

1. 문자열 (String)

자바가 기본으로 제공하는 클래스 중 하나이다. (자료형과 좀 다르게 여김)

데이터 타입의 변수는 큰따옴표(" ")로 나타냄

문자열을 비교할때는 비교 연산자가 아닌 메소드를 이용한다.

equals()

양 쪽에 있는 내용을 비교한 값을 **boolean type**으로 반환한다

```
public class EqualsTest{
    public static void main(String[] args){
        String a = "JAVA"
        String b = "JAVA"
        String c = new String("JAVA");
        String d = new String("JAVA");

        System.out.println( a == b ); // true   주소값을 비교
        System.out.println( b == c ); // false   새로운 주소를 부여하여 주소를 비교해서 false가 나옴
        System.out.println( c == d ); // false
        System.out.println( a.equals(b) ); // true   내용 자체를 비교(데이터 값) 비교
        System.out.println( b.equals(c) ); // true
        System.out.println( c.equals(d) ); // true   } }
```

equals과 ==의 차이점

1. 형태의 차이

equals ()는 메소드 입니다. 객체끼리 내용을 비교할 수 있도록 합니다.

==은 비교를 위한 연산자입니다.

2. 주소값 비교와 내용 비교

equals 메소드는 비교하고자 하는 대상의 내용 자체를 비교하지만,

== 연산자는 비교하고자 하는 대상의 주소값을 비교합니다.

equalsIgnoreCase()

대소문자를 무시하고 비교해주는 함수이다.

```
public class Hz {
    public static void main(String[] args) {
        String hz1 = "Homzzang.com";
        String hz2 = "homzzang.com";
        String hz3 = "홈짱닷컴";

        System.out.println(hz1.equalsIgnoreCase(hz2)); // true
        System.out.println(hz1.equalsIgnoreCase(hz3)); // false   } }
```

08.10 노트정리

compareTo()

두개의 값을 비교하여 **int 값으로 반환해주는 함수**이다.

숫자의 비교 같은 경우는 단순히 **크다(1), 같다(0), 작다(-1)** 의 관한 결과값으로 반환
(Byte, Double, Integer, Float, Long 또는 Short 등을 비교할 수 있다)

```
public class CompareToTest{
    public static void main(String[] args){
        Integer x = 3;
        Integer y = 4;
        Double z = 1.0;

        System.out.println( x.compareTo(y) ); // -1
        System.out.println( x.compareTo(3) ); // 0
        System.out.println( x.compareTo(2) ); // 1
        System.out.println( z.compareTo(2.7) ); // -1 } }
```

문자열의 비교 같은 경우는 **같다(0), 그 외 양수/음수값** 같이 참 재미난 결과값으로 반환
(비교가 불가능한 지점의 각 문자열의 아스키값을 기준으로 비교를 해준다.)

```
public class CompareToTest{
    public static void main(String[] args){
        String str = "abcd";

        // 1) 비교대상에 문자열이 포함되어있을 경우
        System.out.println( str.compareTo("abcd") ); // 0 (같은 경우는 숫자나 문자나 0을 리턴)
        System.out.println( str.compareTo("ab") ); // 2
        System.out.println( str.compareTo("a") ); // 3
        System.out.println( str.compareTo("c") ); // -2
        System.out.println( "".compareTo(str) ); // -4

        // 2) 비교대상과 전혀 다른 문자열인 경우
        System.out.println( str.compareTo("zefd") ); // -25
        System.out.println( str.compareTo("ABCD") ); // 32 } }
```

compareToIgnoreCase()

대소문자를 무시하고 비교해주는 함수이다.

```
public class CompareToTest{
    public static void main(String[] args){
        String str = "abcd";
        System.out.println( str.compareToIgnoreCase("ABCD") ); // 0
        System.out.println( str.compareToIgnoreCase("AbcD") ); // 0 } }
```

2. 문자열 메소드

문자열 길이 출력 [length()]

```
String str1 = "Hi,";
String str2 = " Java";
String str3, str4;

System.out.println(str2.length()); ->5
```

문자열 뒤에 문자열 붙이기 (비교 연산자 "+"와 같음) [concat()]

```
String str1 = "Hi,";
String str2 = " Java";
String str3, str4;

str3 = str1.concat(str2)
System.out.println(str3); ->Hi, Java
```

index가 지정한 문자 변환 [charAt(index -> 인덱스 번호가 들어감)]

```
String str1 = "Hi,";
String str2 = " Java";
String str3 = "Hi, Java";
String str4;

char c = str1.charAt(2);
System.out.println(c); ->,
```

모두 대문자 or 소문자로 변환 [toUpperCase():대문자, toLowerCase():소문자]

```
String str1 = "Hi,";
String str2 = " Java";
String str3 = "Hi, Java";
String str4;

str4 = str3.toLowerCase(); //소문자
System.out.println(str4); ->hi, java
str4 = str3.toUpperCase(); //대문자
System.out.println(str4); ->HI, JAVA
```

08.10 노트정리

문자열의 일부 변환 [substring(index시작, index끝)]

```
String str1 = "Hi,";
String str2 = " Java";
String str3 = "Hi, Java";
String str4;

str4 = str3.substring(0, 5); //index 0부터 5전 4까지 출력
System.out.println(str4); ->Hi, J
```

앞뒤 공백 제거 후 반환 [trim()]

```
String str1 = "Hi,";
String str2 = " Java";
String str3 = "Hi, Java";
String str4;

str4 = str2.trim();
System.out.println(str4); ->Java
```

문자열 포함 유무 조사 [contains()]

```
String str1 = "Hi,";
String str2 = " Java";
String str3 = "Hi, Java";
String str4 = "Java";

System.out.println(str3.contains(str1)); ->true
```

시작하는 문자열, 끝나는 문자열이 맞는지 조사

[startWith():첫 문자, endWith():끝 문자]

```
String str1 = "Hi,";
String str2 = " Java";
String str3 = "Hi, Java";
String str4 = "Java";

System.out.println(str3.startsWith("H")); ->true
System.out.println(str3.endsWith("K")); ->false
```

문자열의 길이가 0이면 true 반환 [isEmpty()]

```
String str1 = "Hi,";
String str2 = " Java";
String str3 = "Hi, Java";
String str4 = "Java";

System.out.println(str4.isEmpty()); ->false
str4 = "";
System.out.println(str4.isEmpty()); ->true
```

3. 배열

타입이 같은 데이터를 다룰 때 모든 데이터에 변수를 지정한다면
개수가 너무 많아지는 점을 배열을 통해 동일한 데이터 타입으로 집합을
쉽게 처리할 수 있음

각각 변수로 하기 싫어서 변수로 애용함

ex) 과목 5개 점수 -> 변수 5개 or 하나의 참조 변수(배열)

배열의 선언과 생성 [데이터타입[] 변수or배열이름]

고정된 개수의 동일한 데이터의 타입으로 구성됨

[]는 변수 뒤, 데이터 타입 뒤에 쓸 수 있음

```
int[] str1 = new int[5]; //배열안에 입력을 받고 싶으면 new int + [ ]안에 입력받을 개수 쓰기
int str2[] = {10,100,90,50,40}; //변수를 적을 때는 { }안에 적기
```

//배열 요소에 접근하는 방법

```
int a = str2[2] //방법1 배열의 이름[ ]안에 인덱스 번호(번호는 0부터 센다)를 쓴다.
System.out.println(a);
```

```
for(int i = 0; i<str1.length; i++) { //      은 배열의 길이를 의미함
    str1[i] = sc.nextInt(); //방법2(입력을 받아야 하는 배열에 접근 방법) 반복문을 사용한다.
    System.out.print(str1[i] + " ");
}
```

3. 다차원 배열

1차원적 배열

조건식이 true일 때만 실행문 수행할 때 사용한다.

```
int[] str1 = new int[5];
int str2[] = {10,100,90,50,40}; //1차원 배열
```

2차원 배열

조건식이 true일 때만 실행문 수행할 때 사용한다.

```
int[][] str3 = new int [3] [5]; // 3은 행, 1차원 배열을 의미/ 5는 열, 2차원 배열을 의미함
//표 참고 ↑
```

new int [3] [5] 열 →

행 ↓

0,0	0,1	0,2	0,3	0,4
1,0	1,1	1,2	1,3	1,4
2,0	2,1	2,2	2,3	2,4

```
int[][] str4 = { {100,100,60}, {40,20,100}, {20,30,50} };
```

```
int[][] str5 = new int [3] [3];
```

```
for(int i=0; i<str3.length; i++) { //2차원 배열에 접근 방법 for문 중첩문을 써야함
    for(int j=0; j < str3 [i].length; j++) {
        str3[i][j] = sc.nextInt();
        System.out.println(str3[i][j]);
    }
}
```

4주차 과제

첫 번째 과제 (10039)

32089884	suim7171	 10039	맞았습니다!!	17572 KB	204 ms	Java 11 / 수정	521 B	7초 전
----------	----------	---	---------	----------	--------	--------------	-------	------

```
import java.util.Scanner;

public class bj10039 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int sum = 0;
        int[] str1 = new int[5];
        for(int a= 0; a<str1.length; a++) {
            str1[a] = sc.nextInt();
            if(str1[a]< 40) {
                sum +=40;
            }
            else {
                sum += str1[a];
            }
        }
        System.out.println(sum/5);
    }
}
```