

과목명	객체지향프로그래밍	분반	x	담당교수	김화성 교수님
학과	전자통신공학과	학번	2016707079	이름	하상천
과제명: 상속					

1, 과제설명 (사용자 요구사항 기술: 과제에 대한 설명 및 목표)

11. 철수 학생은 다음 3개의 필드와 메소드를 가진 4개의 클래스 Add, Sub, Mul, Div를 작성하려고 한다(4장 실습문제 11 참고).

- int 타입의 a, b 필드: 2개의 피연산자
- void setValue(int a, int b): 피연산자 값을 객체 내에 저장한다.
- int calculate(): 클래스의 목적에 맞는 연산을 실행하고 결과를 리턴한다.

int a
int b
setValue()
calculate()

int a
int b
setValue()
calculate()

int a
int b
setValue()
calculate()

int a
int b
setValue()
calculate()

Add Sub Mul Div

곰곰 생각해보니, Add, Sub, Mul, Div 클래스에 공통된 필드와 메소드가 존재하므로 새로운 추상 클래스 Calc를 작성하고 Calc를 상속받아 만들면 되겠다고 생각했다. 그리고 main() 메소드에서 다음 실행 사례와 같이 2개의 정수와 연산자를 입력받은 후, Add, Sub, Mul, Div 중에서 이 연산을 처리할 수 있는 객체를 생성하고 setValue() 와 calculate()를 호출하여 그 결과 값을 화면에 출력하면 된다고 생각하였다. 철수 처럼 프로그램을 작성하라. 난이도 5

두 정수와 연산자를 입력하시오>>> 5 7 +
12

2, 사용자 요구사항을 정형적 방법으로 기술 (UML, Pseudo code, 그림등을 이용하여 기술)

접근 권한

클래스 선언

클래스 이름

```

public class Circle {
    public int radius; // 원의 반지름 필드
    public String name; // 원의 이름 필드

    public Circle() { // 원의 생성자 메소드
    }
    public double getArea() { // 원의 면적 계산 메소드
        return 3.14*radius*radius;
    }
}

```

필드(변수)

메소드

□ 추상 메소드(abstract method)

- 선언되어 있으나 구현되어 있지 않은 메소드, abstract로 선언

```
public abstract String getName();  
public abstract void setName(String s);
```

- 추상 메소드는 서브 클래스에서 오버라이딩하여 구현해야 함

□ 추상 클래스(abstract class)의 2종류

1. 추상 메소드를 하나라도 가진 클래스
 - 클래스 앞에 반드시 abstract라고 선언해야 함
2. 추상 메소드가 하나도 없지만 abstract로 선언된 클래스



3. 알고리즘 및 자료구조 설계 내용

Add, Sub, Mul, Div 클래스에 공통된 필드와 메소드가 존재하므로 추상클래스 Calc를 작성하고 Calc를 상속받도록 하였다. main() 메소드에서 두 정수와 연산자를 입력받은 후, switch문을 통해 각 클래스의 객체를 생성한다. setValue 메소드를 통해 피연산자 값을 객체 내에 저장하고 calculate 메소드를 통해 결과 값을 리턴 받고 그 값을 출력해준다.

4. 소스코드 설명 (직접 작성한 소스코드중에 핵심 부분을 발췌하여 설명)

```
abstract void setValue(int a, int b);
```

추상클래스 Calc안에 추상메소드를 만들었다.

```
void setValue(int a, int b){ this.a=a; this.b = b; }
```

추상클래스를 상속받은 후 setValue 메소드를 오버라이딩 하였다. 메소드 이름, 매개변수 개수, 타입, 리턴타입을 모두 동일하게 하였다.

```
int calculate() {return a-b;}
```

추상클래스를 상속받은 후 calculate 메소드를 오버라이딩 하였다. 메소드 이름, 매개변수 개수, 타입, 리턴타입을 모두 동일하게 하였지만, 메소드 내용을 덧셈, 뺄셈, 곱하기, 나누기로 클래스마다 다르게 하였다.

5. 실행결과 및 설명 (실행 결과를 캡처하여 첨부한 후 설명)

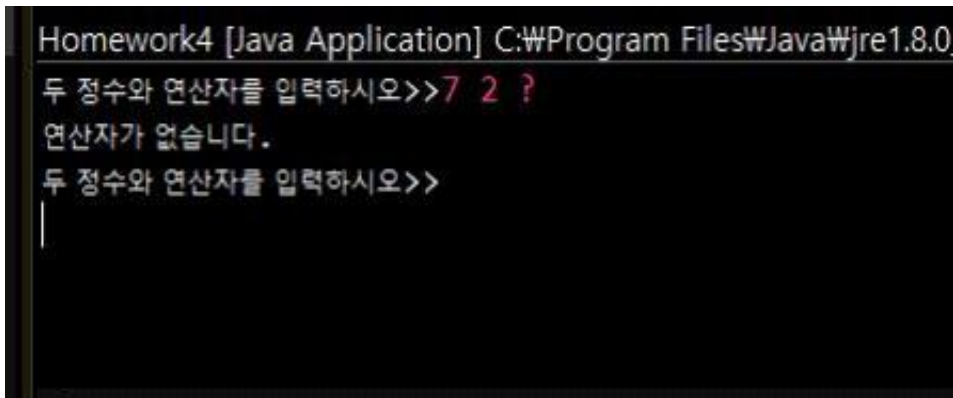
```
<terminated> Homework4 [Java Application] C:\Program Files\Java\jre
두 정수와 연산자를 입력하십시오>>7 2 +
|
|
```

```
<terminated> Homework4 [Java Application] C:\Program Files\Java
두 정수와 연산자를 입력하십시오>>7 2 -
|
|
```

```
<terminated> Homework4 [Java Application] C:\Program Files\
두 정수와 연산자를 입력하십시오>>7 2 *
|14
|
|
```

```
<terminated> Homework4 [Java Application] C:\P
두 정수와 연산자를 입력하십시오>>7 2 /
|
|
```

```
Homework4 [Java Application] C:\Program Files\Ja
두 정수와 연산자를 입력하십시오>>7 0 /
|다시 입력하세요!
|
|두 정수와 연산자를 입력하십시오>>
|
|
```



(그림을 문서에 포함, 글자처럼 취급 옵션, 잉크 절약과 잘 보이게 하기 위해 그림 반전)

작성된 프로그램을 실행한 결과 위와 같은 화면이 출력 되었다.

6. 고찰 (과제를 진행하면서 배운점 이나, 시행 착오 내용, 기타 느낀점)

과제 3과 문제가 비슷하여서 과제 4는 조금 수월하게 한 것 같다. Add, Sub, Mul, Div 클래스에 공통된 필드와 메소드가 존재하므로 추상클래스 Calc를 작성하고 Calc를 상속받도록 하였다. 클래스마다 Calc를 상속받고, setValue 메소드와 calculate 메소드를 오버라이딩 하였다. 메소드 이름과 매개변수 타입, 개수, 리턴타입을 동일하게 하였다. 혹시 다르게 해도 되나 하고 매개변수 타입을 다르게 해보았는데 오류가 발생하였다. 각각의 클래스마다 연산자에 맞게 calculate 메소드의 내용을 작성하였다. 확실히 추상클래스에 공통된 메소드를 작성해놓고, 상속과 오버라이딩을 통해 재작성하니까 머릿속에 정리가 잘 되고, 체계적으로 코딩을 한 것 같다. 기업이나 회사에 들어갔을 때도 교수님 말씀처럼 나 혼자 코딩하는 것이 아니라 수십명에서 수백명이 같이 코딩 작업을 하기 때문에 추상클래스나 인터페이스에 공통된 필드와 메소드를 작성해주고 코딩하는 것이 좋은 것 같다. 오버라이딩 부분이 약간 헷갈렸는데 직접 상속관계와 관련된 문제를 풀어보니 이해가 많이 된 것 같다. 오버라이딩과 오버로딩 부분을 헷갈리지 않도록 복습을 해야겠다.

7. 전체 소스코드 (글자크기 9에 줄간격을 120%로 유지하고 한 줄이 너무 길지 않게 작성)

```
import java.util.Scanner;

public abstract class Calc{
    abstract void setValue(int a, int b);
    abstract int calculate();
}

public class Add extends Calc{
    public Add() {
    }
    private int a,b;
    void setValue(int a, int b) {
        this.a=a; this.b=b;
    }
    int calculate() {
        return a+b;}
}

public class Sub extends Calc{
    public Sub() {
    }
    private int a,b;
    void setValue(int a, int b) {
        this.a=a; this.b=b;
    }
}
```

```

int calculate() {
return a-b;}
}
public class Mul extends Calc{
public Mul() {
}
private int a,b
void setValue(int a, int b) {
this.a=a; this.b=b;
}
int calculate() {
return a*b;}
}
public class Div extends Calc{
public Div() {
}
private int a,b
void setValue(int a, int b) {
this.a=a; this.b=b;
}
int calculate() {
if(b!=0) {return a/b;}
else return 0;} }
public class Homework4 {

public static void main(String[] args) {
int x, y;
Scanner scanner = new Scanner(System.in);
while(true){
System.out.print("두 정수와 연산자를 입력하시오>>");
x= scanner.nextInt();
y = scanner.nextInt();
char key = scanner.next().charAt(0);
switch(key) {
case '+':{
Add add = new Add();
add.setValue(x, y);
System.out.println(add.calculate());
break;
}
case '-':{
Sub sub =new Sub();
sub.setValue(x,y);
System.out.println(sub.calculate());
break;
}
case '*':{
Mul mul =new Mul();
mul.setValue(x,y);
System.out.println(mul.calculate());
break;
}
case '/':{
Div div =new Div();
div.setValue(x,y);
if(div.calculate()!=0) {
System.out.println(div.calculate());
break;}
else {System.out.println("다시 입력하세요!");
continue;}
}
default: {System.out.println("연산자가 없습니다.");
continue;}
}
break;}
}
}

```

(글자크기는 10으로 유지하고 줄간격도 160%를 유지할 것)