

자바 기초 문법

변수, 자료형

자바 기초 문법

- 자바프로그램 작성, 실행
- 변수, 대입문
- 키워드, 식별자
- 기본 자료형
- 상수 변수, 리터럴
- 형 변환
- 연산자 및 우선순위
- if, for, while, break, continue
- 배열
- 향상된 for문
- 이차원배열, 다차원배열
- 메소드

자바프로그램 작성, 컴파일, 실행

소스 코드 작성 (Test.java)

```
public class Test {  
    public static void main(String[] args) {  
        System.out.println("안녕하세요");  
    }  
}
```

컴파일 (Test.class 생성)

```
javac Test.java
```

실행 (Test.class 실행)

```
java Test  
안녕하세요
```

변수(variable)

변수

- 이름이 부착된 메모리 내 저장 공간

자료형 변수명

↓ ↓
int score ;

- 4바이트 정수 값을 저장할 수 있는 score라는 이름의 변수 선언문
- 변수 선언(declaration) 문장
- int는 4바이트 정수(integer) 자료형을 의미하는 키워드
- 자바에서 int 자료형(data type)은 4바이트 공간에 정수를 표현함
- score는 변수 이름
- 메모리 내 4바이트 공간 확보 후 score라는 변수 이름 부착
- 이 4바이트 공간에는 int 자료형(정수, integer)의 값만 저장할 수 있음

메모리 주소
(address)



8000

8004

8008

score

개념적 메모리 표현

변수(variable), 대입(assignment)

- 변수 선언(declaration) 문장

```
int score ;
```

```
score = 100 ;
```

- 정수형 변수 score에 정수 값 100을 대입
- 등호 기호는 좌변, 우변이 같다는 의미가 아님
- 등호 기호 =는 대입연산자(assignment operator)임
- 대입연산자는 연산자 오른쪽의 값을 연산자 왼쪽의 저장공간에 대입함
- 위 문장은 등호 오른쪽의 값 100을 등호 왼쪽의 변수 score에 대입(저장)

메모리 주소
(address)



8000

100

score

8004

8008

개념적 메모리 표현

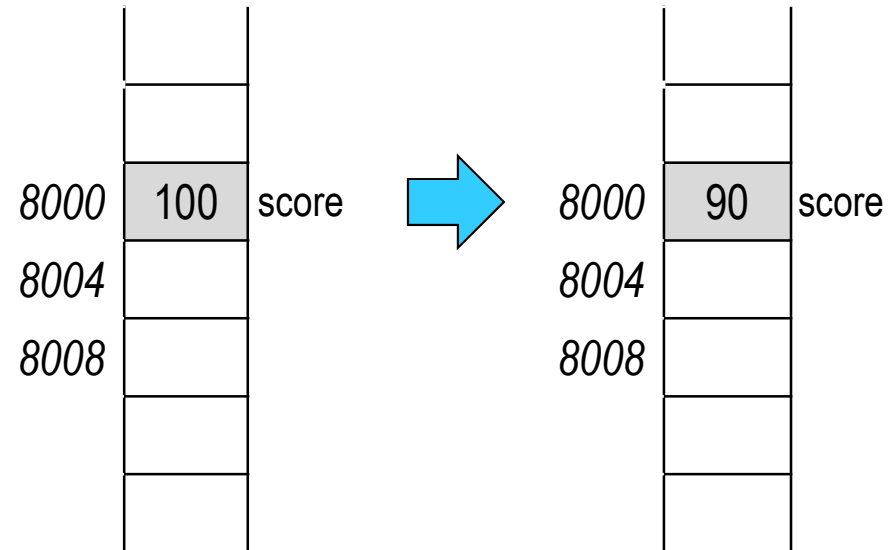
변수(variable), 대입(assignment)

```
int score ;
```

```
score = 100 ;
```

```
score = 90 ;
```

- score에 90을 대입
- score에 이전에 저장되어 있던 값 100이 90으로 변경됨

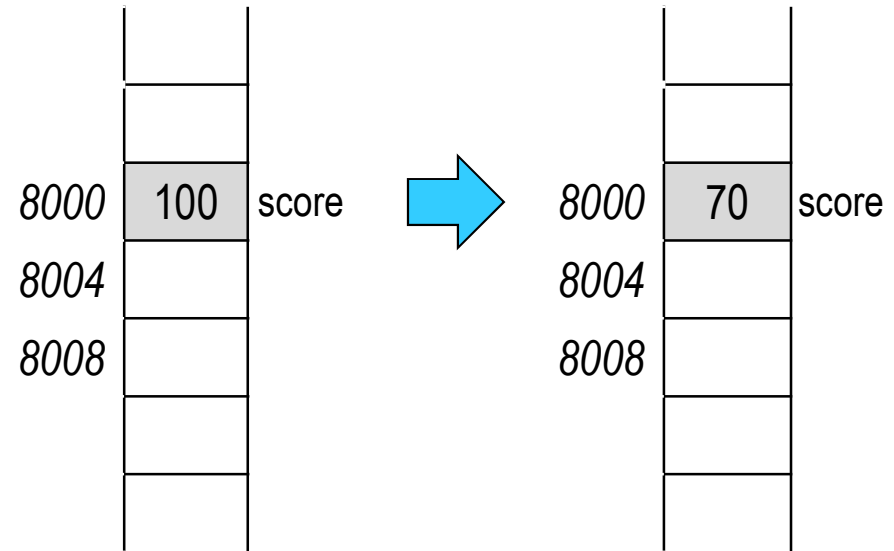


변수(variable), 대입(assignment)

```
int score ;
```

```
score = 100 ;
```

```
score = score - 30 ;
```



- score에 저장된 값 100에서 30을 감한 값 70을 변수 score에 대입(저장)

변수(variable), 대입(assignment)

score = 100 ;

- score라는 이름의 변수가 존재하지 않으므로 **오류 발생**
- 위 대입문 사용 이전에 score라는 변수가 선언되어야 있어야 함

변수(variable), 대입(assignment)

```
int    score ;  
score = 95.8 ;
```

- 변수 score는 정수형 변수이므로 정수 값을 저장할 수 있음
- 위 문장은 실수 값 95.8을 정수형 변수 score에 대입(저장)하는 것이므로 **오류 발생**

변수(variable), 대입(assignment)

```
int    score ;  
score = 100 ;
```

```
int    score = 100 ;
```

- 변수 선언문과 대입문을 각각 작성하는 대신 변수 선언과 대입을 하나의 문장으로 작성 가능

변수(variable), 대입(assignment)

- 8바이트 실수 값을 저장할 수 있는 weight라는 이름의 변수 선언문
- 변수 선언(declaration) 문장
- double은 8바이트 실수 자료형을 의미하는 키워드
- weight는 변수 이름
- 메모리 내 8바이트 공간 확보 후 weight라는 변수 이름 부착
- 이 8바이트 공간에는 double 자료형의 값만 저장할 수 있음

double weight ;

weight = 357.8 ;

- 대입연산자를 사용하여 실수형 변수 weight에 실수 값 357.8을 대입

변수(variable), 대입(assignment)

- 2바이트 문자 값을 저장할 수 있는 gender라는 이름의 변수 선언문
- 변수 선언(declaration) 문장
- char은 2바이트 문자(character) 자료형을 의미하는 키워드
- gender는 변수 이름
- 메모리 내 2바이트 공간 확보 후 gender라는 변수 이름 부착
- 이 2바이트 공간에는 char 자료형의 값만 저장할 수 있음

char gender ;

gender = '여' ;

- 대입연산자를 사용하여 문자형 변수 gender에 문자 값 '여' 를 대입

변수(variable), 대입(assignment)

- true 혹은 false의 불리언 값을 저장할 수 있는 isPassed라는 이름의 변수 선언문
- 변수 선언(declaration) 문장
- boolean은 불리언 자료형을 의미하는 키워드
- isPassed는 변수 이름
- 이 변수 공간에는 boolean 자료형의 값만 저장할 수 있음
- boolean 자료형의 값은 true 혹은 false 둘 중 하나임



```
boolean isPassed ;
```

```
isPassed = true ;
```



- 대입연산자를 사용하여 불리언 자료형 변수 isPassed에 불리언 값 true를 대입

변수(variable), 대입문(assignment)

```
int score;  
score = 86;  
score = score + 1;  
System.out.println(score);  
System.out.println(score+1);
```

← 변수 score에 저장된 값을 모니터에 출력

← 변수 score에 저장된 값에 1을 더한 값을 모니터에 출력

```
int x = 3;  
int y = 4;  
int z = 5, w = 6;  
int m = x;  
int n = x + y;
```

```
int z = 5;  
int w = 6;
```

← 정수형 변수 m을 선언한 후,
변수 x에 저장된 값을 변수 m에 대입(저장)

자바의 기본 자료형

```
public class Test {  
    public static void main(String[] args) {  
        int        age = 23;  
        double     eyeSight = 1.2;  
        char       gender = '여';  
        boolean    foreignerYesNo = false;
```

```
        System.out.println(age);
```

← 정수형 변수 age에 저장된 값을 출력

```
        System.out.println(eyeSight);
```

← 실수형 변수 eyeSight에 저장된 값을 출력

```
        System.out.println(gender);
```

← 문자형 변수 gender에 저장된 값을 출력

```
        System.out.println(foreignerYesNo);
```

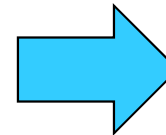
← 불리언 변수 foreignerYesNo에 저장된 값을 출력

```
    }  
}
```

System.out.print(), System.out.println(), System.out.printf()

System.out.print("Hello");	→ Hello를 화면에 출력
System.out.println("Hello");	→ Hello를 화면에 출력한 후 줄바꿈문자 출력
System.out.println();	→ 줄바꿈문자만 출력

```
public class Test {  
    public static void main(String[] args) {  
        int age = 23;  
        System.out.println(age);  
        System.out.print(age);  
        System.out.printf("나이는 %d", age);  
        System.out.println("끝");  
        System.out.println();  
        System.out.println("새로운 시작");  
    }  
}
```



실행 결과

```
23  
23나이는 23끝  
  
새로운 시작
```


초기화되지 않은 지역 변수 사용

```
public class Test {  
    public static void main(String[] args) {  
        int    score ;  
        System.out.println(score) ;  
    }  
}
```

- 초기화되지 않은 지역 변수(local variable)는 사용 시점에 **오류 발생**

변수(variable), 대입문(assignment)

다음 코드의 실행 결과는 무엇인가?

```
public class Test {  
    public static void main(String[] args) {  
        int score;  
        score=100;  
        score=95;  
        System.out.println(score);  
    }  
}
```

다음 코드의 오류는 무엇인가?

```
public class Test {  
    public static void main(String[] args) {  
        score=100;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        int score;  
        System.out.println(score);  
    }  
}
```

변수(variable), 대입문(assignment)

다음 각 코드의 오류는 무엇인가?

```
public class Test {  
    public static void main(String[] args) {  
        int score;  
        score=96.8;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        int score;  
        score=A;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        int score;  
        100 = score;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        int score;  
        score=100;  
        int score;  
        score=99;  
        System.out.println(score);  
    }  
}
```

자바 키워드, 식별자

참조: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/_keywords.html

자바 키워드(예약어, keywords, reserved words)

- abstract, continue, for, new, switch, assert, default, goto, package, synchronized, boolean, do, if, private, this, break, double, implements, protected, throw, byte, else, import, public, throws, case, enum, instanceof, return, transient, catch, extends, int, short, try, char, final, interface, static, void, class, finally, long, strictfp, volatile, const, float, native, super, while

식별자(identifier)

- 변수, 메소드, 클래스 등에 부여하는 이름
- 식별자 명명 규칙
 - ◆ 자바 키워드, true, false, null 사용 불가
 - ◆ 첫 문자로 숫자 사용 불가
 - ◆ 특수문자, 공백 사용 불가 (\$, _는 사용 가능)
 - ◆ 대소문자 구분 (case-sensitive)
 - ◆ 한글 사용 가능
 - ◆ 길이 제한 없음

```
public class Test {  
    public static void main(String[] args) {  
        int    score=10;  
        int    Score=20;  
        int    _score=30;  
        int    $score=40;  
    }  
}
```

자바 키워드, 식별자

다음 각 코드의 오류는 무엇인가?

```
public class Test {  
    public static void main(String[] args) {  
        int switch ;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        int 1score = 88 ;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        int math score = 100 ;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        int score@course = 100 ;  
    }  
}
```

자바의 자료형

자바의 자료형

- Primitive types (기본 자료형): 8개
 - ◆ byte, short, int, long, float, double, char, boolean
- Reference types (레퍼런스 자료형)
 - ◆ 배열 타입
 - ◆ 클래스 타입
 - ◆ 인터페이스 타입

자바의 기본 자료형

참조: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

자바의 기본 자료형(primitive data type)

- 정수 기본 자료형 → int
- 실수 기본 자료형 → double

기본 자료형		바이트 크기	범위	필드 디폴트 값
정수	byte	1	$-2^7 \sim 2^7-1$	0
	short	2	$-2^{15} \sim 2^{15}-1$	0
	int	4	$-2^{31} \sim 2^{31}-1$	0
	long	8	$-2^{63} \sim 2^{63}-1$	0L
실수	float	4	32-bit IEEE-754	0.0f
	double	8	64-bit IEEE-754	0.0d
문자	char	2	16-bit Unicode	'\u0000'
불리언	boolean		true, false	false

자바의 기본 자료형

다음 각 코드의 오류는 무엇인가?

```
public class Test {  
    public static void main(String[] args) {  
        int score = 95.8 ;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        int score = "A" ;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        int gender = 남 ;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        char gender = "여" ;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        int score = 87.0 ;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        boolean isTestPassed = True;  
    }  
}
```


자바의 기본 자료형

✚ 다음은 30000000000을 출력하는 코드이다.

- 어떤 오류가 있는가?
- 오류를 수정하시오.

```
public class Test {  
    public static void main(String[] args) {  
        int money = 30000000000; // 30억  
        System.out.println(money);  
    }  
}
```

상수 변수, 리터럴

상수 변수(constant variable)

- 최초 부여된 값 변경 불가 변수
- final 수식어 사용

```
final double PI = 3.14159;
```

```
class X {  
    static final double PI = 3.14159; // 필드  
}
```

리터럴(literal)

- 소스코드 내에 표현된 값
 - ◆ 예) 12, 12L, 12e-3, '김', 0xAB

리터럴

+ 정수 리터럴

+ 실수 리터럴

+ 문자 리터럴

● 예) '봄', 'A'

+ 불리언 리터럴

● true 혹은 false

+ 문자열(String) 리터럴

● 예) "서울", "A", "2", "3.4", "null", "true"

+ null 리터럴

● null

```
int      n = 12;  
long     m = 12L; // 12l  
int      v1 = 11; // 10진수 11  
int      v2 = 011; // 8진수 11 (10진수 9)  
int      v3 = 0x11; // 16진수 11 (10진수 17)  
int      v4 = 0b11; // 2진수 11 (10진수 3)
```

```
double   v = 12.0; // 12.0d, 12.0D  
float    w = 12.0F; // 12.0f  
double   x = 12E-3; // 12e-3, 0.012
```

```
char c1 = '한';  
char c2 = '\uD55C';  
char c3 = '\n';
```

```
boolean  b = true; // true 혹은 false
```

```
String   s1 = "대한민국";  
String   s2 = null;
```

문자열(스트링,String) 리터럴

String 리터럴

- 큰따옴표로 둘러싸인 문자(들)의 나열

```
public class Test {  
    public static void main(String[] args) {  
        // int    score = null; // 오류  
        String   country=null;  
        System.out.println(country);  
        country="대한민국";  
        System.out.println(country);  
  
        String   memo="경성대학교 공과대학\n소프트웨어학과";  
        System.out.println(memo);  
        memo=country;  
        System.out.println(memo);  
    }  
}
```

기초 – 실습 A

다음은 어떤 코드와 그 실행 결과이다. 이 코드를 완성하시오.

```
public class Test {  
    public static void main(String[] args) {  
  
        System.out.println(name); // 선수 이름  
        System.out.println(age); // 선수 나이  
        System.out.println(weight); // 선수 체중  
        System.out.println(gender); // 선수 성별  
        System.out.println(dopingTestPassed); // 선수 도핑 테스트 결과 (불리언)  
    }  
}
```

실행결과

```
김철수  
21  
65.7  
남  
true
```

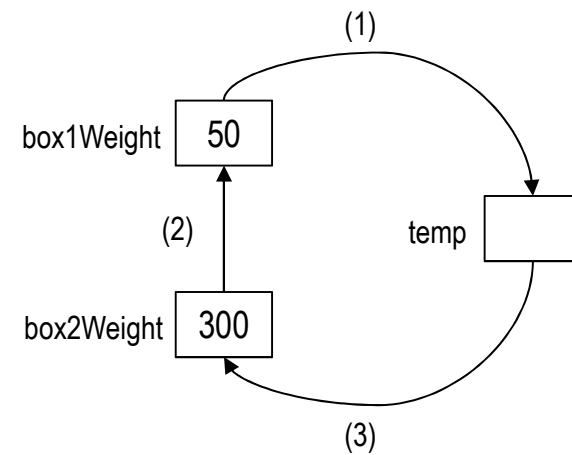
기초 - 실습 B

다음은 두 변수에 저장된 값을 교환한 후 출력하는 코드이다. 이 코드의 아래 빈 곳을 완성하시오.

```
public class Test {  
    public static void main(String[] args) {  
        int box1Weight = 50;  
        int box2Weight = 300;  
  
  
        System.out.println(box1Weight);  
        System.out.println(box2Weight);  
    }  
}
```

실행결과

300
50



기초 - 실습 C

다음은 세 점수를 변수에 대입한 후 그 평균을 출력하는 코드이다.

- 아래 (1), (2), (3) 문장들을 하나의 문장으로 축약하시오
- 아래 (4), (5), (6) 문장들을 하나의 문장으로 축약하시오

```
public class Test {  
    public static void main(String[] args) {  
        int    mathScore=79;           // (1)  
        int    langScore=86;          // (2)  
        int    engScore=95;           // (3)  
        int    total=mathScore+langScore+engScore; // (4)  
        double average=total/3.0;     // (5)  
        System.out.println(average);  // (6)  
    }  
}
```

기초 – 실습 D

- ✚ 다음 (1), (2), (3), (4), (5) 각 문장의 오류가 무엇인지 기술하십시오
- 문장 (3)은 변수 z에 0.5를 저장하려 함

```
public class Test {  
    public static void main(String[] args) {  
        int    x=3.0 ;           // (1)  
        char   y="A" ;           // (2)  
        double z=1/2 ;           // (3)  
        int    w=3000000000 ;    // (4)  
        System.out.println('South Korea'); // (5)  
    }  
}
```


문자열 연결(concatenation) 연산자 +

문자열 연결 연산자(operator) +

- 피연산자(operand) 중 하나가 문자열일 경우 문자열이 아닌 피연산자를 문자열로 변환 후 문자열연결 수행

```
public class Test {  
    public static void main(String[] args) {  
        String v = "부산" + "해운대"; // String concatenation  
        System.out.println(v); // 실행결과 => 부산해운대  
  
        String x = v + "관광";  
        System.out.println(x);  
  
        System.out.println( 123 + 456 ); // 실행결과 => 579  
        System.out.println( "123" + "456" ); // 실행결과 => 123456  
        System.out.println( 123 + "456" ); // 실행결과 => 123456  
        System.out.println( "123" + 456 ); // 실행결과 => 123456  
    }  
}
```

기초 - 실습 E

다음은 어떤 코드와 그 실행 결과이다. 이 코드를 완성하시오.

```
public class Test {  
    public static void main(String[] args) {  
        String name = "이영희";  
        int score = 95;  
        System.out.println(           );  
    }  
}
```

실행결과

이름: 이영희
점수: 95점

기초 - 실습 F

다음은 변수 x, y에 저장된 값의 덧셈, 뺄셈, 곱셈, 나눗셈 결과 값을 아래 실행결과와 같이 출력하는 코드이다. 이 코드를 완성하시오.

```
public class Test {  
    public static void main(String[] args) {  
        double    x = 40;  
        double    y = 8;  
  
  
  
  
  
  
    }  
}
```

실행결과

```
48.0  
32.0  
320.0  
5.0
```

문자열을 숫자로 변환

```
public class Test {  
    public static void main(String[] args) {  
        int score;  
        score = Integer.parseInt("95") ; // "+95", "-95"  
        score = score + 5;  
        System.out.println(score);  
  
        double height;  
        height = Double.parseDouble("185.7") ; // "+185.7", "-185.7"  
        height = height - 0.1;  
        System.out.println(height);  
    }  
}
```

Integer.parseInt("95")

- 숫자 문자열 "95"에 대응하는 정수(int) 95를 반환하는 메소드 호출문

Double.parseDouble("185.7")

- 숫자 문자열 "185.7"에 대응하는 실수(double) 185.7을 반환하는 메소드 호출문

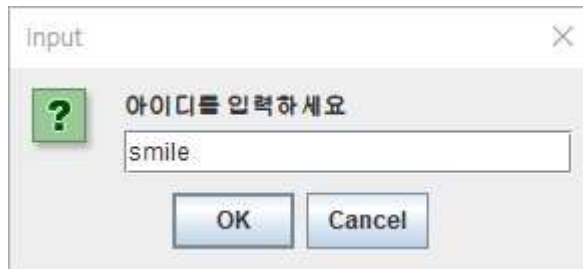
다이얼로그 박스 기반 입출력

이클립스 import 문장 자동 생성 단축키 → CTRL+SHIFT+O

```
import javax.swing.JOptionPane;
public class Test {
    public static void main(String[] args) {
        String id = JOptionPane.showInputDialog("아이디를 입력하세요");
        String message = id + "님, 반갑습니다.";
        JOptionPane.showMessageDialog(null, message);
    }
}
```

JOptionPane.showInputDialog("아이디를 입력하세요")

- 아래와 같은 입력 창 표시하는 메소드 호출문
- OK 버튼 클릭 시 입력한 내용을 문자열로 반환
- Cancel 버튼 클릭 시 null 반환



JOptionPane.showMessageDialog(null, message)

- 변수 message에 저장된 문자열을 아래와 같은 메시지 창에 출력하는 메소드 호출문

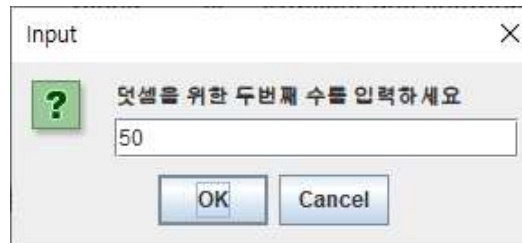
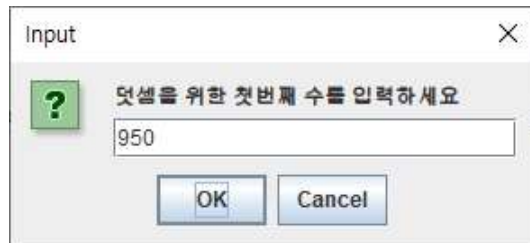


다이얼로그 박스 기반 입출력

```
import javax.swing.JOptionPane;
```

```
public class Test {  
    public static void main(String[] args) {  
        String s1 = JOptionPane.showInputDialog("덧셈을 위한 첫번째 수를 입력하세요");  
        String s2 = JOptionPane.showInputDialog("덧셈을 위한 두번째 수를 입력하세요");  
        int n1 = Integer.parseInt(s1);  
        int n2 = Integer.parseInt(s2);  
        String message = "두 수의 합은 " + (n1+n2) + "입니다.";  
  
        JOptionPane.showMessageDialog(null, message);  
    }  
}
```

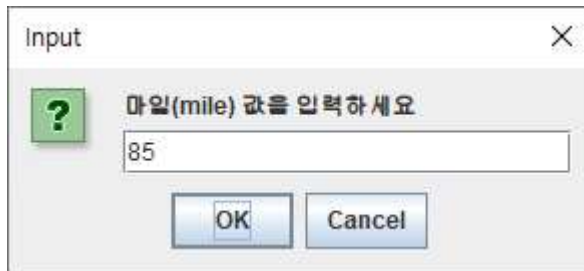
- (n1+n2)에서 괄호를 제거한 후 실행해 보시오
- 괄호를 제거했을 경우의 실행 결과를 설명해 보시오




기초 - 실습 G

다음과 같이 동작하는 자바 코드를 작성하시오

- 1마일 \approx 1.609킬로미터



Input dialog box with a title bar containing a close button (X). The dialog contains a green question mark icon, the text "마일(mile) 값을 입력하세요", a text input field containing "85", and two buttons labeled "OK" and "Cancel".



Message dialog box with a title bar containing a close button (X). The dialog contains a blue information icon, the text "85 마일(mile)은 136.765 킬로미터(km)입니다.", and one button labeled "OK".

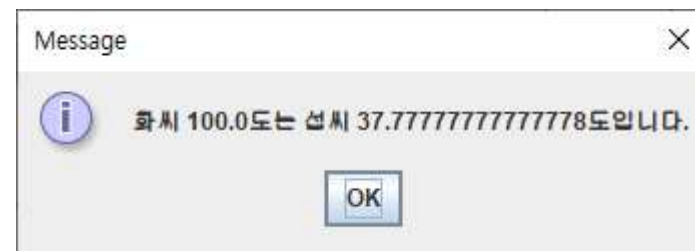
기초 - 실습 H

다음과 같이 동작하는 자바 코드를 작성하시오

- 화씨 온도(F)의 섭씨 온도(C) 변환 공식: $C = (F - 32) \times \frac{5}{9}$



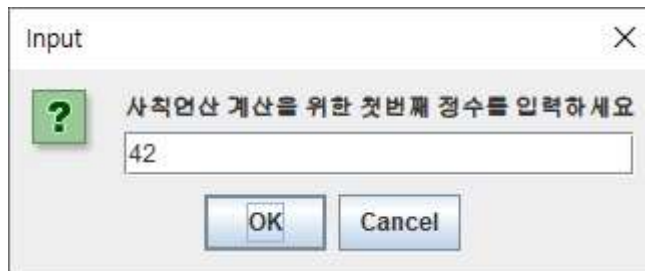
Input dialog box with a green question mark icon. The text reads "화씨 온도 값을 입력하세요" (Enter Fahrenheit temperature value). The input field contains the value "100". There are "OK" and "Cancel" buttons at the bottom.



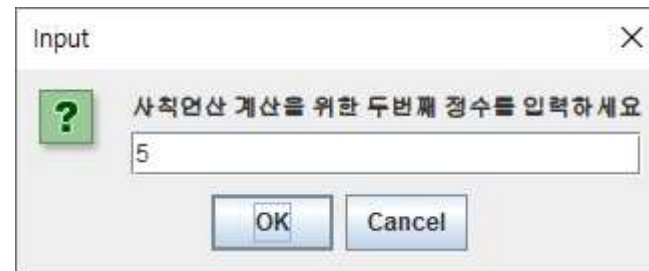
Message dialog box with a blue information icon. The text reads "화씨 100.0도는 섭씨 37.7777777777778도입니다." (Fahrenheit 100.0 degrees is Celsius 37.7777777777778 degrees). There is an "OK" button at the bottom.

기초 - 실습 I

- 다음은 두 수를 입력 받아 그 덧셈, 뺄셈, 곱셈, 나눗셈 결과를 출력하는 코드의 실행 예시이다. 다음과 같이 동작하는 자바 코드를 작성하시오



Input dialog box titled "Input" with a close button (X). It contains a green question mark icon and the text "사칙연산 계산을 위한 첫번째 정수를 입력하세요". The input field contains the number "42". There are "OK" and "Cancel" buttons at the bottom.



Input dialog box titled "Input" with a close button (X). It contains a green question mark icon and the text "사칙연산 계산을 위한 두번째 정수를 입력하세요". The input field contains the number "5". There are "OK" and "Cancel" buttons at the bottom.




Message dialog box titled "Message" with a close button (X). It contains an information icon (i) and the following calculation results:
 $42.0 + 5.0 = 47.0$
 $42.0 - 5.0 = 37.0$
 $42.0 \times 5.0 = 210.0$
 $42.0 / 5.0 = 8.4$
There is an "OK" button at the bottom.

References

 <http://docs.oracle.com/javase/7/docs/api/>

 <https://docs.oracle.com/javase/tutorial/java/>

 김윤명. (2008). 뇌를 자극하는 Java 프로그래밍. 한빛미디어.

 남궁성. 자바의 정석. 도우출판.

 황기태, 김효수 (2015). 명품 Java Programming. 생능출판사.