

## 2차시 정리

```
// public로 선언하면 다른 클래스에서 접근 가능
public class HelloJava {
    public static void main(String[] args) {

        //출력
        // println() 메소드는 여러 타입의 데이터 출력 가능하며 출력 후 다음 행으로 커서 이동
        System.out.println( "Hello World!!" ); //

    }
}
```

```
// 출력
// Hello World!!
```

```
// 식별자란 클래스, 변수, 상수, 메소드 등에 붙이는 이름
// '_', '$'를 제외한 특수 문자, 공백 또는 탭은 식별자로 사용 불가
// 유니코드 문자와 한글은 사용가능하지만 자바언어 키워드는 사용 불가
// 길이 제한 없으며, 대소문자는 구별하고, 첫번째 문자로 숫자는 불가능 하지만 '_', '$'는 사용가능
// 자바 키워드
// abstract, boolean, assert, catch, extends, goto, implements, instanceof
// native, protected, strictf, super, synchronized, this, transient, volatile
```

```
// 식별자란 클래스, 변수, 상수, 메소드 등에 붙이는 이름
// '_', '$'를 제외한 특수 문자, 공백 또는 탭은 식별자로 사용 불가
// 유니코드 문자와 한글은 사용가능하지만 자바언어 키워드는 사용 불가
// 길이 제한 없으며, 대소문자는 구별하고, 첫번째 문자로 숫자는 불가능 하지만 '_', '$'는 사용가능
// 자바 키워드
// abstract, boolean, assert, catch, extends, goto, implements, instanceof
// native, protected, strictf, super, synchronized, this, transient, volatile
```

```
// 클래스이 이름
// 첫번째 문자는 대문자 또는 각 단어의 첫문자만 대문자로 생성
```

```
// 상수 이름
// 모든 문자를 대문자로 표시
```

```
// 자바의 데이터 타입
// 기본 타입 : 8개
// boolean(1)
// char(2)
// byte(1), short(2), int(4), long(8)
// float(4), double(8)
// 래퍼런스(참조) 타입 : 1개(용도 3가지)
// 배열(array), 클래스(class), 인터페이스(interface)
```

```
// 문자열
// 문자열은 기본타입이 아니며 String 클래스로 문자열 표현
```

```
// 변수
// 프로그램 실행 중에 값을 임시 저장하기 위한 공간, 데이터 타입에서 정한 크기의 메모리 할당
```

```
// 리터럴
// 프로그램에서 직접 표현한 값
// 정수, 실수, 문자, 논리, 문자열 리터럴 존재
```

```
// 정수 리터럴
// 10진수 -> 15
```

```

// 2진수 -> 0b0101 : 0b로 시작하면 2진수, 십진수로 5
// 8진수 -> 015 : 0으로 시작하면 8진수, 십진수로 13
// 16진수 -> 0x15 : 0x로 시작하면 16진수, 십진수로 21
// long 타입 리터럴은 숫자 뒤에 L 또는 l을 붙여 표시, 예) long g = 24L;

// 실수 시터럴
// double 타입으로 컴파일 하며 숫자 뒤에 f, d를 명시적으로 붙이기도 함

// 문자 리터럴
// 단일 인용부호 "로 문자 표현
// 역슬래시\ 다음에 4자리 16진수는 2바이트의 유니코드를 나타낸다.

// 특수문자 리터럴
// 백슬래시(\)로 시작

// 논리 리터럴
// true, false 2개가 존재하며 boolean 타입 변수에 치환하거나 조건문에 이용
// 1.0 을 자바에선 참, 거짓으로 사용 불가

// null 리터럴
// 레퍼런스 에 대입 사용

// 문자열 리터럴(스트링 리터럴)
// 이중 인용부호로 묶어 표현

// 숫자 리터럴
// 일정한 범위에서 언더스코어(언더바, _) 허용
// 리터럴의 끝, 소수점 앞뒤, 자료형표시 앞, 0x 중간뒤 에서는 사용할 수 없다.

// var 키워드
// 지역 변수의 선언에만 사용가능 예) var price = 200; -> price는 int 타입으로 결정
// 변수 선언문에 반드시 초기값 지정해야 한다. 예) var name; -> 컴파일 오류

// 상수
// final 키워드를 사용하여 선언하며, 선언시 초기값 지정해야 하고 실행중에 값을 변경할 수 없다.

public class CircleArea {
    public static void main(String[] args) {

        // 상수로 선언
        // 원주율을 상수로 선언
        // 변수이름은 대문자로 선언하며, final를 붙여서 선언한다.
        final double PI = 3.14;

        // double 자료형으로 선언한다.
        double radius = 10.0; // 원의 반지름
        double circleArea = radius*radius*PI; // 원의 면적 계산

        // 원의 면적을 화면에 출력한다.
        // System.out.println를 사용하여 출력
        System.out.println("원의 면적 = " + circleArea);
    }
}

// 출력
// 원의 면적 = 314.0

```

```
// 자동 타입 변환(casting)
// 작은 타입은 큰 타입으로 자동 변환
// 피 연산자의 데이터 타입에 맞춰서 연산의 사용되는 값들이 자동 캐스팅 된다.

// 자동 타입 변환이 안되는 경우 : 큰 타입이 작은 타입으로 변환할 때
// 강제 타입 변환 : (변환될 타입)을 앞에 붙여서 강제 변환, 데이터 누락의 위험 있음
```

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

        // 1바이트와 4바이트 정수형 변수 선언
        byte b = 127;
        int i = 100;

        // byte 형은 -128에서 127 사이의 수로 반환
        // 범위가 오버되는 값이면 -256을 하면서 데이터 값을 구한다.
        System.out.println(b+i);
        System.out.println(10/4); // 정수형을 나누는 것이므로 결과도 정수형이다.
        System.out.println(10.0/4); // 실수형을 나누는 것이므로 결과도 실수형이다.
        System.out.println((char)0x12340041); // 강제타입변환의 결과로 0x41이 된다.
        System.out.println((byte)(b+i));
        System.out.println((int)2.9 + 1.8);
        System.out.println((int)(2.9 + 1.8));
        System.out.println((int)2.9 + (int)1.8);
    }
}

// 출력
// 227
// 2
// 2.5
// A
// -29
// 3.8
// 4
// 3
```

```

// System.in : 표준 입력 스트림
// 입력 값을 바이트(문자 아님)로 리턴
// 따라서 키 값을 바이트 데이터로 넘겨주므로 응용프로그램이 문자 정보로 변환해야 함

// java.util.Scanner 패키지에 Scanner 클래스를 사용하여 사용
// 입력되는 키 값을 공백으로 구분되는 아이템 단위로 읽음

// Scanner클래스에 내장 메서드
// .next() : String형으로 다음 토큰을 문자열로 리턴
// .nextByte() : byte형으로 다음 토큰을 byte 타입으로 리턴
// .nextShort() : Short형으로 다음 토큰을 Short 타입으로 리턴
// .nextInt() : Int형으로 다음 토큰을 Int 타입으로 리턴
// .nextLong() : Long형으로 다음 토큰을 Long 타입으로 리턴
// .nextFloat() : Float형으로 다음 토큰을 Float 타입으로 리턴
// .nextDouble() : Double형으로 다음 토큰을 Double 타입으로 리턴
// .nextBoolean() : Boolean형으로 다음 토큰을 Boolean 타입으로 리턴
// .nextLine() : String형으로 '\n'을 포함하는 한 라인을 읽고 '\n'을 버리고 리턴
// .close() : void형으로 Scanner에 사용 종료

import java.util.Scanner;

public class ScannerEx {
    public static void main(String args[]) {

        System.out.println("이름, 도시, 나이, 체중, 독신 여부를 빈칸으로 분리하여 입력하세요");
        Scanner scanner = new Scanner(System.in);

        String name = scanner.next(); // 문자열 읽기
        System.out.print("이름은 " + name + ", ");

        String city = scanner.next(); // 문자열 읽기
        System.out.print("도시는 " + city + ", ");

        int age = scanner.nextInt(); // 정수 읽기
        System.out.print("나이는 " + age + "살, ");

        double weight = scanner.nextDouble(); // 실수 읽기
        System.out.print("체중은 " + weight + "kg, ");

        String single = scanner.next(); // 문자열 읽기
        System.out.println("독신 여부는 " + single + "입니다.");

        scanner.close(); // scanner 닫기
    }
}

// 출력
// 이름, 도시, 나이, 체중, 독신 여부를 빈칸으로 분리하여 입력하세요
// 진석 서울 24 88 No
// 이름은 진석, 도시는 서울, 나이는 24살, 체중은 88.0kg, 독신 여부는 No입니다.

```

```

// 연산자 우선순위
// 높은
// ++ --
// +(부호) -(부호) ++ --
// 형변환
// * / %
// +(덧셈) -(뺄셈)
// << >> >>>
// <> <= >= instanceof
// == !=
// & (비트 AND)
// ^ (비트 XOR)
// | (비트 OR)
// && (논리 AND)
// || (논리 OR)
// ? : (조건)
// += -= *= /= %= &= ^= |= <<= >>= >>>=
// 낮음

// % : 나머지
// / : 몫

```

```
import java.util.Scanner;
```

```

public class ArithmeticOperator {
    public static void main(String[] args) {

        // 사용자 입력값 저장
        Scanner scanner = new Scanner(System.in);
        System.out.print("경과시간을 초단위로 입력하세요: ");

        // 변수 초기화
        int time = scanner.nextInt(); // 정수 입력
        int second = time % 60; // 60으로 나눈 나머지는 초
        int minute = (time / 60) % 60; // 60으로 나눈 몫을 다시 60으로 나눈 나머지는 분
        int hour = (time / 60) / 60; // 60으로 나눈 몫을 다시 60으로 나눈 몫은 시간

        // 출력
        System.out.print(time + "초는 ");
        System.out.print(hour + "시간, ");
        System.out.print(minute + "분, ");
        System.out.println(second + "초입니다.");

        // 사용자 입력 종료
        scanner.close();
    }
}

// 결과
// 경과시간을 초단위로 입력하세요: 33567
// 33567초는 9시간, 19분, 27초입니다.

```

```

// 증감연산자
// ++, --
// a++ : 증가 전의 값 변환하고 a를 1증가
// a-- : 감소 전의 값 변환하고 a를 1감소
// ++a : a를 1증가하고 증가된 값 변환
// --a : a를 1감소하고 감소한 값 반환

public class AssignmentIncDecOperator {
    public static void main(String[] args) {

        // 변수 선언
        int a=3, b=3, c=3;

        // 대입 연산자 사례
        a += 3; // a=a+3 = 6
        b *= 3; // b=b*3 = 9
        c %= 2; // c=c%2 = 1

        // 출력
        System.out.println("a=" + a + ", b=" + b + ", c=" + c);
        int d=3;

        // 증감 연산자 사례
        a = d++; // a=3, d=4
        System.out.println("a=" + a + ", d=" + d);
        a = ++d; // d=5, a=5
        System.out.println("a=" + a + ", d=" + d);
        a = d--; // a=5, d=4
        System.out.println("a=" + a + ", d=" + d);
        a = --d; // d=3, a=3
        System.out.println("a=" + a + ", d=" + d);
    }
}

// 출력
// a=6, b=9, c=1
// a=3, d=4
// a=5, d=5
// a=5, d=4
// a=3, d=3

```

```

public class LogicalOperator {
    public static void main(String[] args) {

        // 비교 연산
        System.out.print("'a' > 'b' : ");
        System.out.println('a' > 'b');

        System.out.print("3 >= 2 : ");
        System.out.println(3 >= 2);

        System.out.print("-1 < 0 : ");
        System.out.println(-1 < 0);

        System.out.print("3.45 <= 2 : ");
        System.out.println(3.45 <= 2);

        System.out.print("3 == 2 : ");
        System.out.println(3 == 2);

        System.out.print("3 != 2 : ");
        System.out.println(3 != 2);

        System.out.print("! (3 != 2) : ");
        System.out.println(!(3 != 2));

        // 비교 연산과 논리 연산 복합
        System.out.print("(3 > 2) && (3 > 4) : ");
        System.out.println((3 > 2) && (3 > 4));

        System.out.print("(3 != 2) || (-1 > 0) : ");
        System.out.println((3 != 2) || (-1 > 0));

        System.out.print("(3 != 2) ^ (-1 > 0) : ");
        System.out.println((3 != 2) ^ (-1 > 0));

    }
}

// 출력
// 'a' > 'b' : false
// 3 >= 2 : true
// -1 < 0 : true
// 3.45 <= 2 : false
// 3 == 2 : false
// 3 != 2 : true
// !(3 != 2) : false
// (3 > 2) && (3 > 4) : false
// (3 != 2) || (-1 > 0) : true
// (3 != 2) ^ (-1 > 0) : true

```

```
// 조건 연산자 ?
```

```
// (condition) ? (opr2) : (opr3)
```

```
// 세 개의 피연산자로 구성된 삼항 연산자
```

```
// condition이 참이면 opr2를 반환, 거짓이면 opr3 반환
```

```
public class TernaryOperator {  
    public static void main(String[] args) {  
  
        // 변수 선언  
        int a = 3, b = 5;  
  
        // 조건연산자 사용  
        System.out.println("두 수의 차는 " + ((a>b)?(a-b):(b-a)));  
    }  
}
```

```
// 결과
```

```
// 두 수의 차는 2
```



```

// 비트 논리 연산과 비트 시프트 연산

// 비트 연산 : 비트 논리 연산 과 비트 시프트 연산이 있음

// 비트 논리 연산
// 비트끼리 AND(&), OR(|), XOR(^), NOT(~) 연산
// 1 : 참
// 0 : 거짓

// 비트 시프트 연산
// 비트를 오른쪽이나 왼쪽으로 이동
// a >> b : 산술적 오른쪽 시프트 -> 결과값 : a / (b^2)
// a의 각 비트를 오른쪽으로 b번 시프트 한다. 최상위 비트의 빈자리는 시프트 전의 최상위 비트로 다시 채운다.
// a >>> b : 논리적 오른쪽 시프트 -> 결과값 : a / (b^2)
// a의 각 비트를 오른쪽으로 b번 시프트 한다. 최상위 비트의 빈자리는 항상 0으로 채운다.
// a << b :: 산술적 왼쪽 시프트 -> 결과값 : a * (b^2)
// a의 각 비트를 왼쪽으로 b번 시프트 한다. 최하위 비트의 빈자리는 항상 0으로 채운다.

public class BitOperator {
    public static void main(String[] args) {

        short a = (short)0x55ff;
        short b = (short)0x00ff;

        // 비트 논리 연산
        // printf("%x\n", Value) : Value값을 16진수 형식으로 출력
        System.out.println("[비트 연산 결과]");
        System.out.printf("%04x\n", (short)(a & b)); // 비트 AND
        System.out.printf("%04x\n", (short)(a | b)); // 비트 OR
        System.out.printf("%04x\n", (short)(a ^ b)); // 비트 XOR
        System.out.printf("%04x\n", (short)(~a)); // 비트 NOT
        byte c = 20; // 0x14
        byte d = -8; // 0xf8

        // 비트 시프트 연산
        System.out.println("[시프트 연산 결과]");
        System.out.println(c <<2); // c를 2비트 왼쪽 시프트
        System.out.println(c >>2); // c를 2비트 오른쪽 시프트. 0 삽입
        System.out.println(d >>2); // d를 2비트 오른쪽 시프트. 1 삽입
        System.out.printf("%x\n", (d >>>2)); // d를 2비트 오른쪽 시프트. 0 삽입
    }
}

// 출력
// [비트 연산 결과]
// 00ff
// 55ff
// 5500
// aa00
// [시프트 연산 결과]
// 80
// 5
// -2
// 3fffffff

```

// if 문

import java.util.Scanner;

```
public class SuccessOrFail {
    public static void main(String[] args) {
        // 사용자 입력 함수
        Scanner scanner = new Scanner(System.in);
        System.out.print("점수를 입력하시오: ");

        // 입력한 값을 변수로 입력
        int score = scanner.nextInt();

        // 조건문
        if (score >= 80)
            System.out.println("축하합니다! 합격입니다.");

        // 사용자 입력 함수 종료
        scanner.close();
    }
}
```

// 출력

// 점수를 입력하시오: 100

// 축하합니다! 합격입니다.

// 조건문 if-else

import java.util.Scanner;

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

        // 사용자 입력 함수
        Scanner num = new Scanner(System.in);
        System.out.print("수를 입력하시오: ");

        // 변수 입력
        int number = num.nextInt();

        // 조건문
        if (number % 3 == 0)
            System.out.println("3의 배수입니다.");
        else
            System.out.println("3의 배수가 아닙니다.");

        num.close();
    }
}
```

// 출력

// 수를 입력하시오: 123

// 3의 배수입니다.

// 수를 입력하시오: 121

// 3의 배수가 아닙니다.

```
// 다중 if-else
// 조건문이 너무 많은 경우 switch 문 사용 권장

import java.util.Scanner;

public class Grading {
    public static void main(String[] args) {

        // 사용자 입력 함수
        Scanner scanner = new Scanner(System.in);
        System.out.print("점수를 입력하세요(0~100): ");

        // 변수 저장
        int score = scanner.nextInt(); // 점수 읽기
        char grade;

        // 조건문
        if(score >= 90) // score가 90 이상
            grade = 'A';
        else if(score >= 80) // score가 80 이상 90 미만
            grade = 'B';
        else if(score >= 70) // score가 70 이상 80 미만
            grade = 'C';
        else if(score >= 60) // score가 60 이상 70 미만
            grade = 'D';
        else // score가 60 미만
            grade = 'F';

        // 출력
        System.out.println("학점은 " + grade + "입니다.");

        // 사용자 입력 함수 종료
        scanner.close();
    }
}

// 출력
// 점수를 입력하세요(0~100): 88
// 학점은 B입니다.

// 점수를 입력하세요(0~100): 59
// 학점은 F입니다.
```

```
// 중첩 if-else문 사례
```

```
import java.util.Scanner;
```

```
public class Nestedif {
```

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

```
        // 사용자 입력 객체 생성
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        // 점수 입력 후 변수에 대입
```

```
        System.out.print("점수를 입력하세요(0~100): ");
```

```
        int score = scanner.nextInt();
```

```
        // 학년 입력 후 변수에 대입
```

```
        System.out.print("학년을 입력하세요(1~4): ");
```

```
        int year = scanner.nextInt();
```

```
        // 조건문
```

```
        if(score >= 60) { // 60점 이상
```

```
            if(year != 4)
```

```
                System.out.println("합격!"); // 4학년 아니면 합격
```

```
            else if(score >= 70)
```

```
                System.out.println("합격!"); // 4학년이 70점 이상이면 합격
```

```
            else
```

```
                System.out.println("불합격!"); // 4학년이 70점 미만이면 불합격
```

```
        }
```

```
        else // 60점 미만 불합격
```

```
            System.out.println("불합격!");
```

```
        // 사용자 입력 함수 종료
```

```
        scanner.close();
```

```
    }
```

```
}
```

```
// 출력
```

```
// 점수를 입력하세요(0~100): 100
```

```
// 학년을 입력하세요(1~4): 3
```

```
// 합격!
```

```

// switch문
// case 와 default 부분으로 나뉜다.
// case문의 값은 문자, 정수, 문자열 리터럴만 허용
// 실수 리터럴, 수식은 허용되지 않음

// 각 문단에 break가 필수적으로 있어야 각 케이스를 잘 반환할 수 있음

import java.util.Scanner;

public class GradingSwitch {
    public static void main(String[] args) {

        // 사용자 입력 함수 선언
        Scanner scanner = new Scanner(System.in);

        // 변수 입력
        System.out.print("점수를 입력하세요(0~100): ");
        int score = scanner.nextInt();

        char grade;

        switch (score/10) {
            case 10: // score = 100
            case 9: // score는 90~99
                grade = 'A';
                break;
            case 8: // score는 80~89
                grade = 'B';
                break;
            case 7: // score는 70~79
                grade = 'C';
                break;
            case 6: // score는 60~69
                grade = 'D';
                break;
            default: // score는 59 이하
                grade = 'F';
        }
        // 출력
        System.out.println("학점은 " + grade + "입니다");

        // 사용자 입력 함수 종료
        scanner.close();
    }
}

// 출력
// 점수를 입력하세요(0~100): 89
// 학점은 B입니다

```

```

package project9011;
import java.util.Scanner;

public class project9011 {
    public static void main(String[] args) {
        // 사용자 입력 함수 선언
        Scanner scanner = new Scanner(System.in);

        // 변수 선언 및 초기화
        System.out.print("이름과 국어, 영어, 수학의 점수를 차례로 입력해주세요(0~100): ");
        String name = scanner.next();
        int kor = scanner.nextInt(); // 정수로 점수 읽기
        int eng = scanner.nextInt(); // 정수로 점수 읽기
        int mat = scanner.nextInt(); // 정수로 점수 읽기

        int score = kor + eng + mat;
        float avg = (float)score / 3; // 평균을 낼때 소수점도 중요하므로 실수형으로 캐스팅한다.
        char grade;

        // 학점 계산
        switch ((int)avg/10) { // case는 실수형 리터럴을 인식하지 못하므로 정수형으로 바꿔준다.
            case 10: // score = 100
            case 9: // score는 90~99
                grade = 'A';
                break;
            case 8: // score는 80~89
                grade = 'B';
                break;
            case 7: // score는 70~79
                grade = 'C';
                break;
            case 6: // score는 60~69
                grade = 'D';
                break;
            default: // score는 59 이하
                grade = 'F';
        }

        // 출력문
        System.out.println("귀하의 성적은 다음과 같습니다.");
        System.out.print("이름: " + name + "\n수학점수: " + kor + "점, ");
        System.out.print("영어점수: " + eng + "점, 수학점수: " + mat + "점");
        System.out.println("총점: " + score + "점, 평균: " + avg + "점, 학점은 "+grade+" 입니다");

        // 사용자 입력 함수 종료
        scanner.close();
    }
}

// 출력
// 이름과 국어, 영어, 수학의 점수를 차례로 입력해주세요(0~100): 진석 79 69 97
// 귀하의 성적은 다음과 같습니다.
// 이름: 진석
// 수학점수: 79점, 영어점수: 69점, 수학점수: 97점
// 총점: 245점, 평균: 81.666664점, 학점은 B 입니다

```

```
import java.util.Scanner;

public class CoffeePrice {
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("(메뉴 : 에스프레소, 카푸치노, 카페라떼, 아메리카노)");
        System.out.print("무슨 커피 드릴까요? ");
        String order = scanner.next();

        int price=0;

        switch (order) {
            case "에스프레소":
            case "카푸치노":
            case "카페라떼":
                price = 3500;
                break;
            case "아메리카노" :
                price = 2000;
                break;
            default:
                System.out.println("메뉴에 없습니다!");
        }

        if(price != 0)
            System.out.print(order + "는 " + price + "원입니다");

        scanner.close();
    }
}

// 출력
// (메뉴 : 에스프레소, 카푸치노, 카페라떼, 아메리카노)
// 무슨 커피 드릴까요? 카푸치노
// 카푸치노는 3500원입니다
```