마다 수강 과목 수가 다르다고 할 때, 각 학생의 수강과목을 입력 받아 출력하는 프로그램의 부분 소스를 완성 하세요. 주석으로 표시된 부분을 구현하면 됩니다.

//중첩 클래스

class NestStudent {

1) 필드 선언

// 학생 이름(private, String)

// 수강과목(private, Subject)

public NestStudent(String name, int cnt) {

//학생 이름 초기화

//Subject 객체 생성, 생성자 매개변수로 cnt 전달

}

void stdPrint() { // 객체 내용 출력

// 학생 이름 출력

// Subject 객체의 subjectprint() 를 호출하여 수강과목 출력

}

//인스턴스 멤버 클래스

class Subject {

// 수강 과목을 저장할 배열 선언

public Subject(int num) {

//매개변수로 받은 값만큼 배열을 생성하고 수강과목을 입력 받아 저장

}

public void subjectprint() {

//배열에 저장된 수강과목 출력

}

}

}

public class Answer08 {

public static void main(String[] args) {

NestStudent obj[] = { new NestStudent("홍길동", 2), new NestStudent("임꺽정", 3),

new NestStudent("심청이", 1) };

for (NestStudent ns : obj)

ns.stdPrint();

}

```
수강 과목(1): 자바
수강 과목(2): 파이썬
임꺽정==>
수강 과목(1): 선형대수
수강 과목(2): 이산수학
수강 과목(3): 컴파일러
심청이==>
수강 과목(1): 데이터베이스
=====
홍길동학생의 수강과목
자바
파이썬
=====
임꺽정학생의 수강과목
선형대수
이산수학
컴파일러
=====
심청이학생의 수강과목
데이터베이스
```

```
}
```

[프로그램 소스]

```
package practice8;

import java.util.Scanner;

//중첩 클래스
class NestStudent {
    // 1) 필드 선언
    // 학생 이름(private, String)
    private String name;
    // 수강과목(private, Subject)
    private Subject subject;

    public NestStudent(String name, int cnt) {
        //학생 이름 초기화
        this.name = name;
        System.out.println(name + "==>");
        //Subject 객체 생성, 생성자 매개변수로 cnt 전달
        subject = new Subject(cnt);
    }

    void stdPrint() { // 객체 내용 출력
        // 학생 이름 출력
        System.out.println("=====");
        System.out.println(name + "학생의 수강과목");
        // Subject 객체의 subjectprint() 를 호출하여 수강과목 출력
        subject.subjectprint();
    }

    //인스턴스 멤버 클래스
    class Subject {

        // 수강 과목을 저장할 배열 선언
        private String[] subs;

        public Subject(int num) {
            //매개변수로 받은 값만큼 배열을 생성하고 수강과목을 입력 받아 저장
            subs = new String[num];
            Scanner sc = new Scanner(System.in);
            for (int i = 0; i < num; i++) {
                System.out.print("수강과목(" + (i + 1) + ") : ");
                subs[i] = sc.next();
            }
        }
    }
}
```

```

    }

    public void subjectprint() {
        //배열에 저장된 수강과목 출력
        for (int i = 0; i < subs.length; i++) {
            System.out.println(subs[i]);
        }
    }
}
}
}

```

```

package practice8;

public class Answer08 {
    public static void main(String[] args) {
        NestStudent obj[] = { new NestStudent("홍길동", 2), new NestStudent("임꺽정", 3),
                               new NestStudent("심청이", 1) };
        for (NestStudent ns : obj)
            ns.stdPrint();
    }
}

```

[실행 결과]

홍길동==>

수강과목(1) : 자바

수강과목(2) : 파이썬

임꺽정==>

수강과목(1) : 선형대수

수강과목(2) : 이산수학

수강과목(3) : 컴파일러

심청이==>

수강과목(1) : 데이터베이스

=====

홍길동학생의 수강과목

자바

파이썬

=====

임꺽정학생의 수강과목

선형대수

이산수학

컴파일러

=====

심청이학생의 수강과목

데이터베이스