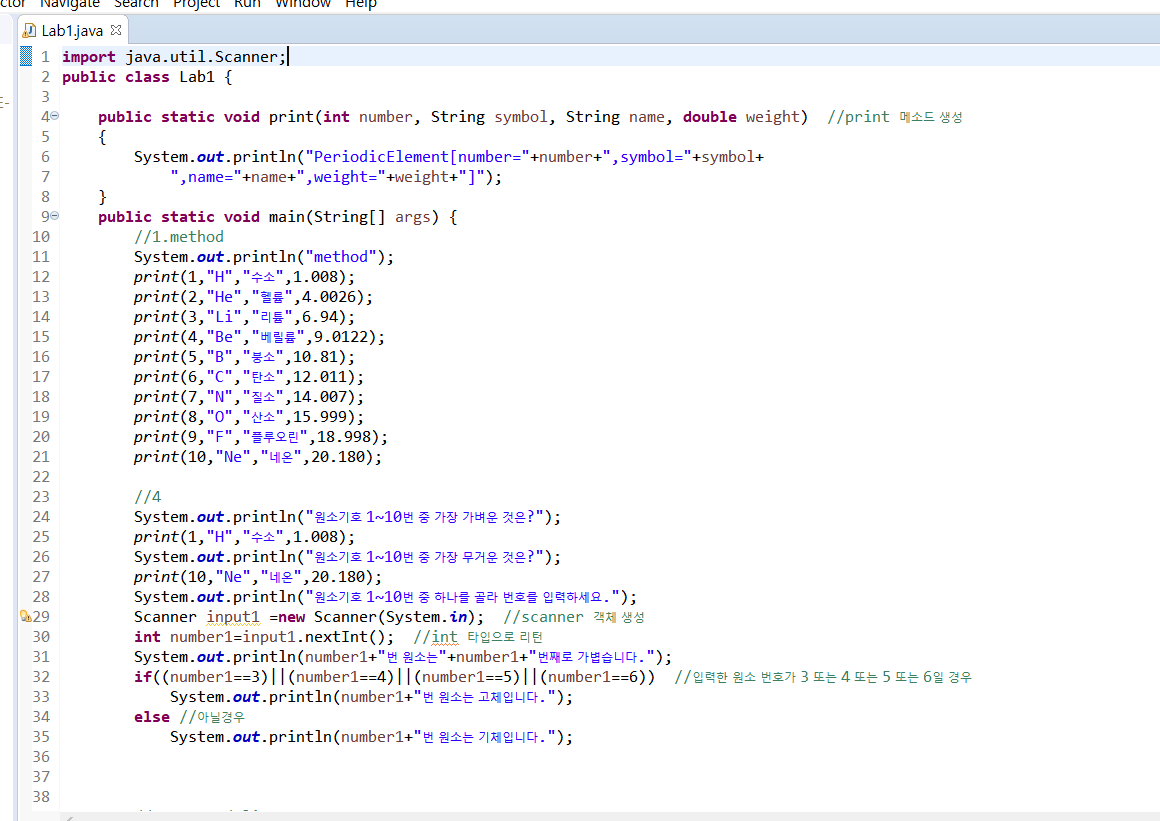
Lab 1&DataType

컴퓨터 공학과

32191826 박주희

**<Lab1>**

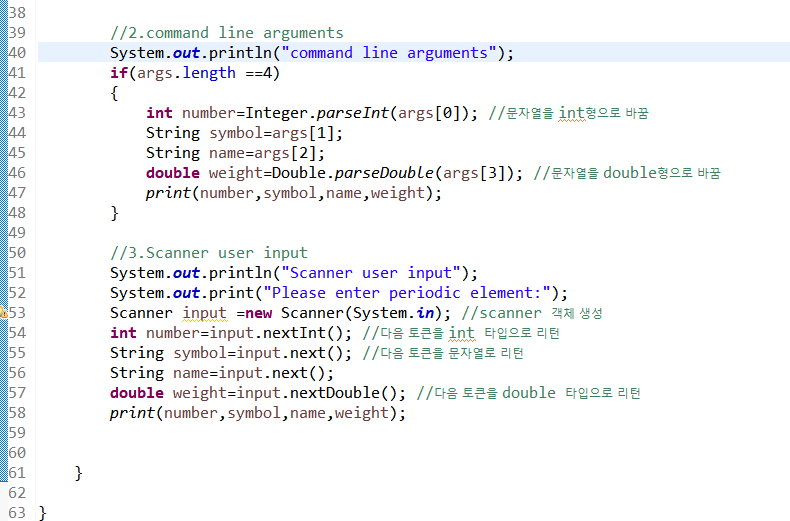


#4~#8

print 메소드를 생성한다. print 메소드는 number, symbol, name, weight를 입력했을 때 PeriodicElement [number=” + number + “, symbol=” + symbol + “, name=” + name + “, weight=” + weight + “] 형태로 나오도록 하는 메소드이다.

#29~#30

scanner 객체는 System.in 객체를 이용하여 키보드로부터 바이트 정보들을 입력받고 원하는 타입(int)으로 변환하여 리턴한다.

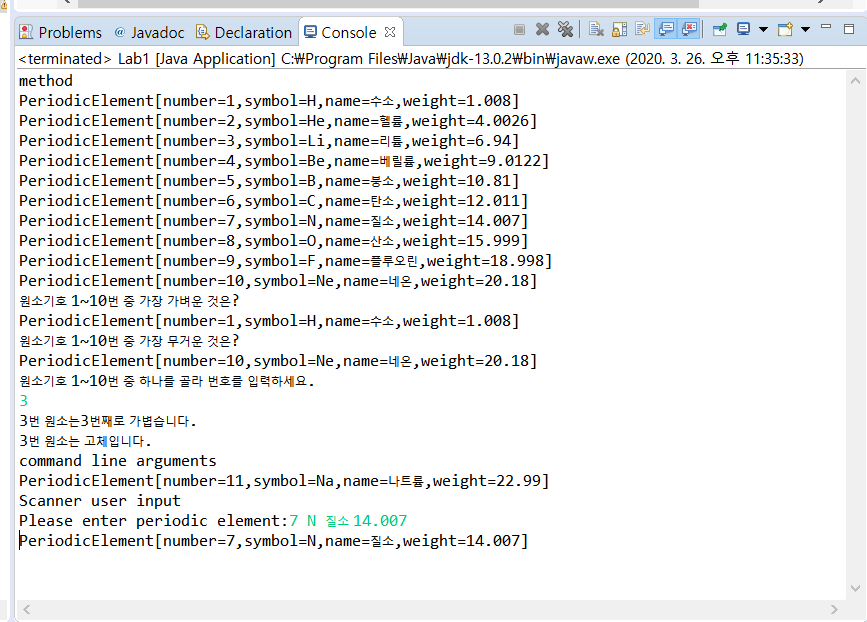


#43~#48

args 문자열 배열의 원소 0번째, 1번째, 2번째, 3번째를 각각 number, symbol, name, weight에 저장한다. number과 weight는 문자열 형태가 아니므로 각각 int형, double형으로 변환하여 저장한다.

#53

scanner 객체는 System.in 객체를 이용하여 키보드로부터 바이트 정보들을 입력받고 원하는 타입으로 변환하여 리턴한다.



**<DataTypeTest>**

1. boolelan

true 또는 false

1. char

char형은 영어, 한자, 한글 모두 가능

1. String

\udddd (dddd is 4-digit number)는 유니코드를 표현

1. Numeric Type: byte, short, int, long, float, double



1. Type Conversion

자동 타입 변환: 타입이 일치하지 않을 때, 컴파일러는 오류 대신 작은 타입을 큰 타입으로 자동 변환한다.

ex) System.***out***.println("(10.0/4)=" + (10.0/4));

//실수 연산을 하기 위해 4가 4.0으로 자동 변환

강제 타입 변환: 개발자가 강제로 타입 변환을 지시하는 경우 (데이터 손실이 발생)

ex) System.***out***.println((**int**)2.9 + (**int**)1.8); // 2 + 1 => 3

integer🡪 string

ex) **int** i = -10; // -10

String s1 = Integer.*toString*(i); // “-10”

String s2 = String.*valueOf*(i); // “-10”

String s4 = String.*format*("%d", i); // “-10”

string 🡪 integer

ex) String x = "-123"; // “-123”

**int** y = Integer.*parseInt*(x); // -123 (integer로 변환이 가능한 경우만 변환 가능 그 외엔 run-time exception error)

**int** z = Integer.*valueOf*(x); // -123 (integer로 변환이 가능한 경우만 변환 가능 그 외엔 run-time exception error)

1. Enum(열거형) - 서로 연관된 상수들의 집합

ex) enum Week {월, 화, 수, 목, 금, 토, 일}

for( Week day:Week.values())

System.out.print(day+”요일”);

🡪 월요일 화요일 수요일 목요일 금요일 토요일 일요일

코드가 단순해지고 가독성이 좋다.