

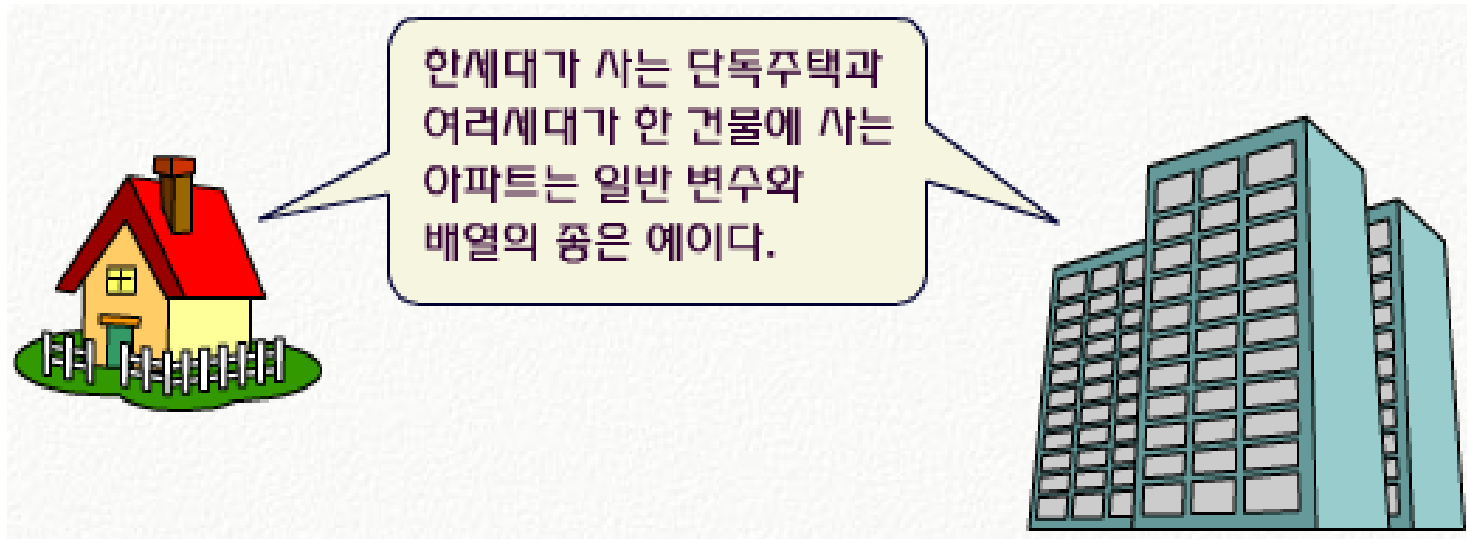
JAVA 프로그램 실습

배열

경북대학교
소프트웨어융합과
배 희 호 교수

Array

- 배열은 동일한 자료 유형이 여러 개 필요한 경우에 이용할 수 있는 자료 구조
 - 일반 변수가 단독주택이라면 배열은 아파트
- 배열은 연속적인 항목들이 동일한 크기로 Memory에 순서적으로 저장되는 구조로 그 사용이 간편



Array

- 하나의 학급에 있는 학생 40명의 평균(실수) 점수를 저장해야 하는 경우
 - 배열이 없을 때는 `double score1, score2, score3` 이렇게 **40개의 변수가** 필요
- 여러 개의 Data를 묶을 수 있는 자료형이 있는 경우
`double score[] = new double[40]` 이렇게 선언해서 `score`라는 하나의 이름으로 관리
- 배열을 선언하면 하나의 이름으로 관리하고 각 공간에 Index를 부여하기 때문에 여러 개의 Data를 반복문을 이용해서 접근할 수 있어서 많은 양의 Data를 적은 양의 Code로 처리할 수 있게 됨

Array

- 배열의 선언

- 자료형[] 배열명; // 추천

- 자료형 배열명[]; // C언어 style

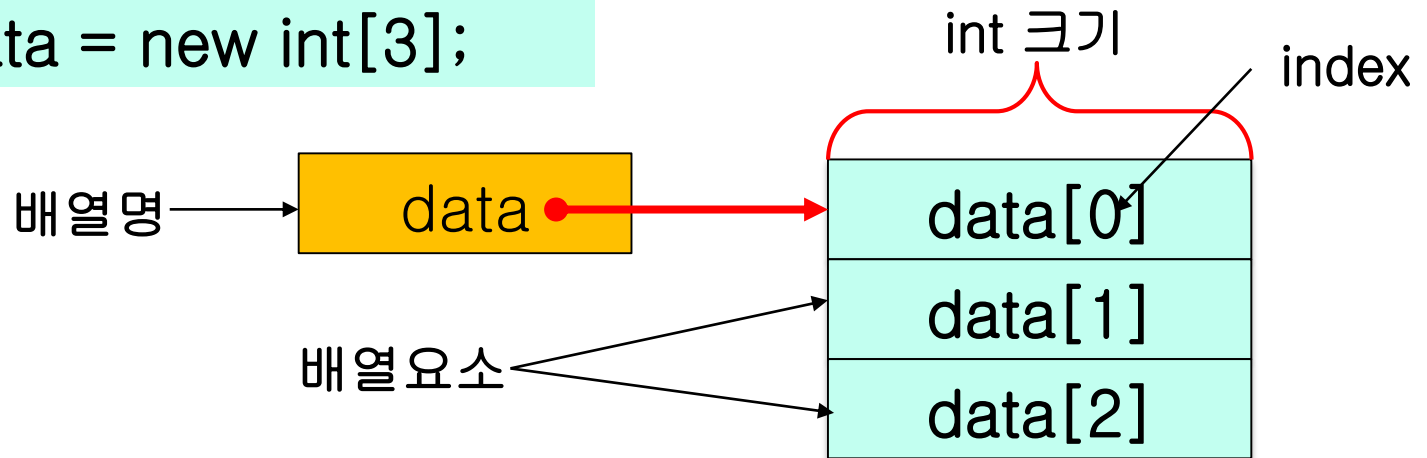
Array

- 구성 요소
 - 배열 요소의 자료형
 - 배열 요소의 자료형(컬렉션의 자료형이 아니고 컬렉션을 구성하는 데이터들의 자료형)
 - 예) int, double 등
 - 배열의 크기
 - 배열 요소의 개수
 - 예) [], [7] 등
 - 크기의 개수에 따라 1차원 배열과 2차원 배열 이상의 다차원 배열로 구분
 - 배열명
 - Memory 공간을 사용하므로 재사용을 위해서는 이름이 있어야 재사용 가능

Array

- 같은 종류(Data Type이 동일)의 Data를 하나의 이름으로 표시되는 Data의 묶음
 - 같은 유형의 Data를 여러 개 저장
 - 여러 개의 연속적인 Memory
 - 순서적인 이름을 가짐

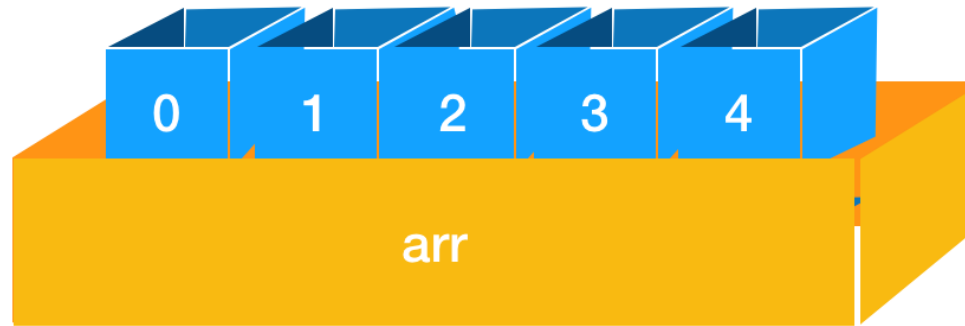
```
int[] data = new int[3];
```



- 참조 변수(reference variable)를 통해 배열을 사용
 - 참조 변수는 변수의 일종
 - 참조 변수는 배열을 가리킴

Array

■ 1차원 배열과 2차원 배열



```
int[] arr = new int[5];
```

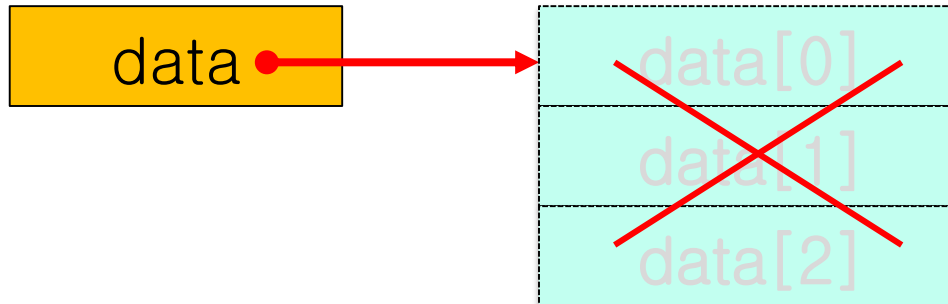


```
double[][] doubleArr = new double[2][5];
```

Array

- 참조변수 선언만으로는 배열이 만들어지지 않음

```
int[] data;    // int 배열을 가리키기 위한 참조변수 선언
```



```
int[] data;  
data = new int[3];    // 배열을 만들고 참조변수에 대입
```

```
int[] data = new int[3];    // 한 문장으로 작성
```


Array

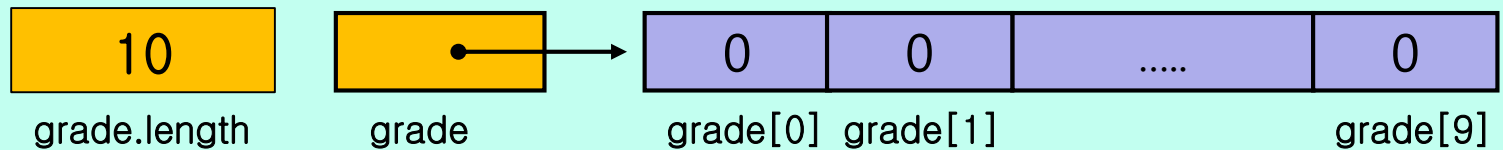
- 배열에 Memory 할당(Data를 저장할 실제 공간 생성) – Data를 저장할 수 있는 공간을 할당
 - 배열명 = new 자료형[개수]
 - 모두 기본 값으로 초기화
 - Heap Memory는 할당되는 순간 초기 값이 없으면 기본 값으로 초기화
 - 자료형 배열명[] = {Data 나열}
 - 나열된 Data를 가지고 생성하는 것으로 배열을 처음 생성할 때만 가능한 방식
 - 자료형 배열명[] = new 자료형[] {Data 나열}
 - 나열된 Data를 가지고 생성
 - 배열명 = null
 - 현재 참조할 Data가 없다면 null을 대입하는 것도 가능

Array 객체 생성 방법

■ 배열 생성하는 방법

■ 명시적인 방법('new' 연산자를 사용)

```
int[ ] grade;  
grade = new int[10];
```



■ 배열의 선언과 함께 배열 객체를 동시에 생성

```
int[ ] grade = new int[10];
```

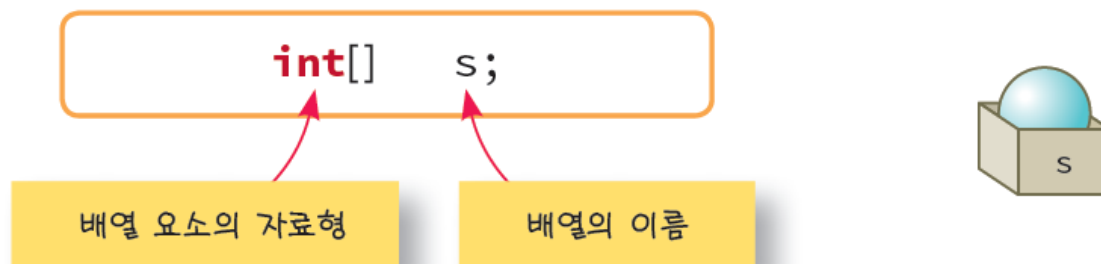
■ 초기 값을 이용하는 방법

```
int[ ] grade = new int[] {45, 89, 100, 40, 80};
```

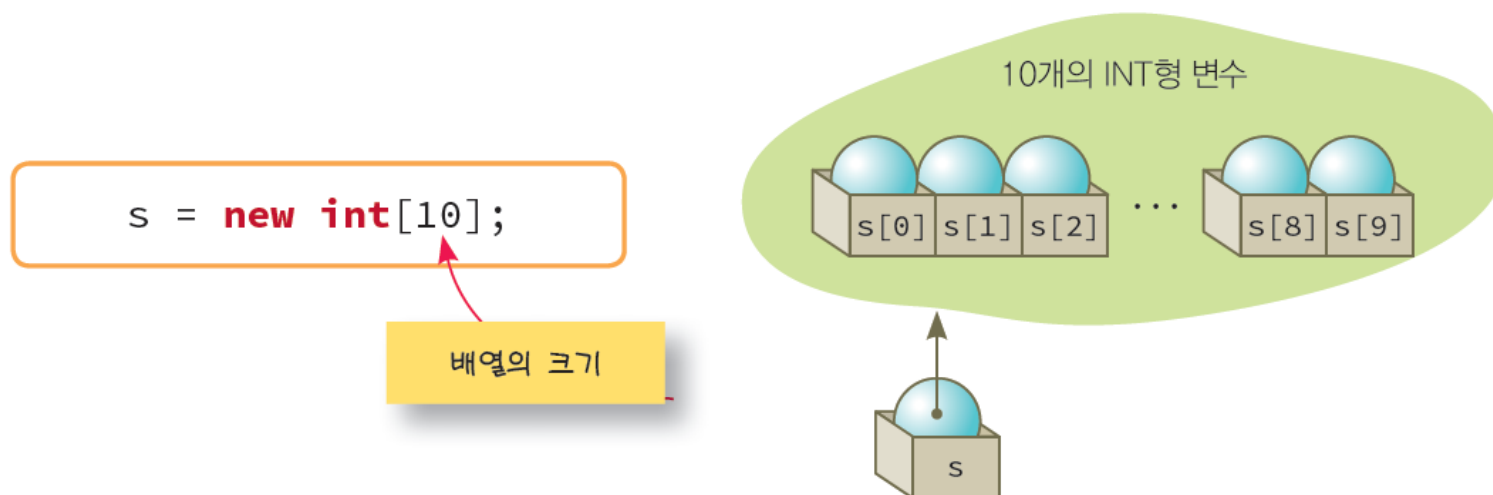
```
int[ ] grade = {45, 89, 100, 40, 80};
```

Array 객체 생성 방법

■ 먼저 배열 참조 변수부터 선언



■ 배열을 new 연산자를 사용하여 생성



Array 객체 생성 방법

- 5개의 정수형 원소 {2, 5, 3, 9, 6}을 갖는 배열 test를 정의 하려보아라

[사용 방법 1]

```
int[] test = null;    // 선언
test = new int[5];    // 정의
test[0] = 2;          // 초기화
test[1] = 5;
test[2] = 3;
test[3] = 9;
test[4] = 6;
```

[사용 방법 2]

```
int[] test = new int[] {2, 5, 3, 9, 6};
```

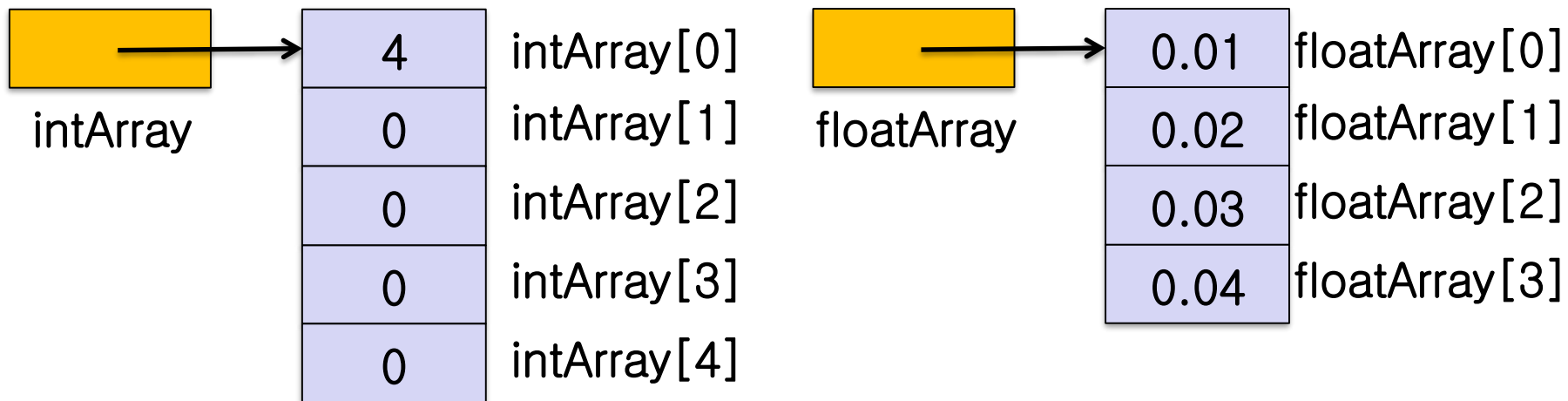
[사용 방법 3]

```
int[] test = {2, 5, 3, 9, 6};
```

Array 객체 생성 방법

- 배열을 초기화하면서 생성한 결과

```
int intArray[] = {4, 3, 2, 1, 0};  
float floatArray[] = {0.01f, 0.02f, 0.03f, 0.04f};
```



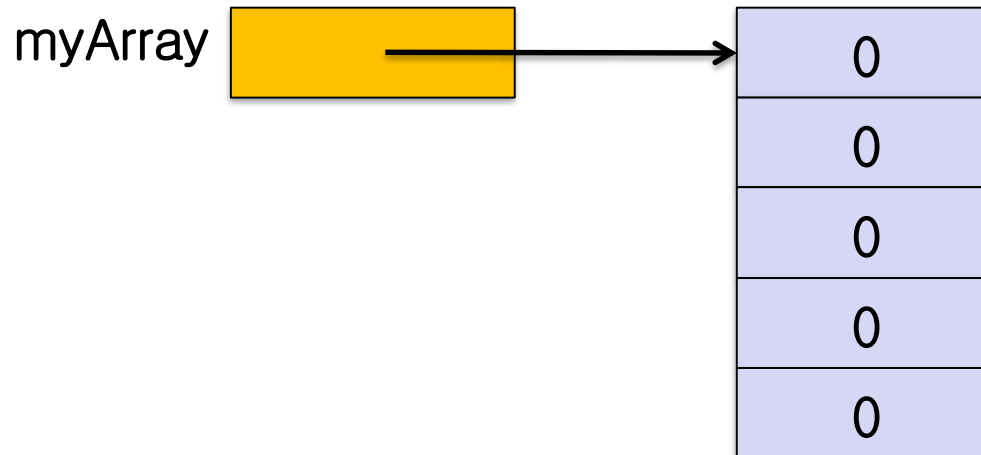
- 배열에 저장되는 값은 모두 같은 Data 유형이어야 함
- 배열의 방 번호를 Index라고 함
- 배열의 Index는 항상 양의 int Type 임

Array 객체 생성 방법

■ 배열의 기본 값(default value)

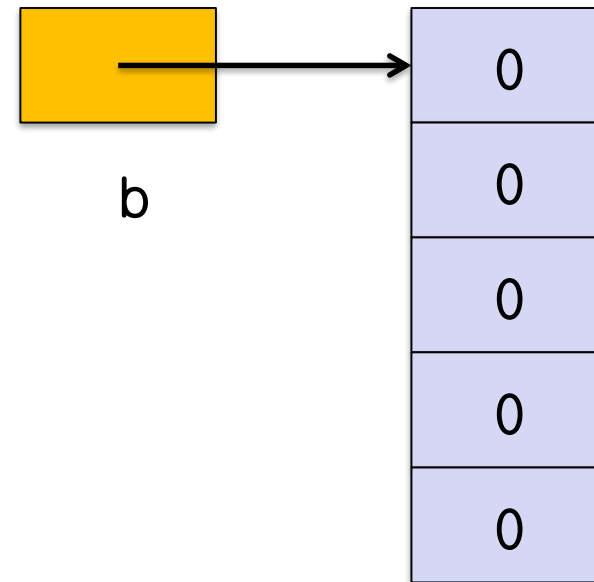
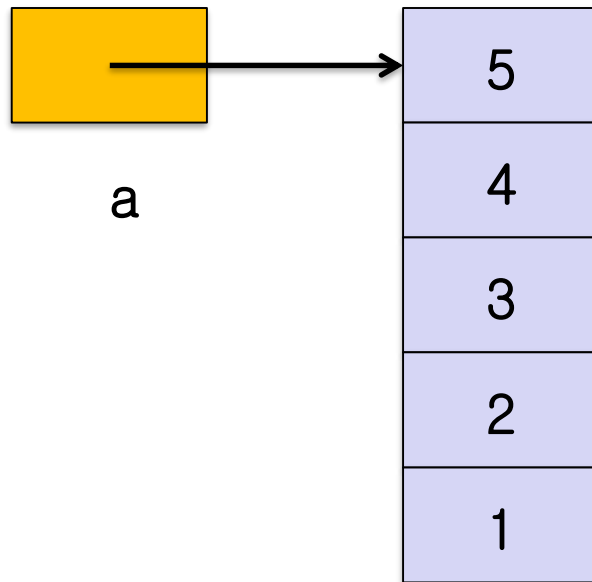
배열은 기본 값 (default value)이 들어 있는 상태로 만들어짐

```
int[] myArray = new int[5];           // 0이 초기화
double[] myArray = new double[5];     // 0.00이 초기화
boolean[] myArray = new boolean[5];   // false가 초기화
char[] myArray = new char[5];         // 0이 초기화
```



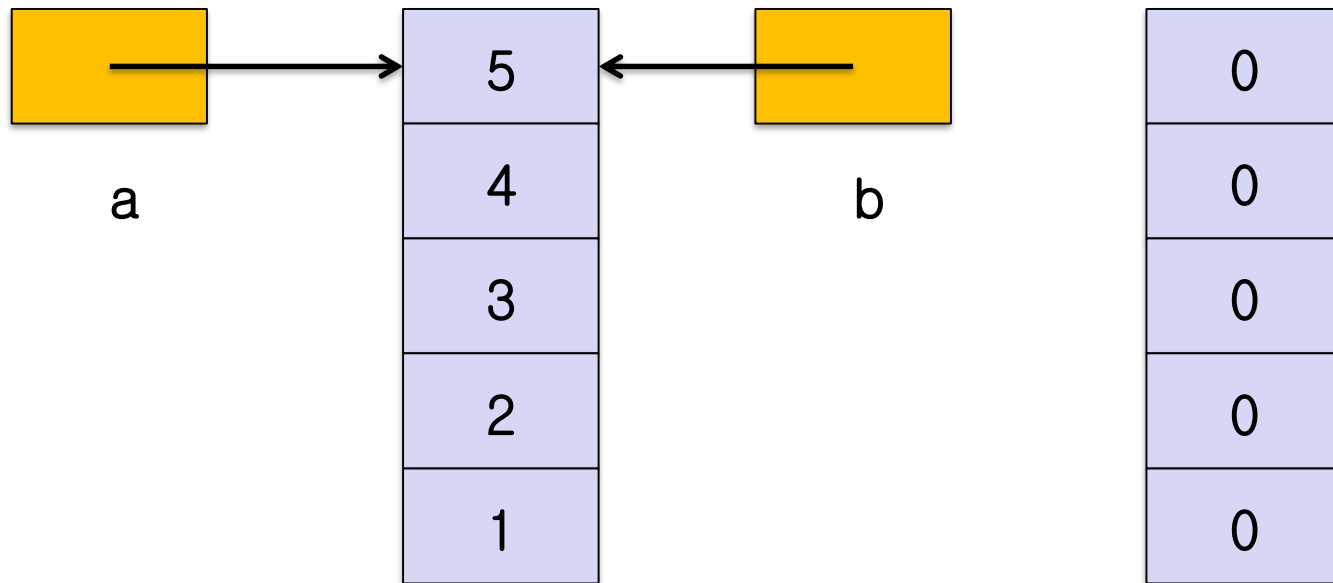
Array 객체 생성 방법

```
int[] a = {5, 4, 3, 2, 1};  
int[] b = new int[5];    // default value 0이 들어 있음
```



Array 객체 생성 방법

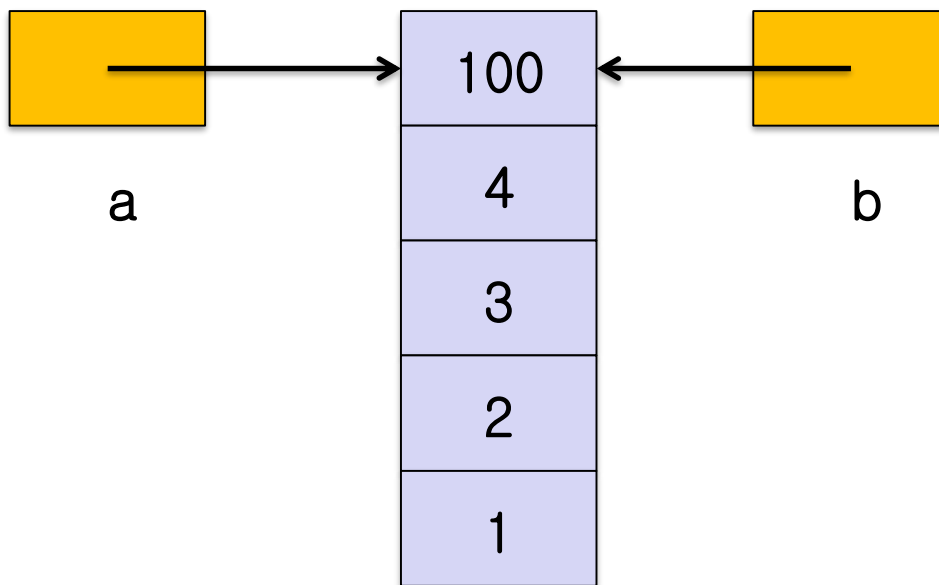
```
int[] a = {5, 4, 3, 2, 1};  
int[] b = new int[5];    // default value 0이 들어 있음  
b = a;
```



쓰레기 수거
Garbage collection

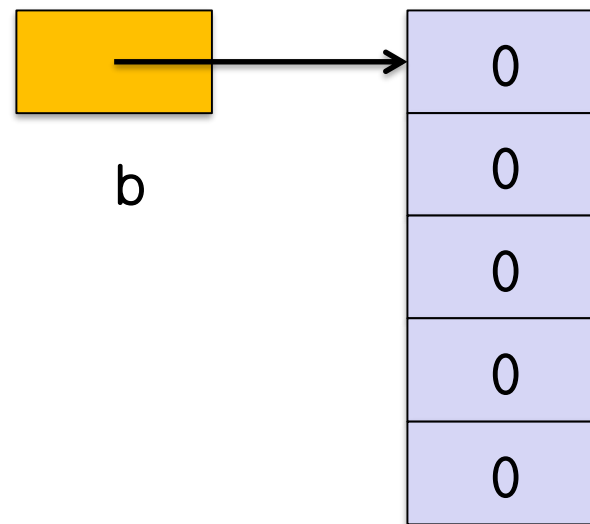
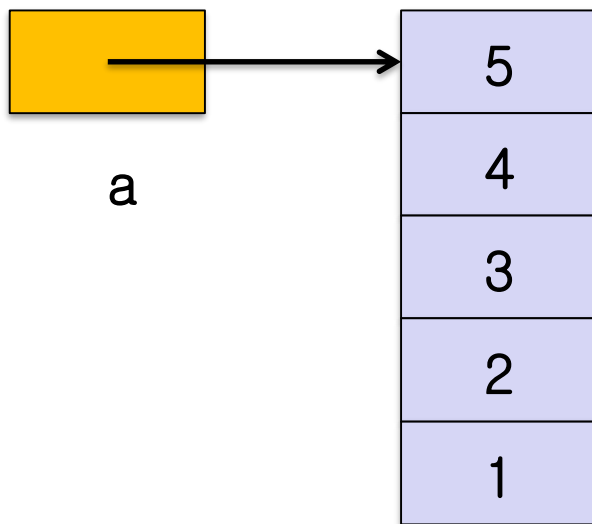
Array 객체 생성 방법

```
int[] a = {5, 4, 3, 2, 1};  
int[] b = new int[5];    // default value 0이 들어 있음  
b = a;  
b[0] = 100;
```



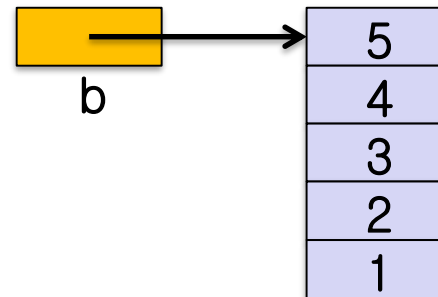
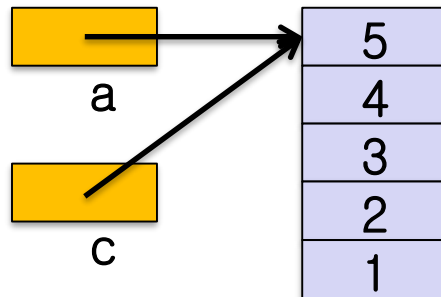
참조변수에 대한 == 연산자

```
int[] a = {5, 4, 3, 2, 1};  
int[] b = {5, 4, 3, 2, 1};  
if (a == b) {  
    System.out.println("true");  
} else {  
    System.out.println("false");  
}
```



참조변수에 대한 == 연산자

```
int[] a = {5, 4, 3, 2, 1};  
int[] b = {5, 4, 3, 2, 1};  
int[] c = a;  
if (a == b) {  
    System.out.println("true");  
} else {  
    System.out.println("false");  
}  
if (a == c) {  
    System.out.println("true");  
} else {  
    System.out.println("false");  
}
```



배열 초기화

- 참조변수 선언, 배열 생성, 배열 초기화를 한번에 할 수도 있음

```
char[] letters = {'a', 'b', 'c'}; // 이 한 문장은  
// 아래 다섯 문장과 같음
```

```
char[] letters ;  
letters = new char[3];  
letters[0] = 'a';  
letters[1] = 'b';  
letters[2] = 'c';
```

- 배열의 크기는 자동으로 정해짐

배열의 크기

- 배열이 구성되고 나면 배열의 크기는 고정됨
 - 배열이 구성된 후 배열의 길이를 변경할 수 없음
- 참조 변수 x가 배열을 가리킬 때 **x.length**는 배열에 할당된 Memory Cell의 개수를 알려줌
(length에 괄호가 붙지 않음에 유의!)

```
int[] test = new int[10];
```

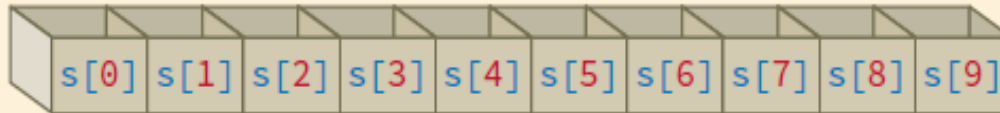
```
for (int i = 0; i < test.length; i++) {  
    System.out.println(myArray[i]);  
    // i번 방의 값을 읽어 출력  
}
```

배열 Index

- 다음과 같은 배열을 가정하자

```
int[] s = new int[10];
```

배열은 하나의 이름을 공유한다.

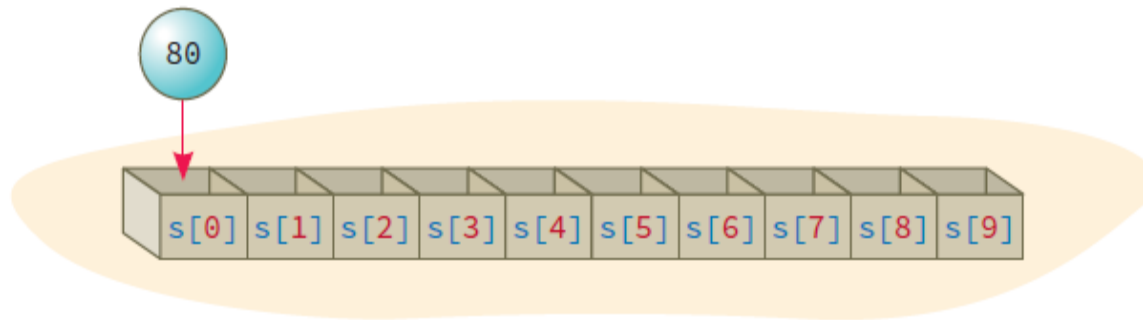


- 배열 요소에는 번호가 붙어 있는데 이것을 Index라고 부름
 - 첫 번째 요소의 번호는 0이고, 마지막 요소의 번호는 9가 됨

배열 Index

- 배열 Index를 통한 요소의 접근
 - 배열은 변수들이 모인 것이니, 배열을 이루고 있는 배열 요소는 하나의 변수로 생각하면 됨
 - 배열의 첫 번째 요소에 80을 저장하려면 다음과 같이 함

```
s[0] = 80;
```



배열 Index

■ 배열 Index

- Index는 0부터 시작
- 마지막 Index는 (배열 크기 - 1)
- Index는 정수 type만 가능

```
int intArray = new int[5];
```

```
int n = intArray[-2]; // 실행 오류. -2는 인덱스로 적합하지 않음
```

```
int m = intArray[5]; // 실행 오류. 5는 인덱스의 범위(0~4)를 넘었음
```

■ 배열의 크기

- 배열의 크기는 배열 참조 변수를 선언할 때 결정되지 않음
- 배열의 크기는 배열 생성 시에 결정되며, 나중에 바꿀 수 없음
- 배열의 크기는 배열의 length라는 Field에 저장

```
int size = intArray.length;
```


배열 접근 방법

- 배열 원소 접근
 - 반드시 배열 생성 후 접근

```
int intArray [];  
intArray[4] = 8;    // 오류, intArray가 초기화되어 있지 않음
```

- 배열 변수명과 [] 사이에 원소의 Index를 이용하여 접근
 - 배열의 Index는 0부터 시작
 - 배열의 마지막 항목의 Index는 (배열 크기 - 1)

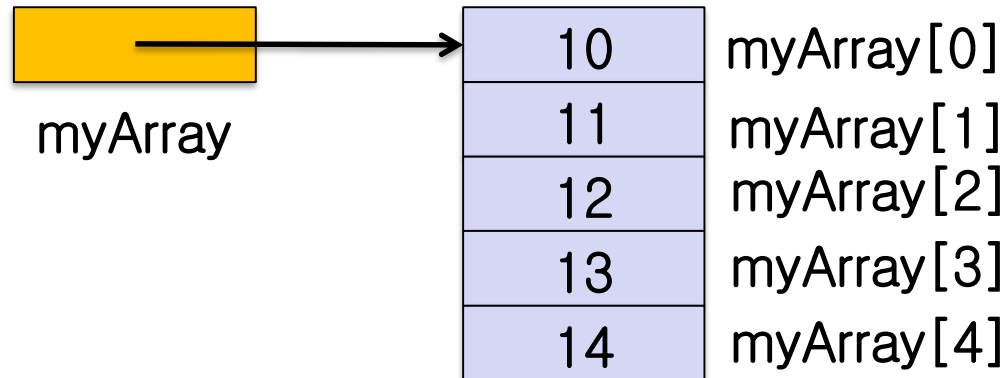
```
int[] intArray;  
intArray = new int[10];
```

```
intArray[3]=6;           // 배열에 값을 저장  
int n = intArray[3];     // 배열로부터 값을 읽음
```

배열 접근 방법

■ 배열에서 값 읽기

```
for (int i = 0; i < 5; i++){  
    System.out.println(myArray[i]); // i번 방의 값을 읽어 출력  
}
```



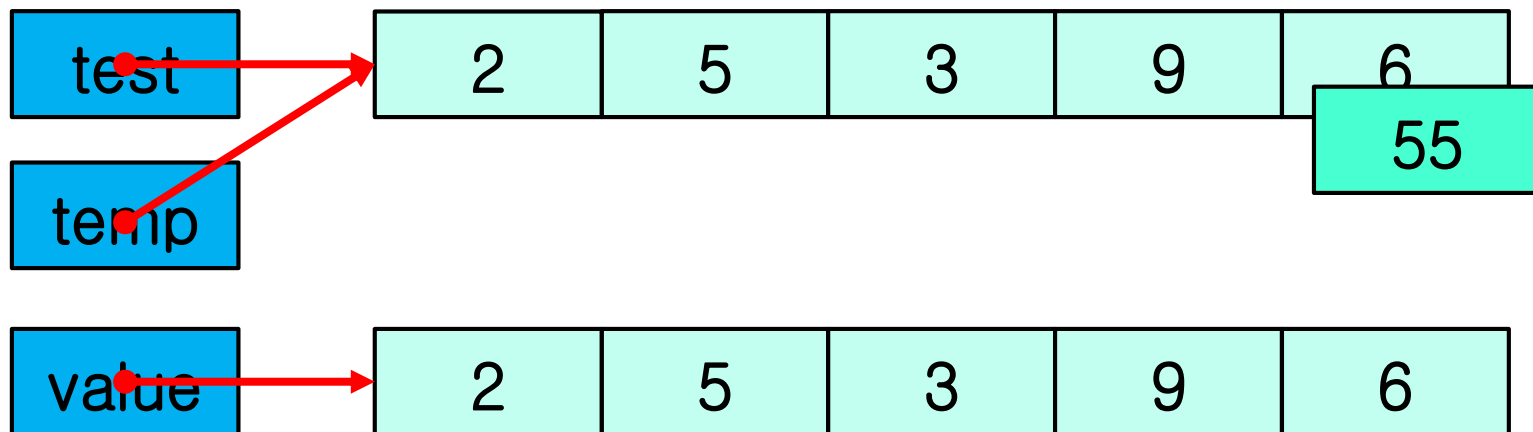
myArray가 배열을 가리키는 참조변수일 때
myArray[i]는 배열의 i번 방을 의미

배열의 복사

- 얇은 카피(shallow copy)
 - 배열의 주소 값을 넘김 (reference assignment)
 - 예) 메소드의 매개 변수 사용
- 깊은 카피(deep copy)
 - 배열의 값 자체를 넘김 (value assignment)

배열의 복사

```
int[] test = {2, 5, 3, 9, 6};  
int[] temp = test;           //reference assignment  
int[] value = new int[5];  
System.arraycopy(test, 0, value, 0, test.length);  
                                //value assignment  
temp[4] = 55;
```



배열에서 자주 사용되는 Error들

- 배열의 선언을 하지 않았을 경우
 - invalid indirection error message
- 배열의 Index 영역을 벗어난 경우
 - 충돌(crash) 혹은 Memory로부터 접근 가능한 원소가 아니라는 Message 출력
- 배열의 모든 원소를 검침할 만큼 반복문의 Index 크기가 충분하지 못한 경우
 - 배열의 원소가 10개인데 for 문의 Index가 충분하지 못한 경우
- 배열의 초기화를 생략한 경우
 - 대부분의 Compiler가 자동으로 0으로 초기화시키지만 그렇지 않을 수 있음

1차원의 배열에 값 지정과 출력 방법

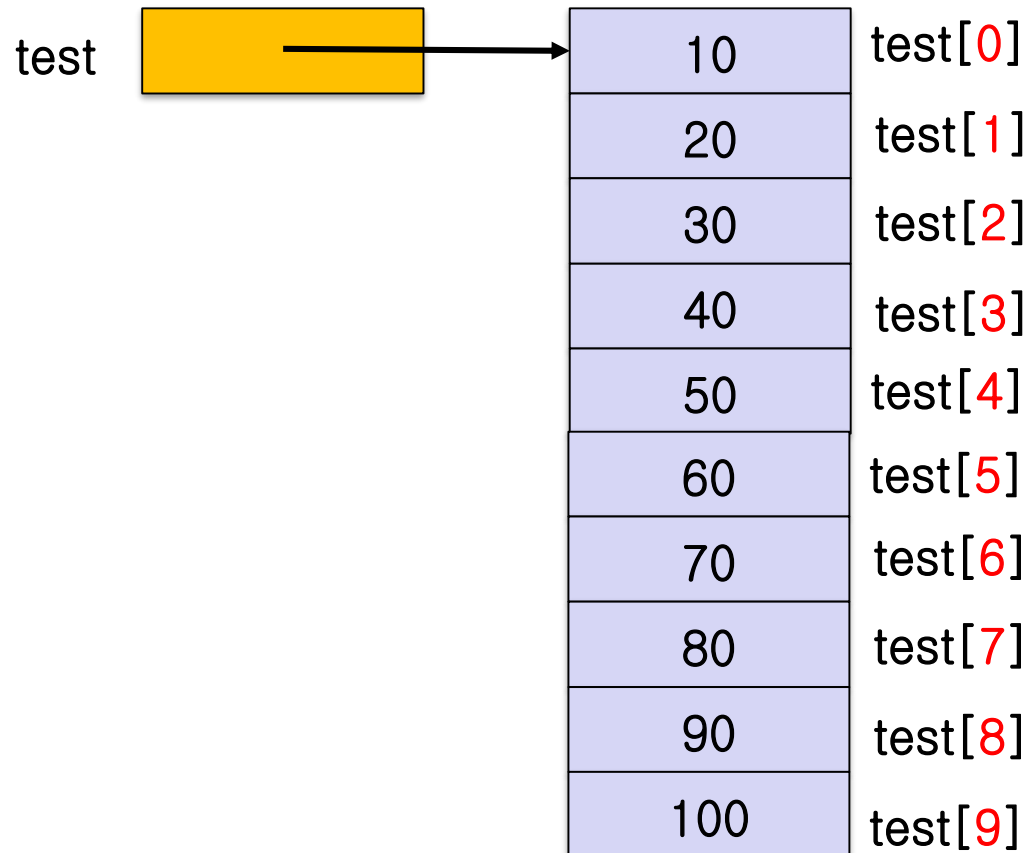
```
public class Array01 {  
    public static void main(String[] args) {  
        int[] score = new int [5]; //5명의 점수를 저장하기 위한 배열 선언  
  
        score[0] = 95; //배열의 원소에 접근하여 점수를 저장  
        score[1] = 70;  
        score[2] = 80;  
        score[3] = 75;  
        score[4] = 100;  
  
        for (int i = 0; i < 5; i++)    //반복문으로 배열을 일괄 처리함  
            System.out.println( (i+1) + " th score[ " + i + " ] = " + score[i]);  
    }  
}
```

1차원 배열의 총합과 평균 구하기

```
public class Array02 {  
    public static void main(String[] args) {  
        int[] score = {95, 70, 80, 75, 100};  
        int total = 0;  
        double ave;  
  
        for (int i = 0; i < 5; i++)    //반복문으로 배열을 일괄 처리함  
            total += score[ i ];    //총합을 구함  
        ave = (double) total / 5.0; //평균을 구함  
  
        System.out.println(" Total = " + total); //총합 출력  
        System.out.println(" Ave  = " + ave);    //평균 출력  
    }  
}
```

1차원 배열 예제 1

- 크기가 10인 정수형 배열을 생성하고 여기에 10부터 100까지의 값으로 배열을 채우는 프로그램을 작성해 보자



1차원 배열 예제 1

■ 선언 시 초기화를 이용

```
int[] test = {10, 20, 30, 40, 50, 60, 70, 80, 90};
```

```
int[] test = new int[] {10, 20, 30, 40, 50, 60,  
                        70, 80, 90};
```

```
int[] test = new int[10];  
for (int i = 0; i < test.length; i++ )  
    test[i] = (i + 1) * 10;
```

1차원 배열 예제 1

```
public static void main(String[] args) {  
    int[] test = new int[10];  
    // int[] test = {10, 20, 30, 40, 50, 60, 70, 80, 90};  
  
    for (int x = 0; x < test.length; x++)  
        test[x] = (x + 1) * 10;  
  
    for (int i = 0; i < test.length; i++)  
        System.out.println("test[" + i + "] = " + test[i]);  
}
```

foreach 문

- foreach(for-each)문은 for, while, do-while 반복문과 같은 배열 탐색 기법
 - 일반적인 for 반복문과 동일하게 for Keyword를 사용
 - 반복문 내에 Counter 변수를 선언하고 Colon(:) 다음 배열 이름을 순서대로 선언
 - 일반적으로 배열이나 Collection 클래스(ArrayList ... 등)를 반복하는 데 사용
 - 장점
 - Index를 생성해 접근하는 단순 for문 보다 수행속도가 빠름
 - Code가 짧고, 가독성도 높음

foreach 문

```
int[] arr = {0, 1, 2, 3, 4};  
for (int i : arr) {  
    System.out.println(arr[i]);  
}
```

■ foreach 문의 한계

- 반복문 내에서 배열이나 List의 값을 변경 하거나 추가할 수 없음

```
for(int i : arr) {  
    arr[i] = 3;    // 오류 발생!!, i가 무엇?  
    arr[i+1] = 4; // 오류 발생!! i가 대체 무엇?  
}
```

- 배열을 역순으로 탐색할 수 없음

1차원 배열 예제 1-1

- 10개의 정수를 입력 받고, 입력한 수를 역순으로 보여주는 프로그램을 작성하여라

입력 : 0 11 2 33 4 55 6 77 8 99 <enter>

출력 : 99 8 77 6 55 4 33 2 11 0

1차원 배열 예제 1-1

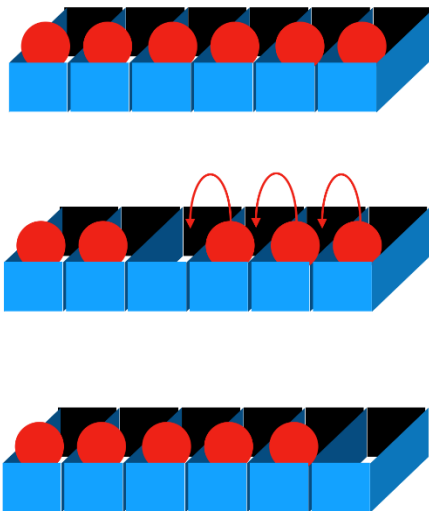
```
Scanner keyboard = new Scanner(System.in);  
int[] data;           // 참조변수 선언  
data = new int[10];   // 배열 구성
```

```
System.out.print(" 10개의 정수형 데이터 입력 : ");  
// 입력된 값들을 data[0], data[1],..., data[9]에 차례로 저장  
for (int i= 0; i < data.length; i++)  
    data[i] = keyboard.nextInt();
```

```
System.out.print( " 역순으로 데이터 출력 : ");  
// data[9], data[8], ..., data[0]에 저장된 값들을 차례로 출력  
for (int i= data.length - 1; i >= 0; i--)  
    System.out.print(data[i] + " ");  
System.out.println();
```

배열의 한계점

- 배열의 개수가 정해지면 수정이 불가능함
- 배열의 중간 원소가 삭제되면 중간이 비어있는 형태가 됨
 - 순차적인 배열을 만들기 위해 배열의 원소를 옮기는 작업을 해야 함



- 배열 중간에 원소를 삽입하고 싶으면 위와는 반대로 원소를 옮기는 작업을 하고 값을 삽입해야 함

도입 예제

- 5명의 JAVA 성적을 저장하고 총점과 평균을 구하는 프로그램을 작성하여라.



도입 예제

```
public static void main(String[] args) {  
    int[] score = new int[5]; //5명의 점수를 저장하기 위한 배열 선언  
    int total = 0;  
    float avg;  
  
    score[0] = 94;  
    score[1] = 70;  
    score[2] = 84;  
    score[3] = 74;  
    score[4] = 100;  
  
    for(int i = 0; i < score.length; i++){  
        total += score[i];  
    }  
    avg = (float) total / score.length;  
}
```

도입 예제

```
for(int i = 0; i < score.length; i++) {  
    System.out.println("score[" + i + "] = " + score[i]);  
}  
System.out.println("total = " + total);  
System.out.println("avg = " + avg);  
}
```

역순 출력

- 임의의 문자열을 입력 받아 역순으로 출력하는 Program을 작성하시오

문자열 입력 : HELLO

출력 문자열 : OLLEH

역순 출력

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    String temp;  
  
    System.out.print("영문 문자열 입력 : ");  
    temp = keyboard.nextLine();  
  
    char[] arr = new char[temp.length()];  
    for (int i = 0; i < arr.length; i++) {  
        arr[i] = temp.charAt(arr.length - 1 - i);  
    }  
  
    System.out.printf("%s", new String(arr));  
}
```

대소문자 변경

- 문자열을 입력 받아서 대문자는 소문자로, 소문자는 대문자로 바뀌어서 출력하는 프로그램을 작성하시오.

대소문자 변경

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    String temp;  
  
    System.out.print("영문 문자열 입력 : ");  
    temp = keyboard.nextLine();  
  
    char[] arr = new char[temp.length()];  
    for (int i = 0; i < temp.length(); i++) {  
        if (temp.charAt(i) >= 'A' && temp.charAt(i) <= 'Z')  
            arr[i] = (char) (temp.charAt(i) + 32);  
        else if (temp.charAt(i) >= 'a' && temp.charAt(i) <= 'z')  
            arr[i] = (char) (temp.charAt(i) - 32);  
        else  
            arr[i] = temp.charAt(i);  
    }  
  
    System.out.printf("%s", new String(arr));  
}
```

학생 몸무게

- 학생 10명의 몸무게를 입력 받아 가장 몸무게가 무거운 학생과 가장 가벼운 학생의 이름과 몸무게를 출력하시오
- 단 몸무게는 소수점 1자리까지 입력 받음
- 몸무게는 30.0 Kg ~ 120.0 Kg

학생 몸무게

```
public static void main(String[] args) {
    Scanner keyboard = new Scanner(System.in);
    String[] name = {"홍길동", "김대한", "한민국", "나이쁨", "이기쁨",
                    "이자바", "나똥똥", "정문제", "김민주", "노재팬"};
    float[] weight = new float[name.length];
    float max = 30.0f;
    String maxname = "";
    float min = 120.0f;
    String minname = "";
    int i = 0;
    while (i < weight.length) {
        System.out.printf("(%d/%d) %s님의 몸무게 입력 : ", i+1,
                           weight.length, name[i]);
        weight[i] = keyboard.nextFloat();
        if (weight[i] >= max && weight[i] <= min)
            i++;
        else
            System.out.println("ERROR");
    }
}
```


학생 몸무게

```
for (i = 0; i < weight.length; i++) {  
    if (max < weight[i]) {  
        max = weight[i];  
        maxname = name[i];  
    }  
}
```

```
for (i = 0; i < weight.length; i++) {  
    if (min > weight[i]) {  
        min = weight[i];  
        minname = name[i];  
    }  
}
```

학생 몸무게

```
for (i = 0; i < weight.length; i++)  
    System.out.printf(" %s님 몸무게 : %.1f\n", name[i], weight[i]);  
System.out.printf(" 가장 가벼운 %s님 몸무게 : %.1f\n",  
                    minname, min);  
System.out.printf(" 가장 무거운 %s님 몸무게 : %.1f\n",  
                    maxname, max);  
}
```

학생 몸무게

- 문제에서 몸무게 올림차순 순으로 출력하시오
 - 내림차순과 올림차순 ?
 - Bubble Sort와 Selection Sort ?

학생 몸무게 정렬(sorting)

- Sorting(정렬)
 - 자료를 순서대로 나열하는 것
- Selection Sort(선택 정렬)
 - n개의 Data중 가장 작은 Data를 찾아서 첫째 원소와 교환
 - 나머지 n-1개 Data중 가장 작은 Data를 찾아서 둘째 원소와 교환
 - 이러한 방식을 1개 Data가 남을 때까지 반복 수행

10	4	6	1	5
----	---	---	---	---

a[0] ~a[4]의 최대값 위치 찾은 후 a[0]와 교환

1	4	6	10	5
---	---	---	----	---

a[1] ~a[4]의 최대값 위치 찾은 후 a[1]와 교환

1	4	6	10	5
---	---	---	----	---

a[2] ~a[4]의 최대값 위치 찾은 후 a[2]와 교환

1	4	5	10	6
---	---	---	----	---

a[3] ~a[4]의 최대값 위치 찾은 후 a[3]와 교환

1	4	5	6	10
---	---	---	---	----

정렬된 결과

학생 몸무게 정렬(sorting)

```
void select_sort(int data[ ], int n) { ... data : 정렬할 배열, n : 배열의 크기
    int i; ... 정렬되지 않은 시작 인덱스
    int min_i; ... 최소값의 인덱스
    int j, temp;

    for (i = 0; i < n-1; i++) {
        /* data[i] ~ data[n-1]의 최소값의 인덱스 찾기 */
        min_i = i; ... data[i]가 최소값이라고 가정
        for (j = i+1; j < n; j++)
            if (data[j] < data[min_i]) ... data[j]가 현재 최소값보다 작으면
                min_i = j; ... 최소값 인덱스 변경
        if (i != min_i) { ... i가 최소값 인덱스가 아니면 두 위치의 자료 교환
            temp = data[i];
            data[i] = data[min_i];
            data[min_i] = temp;
        }
    }
}
```

학생 몸무게 정렬(sorting)

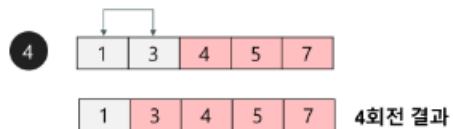
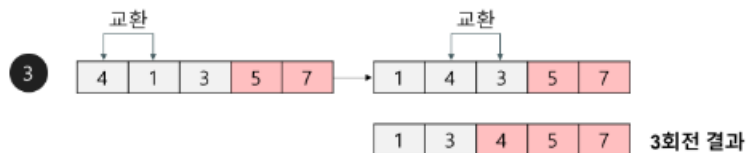
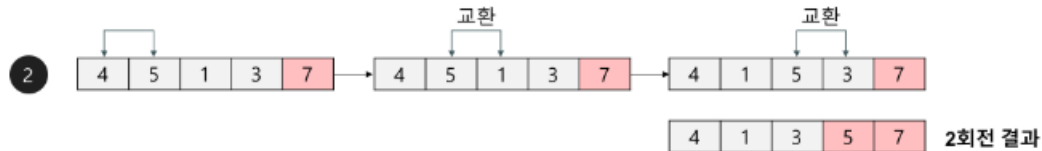
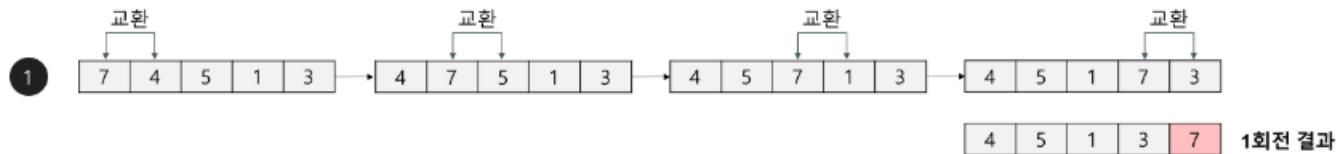
- Bubble Sort(버블 정렬)
 - 서로 인접한 두 원소를 검사하여 정렬하는 Algorithm
 - 인접한 2개의 Record를 비교하여 크기가 순서대로 되어 있지 않으면 서로 교환
 - 선택 정렬과 기본 개념이 유사함
- Bubble Sort Algorithm
 - 버블 정렬은 첫 번째 Data와 두 번째 Data를, 두 번째 Data와 세 번째 Data를, 세 번째와 네 번째를, ... 이런 식으로 (마지막-1)번째 Data와 마지막 Data를 비교하여 교환하면서 Data를 정렬
 - 1회전을 수행하고 나면 가장 큰 Data가 맨 뒤로 이동하므로 2회전에서는 맨 끝에 있는 Data는 정렬에서 제외되고, 2회전을 수행하고 나면 끝에서 두 번째 Data까지는 정렬에서 제외
 - 이렇게 정렬을 1회전 수행할 때마다 정렬에서 제외되는 Data가 하나씩 줄어든다

학생 몸무게 정렬(sorting)

■ Bubble Sort

초기상태

7	4	5	1	3
---	---	---	---	---



오름차순
완성상태

1	3	4	5	7
---	---	---	---	---

학생 몸무게(Sorting)

■ Selection Sort

```
for (i = 0; i < weight.length - 1; i++) {  
    least = i;  
    for (int j = i + 1; j < weight.length; j++) {  
        if (weight[j] < weight[least])  
            least = j;  
        temp = weight[i];  
        weight[i] = weight[j];  
        weight[least] = temp;  
  
        tempname = name[i];  
        name[i] = name[j];  
        name[least] = tempname;  
    }  
}
```


학생 몸무게(Sorting)

■ Bubble Sort

```
for (i = 0; i < weight.length; i++) {  
    for (int j = 0; j < weight.length - i - 1; j++)  
        if (weight[j] > weight[j + 1]) {  
            temp = weight[j];  
            weight[j] = weight[j + 1];  
            weight[j + 1] = temp;  
  
            tempname = name[j];  
            name[j] = name[j + 1];  
            name[j + 1] = tempname;  
        }  
}
```

중복 배제 정수 입력

- 정수 10개를 저장할 수 있는 배열을 생성해서 1 ~ 45까지의 숫자를 중복되지 않게 입력하는 Program을 작성

```
public class Main {  
    static final int SIZE = 10;  
  
    public static void main(String[] args) {  
        int num[] = new int[SIZE];  
        Scanner scanner = new Scanner(System.in);
```

중복 배제 정수 입력

```
int i = 0;
while (i < num.length) {
    System.out.printf(" %d번째 정수 입력하세요 ", i + 1);
    num[i] = scanner.nextInt();
    if (num[i] >= 1 && num[i] <= 45) {
        for (int j = 0; j < i; j++) {
            if (num[i] == num[j]) {
                System.out.printf("중복되는 수 입니다 다시입력하세요.
                                   ( 중복숫자 %d)\n", num[j]);

                i--;
            }
        }
        i++;
    } else {
        System.out.println("입력 오류");
    }
}
```

중복 배제 정수 입력

```
System.out.println("당신이 입력한 숫자는 : ");  
for (i = 0; i < num.length; i++) {  
    System.out.print(num[i] + " ");  
}  
}  
}
```

Lotto 번호

- 1 ~ 45까지의 숫자 6개를 랜덤으로 생성하고, 사용자가 입력한 숫자를 비교하여 6개 전부를 맞추면 1등, 5개를 맞췄으면 2등, 4개를 맞췄으면 3등, 3개를 맞췄으면 3등, 그 외는 '당첨되지 않았습니다'라는 Message가 출력되는 Program을 작성하여라



Lotto 번호

- Random Number
 - Softwarw를 개발할 때 난수(Random Number)에 대한 필요성은 빈번히 발생
 - 특히 Game은 주사위를 던지거나 Card를 섞을 때 난수를 사용
- JAVA에서 난수를 발생시키는 2가지 방법
 - Math 클래스의 random() 메소드
 - Random 클래스

Lotto 번호

- Random Class 이용하여 난수 발생하기

```
Random random = new Random();
```

①

```
int test = random.nextInt();
```

②

- $-2,147,483,648 < \text{test} < 2,147,483,647$

- Random Class를 이용해서 원하는 범위내의 난수를 발생하려면

```
int r = random.nextInt(<큰수> - <작은수> + 1) + <작은수>;
```

- 예) 1 ~ 100 사이의 값

```
int r = random.nextInt(100 - 1 + 1) + 1 ;
```

```
int r = random.nextInt(100) + 1 ;
```

Lotto 번호

```
public static void main(String[] args) throws IOException {  
    Random Random = new Random();  
    Scanner keyboard = new Scanner(System.in);  
    int[] lotto = new int[6];  
    int[] mylotto = new int[6];  
    boolean flag = false;  
    String result = "";
```


Lotto 번호

```
int i = 0;
while (i < lotto.length) {
    int temp = Random.nextInt(45) + 1; // 1~45까지
    for (int j = 0; j <= i; j++) {
        if (temp == lotto[j])
            break;
        else
            lotto[i] = temp;
    }
    i++;
}
```

Lotto 번호

```
i = 0;
while (i < mylotto.length) {
    System.out.printf("%d/%d번째 Lotto 숫자 입력 : ", i + 1,
                      mylotto.length);

    int temp = keyboard.nextInt();
    if (temp >= 1 && temp <= 45) {
        for (int j = 0; j <= i; j++) {
            if (i != 0 && j == i)
                break;
            if (temp == mylotto[j]) {
                flag = false;
                System.out.print("숫자가 중복되었습니다");
                System.in.read();
                break;
            }
        }
    }
}
```

Lotto 번호

```
        } else {  
            mylotto[i] = temp;  
            flag = true;  
        }  
    }  
    if (flag == true)  
        i++;  
} else {  
    System.out.print("1 ~ 45사이의 숫자 입력해 주세요");  
    System.in.read();  
}  
}
```

Lotto 번호

```
int check = 0;
for (i = 0; i < lotto.length; i++) {
    for (int j = 0; j < mylotto.length; j++) {
        if (lotto[i] == mylotto[j]) {
            check++;
        }
    }
}
```

```
switch (check) {
    case 6:
        result = "1등";
        break;
    case 5:
        result = "2등";
        break;
```

Lotto 번호

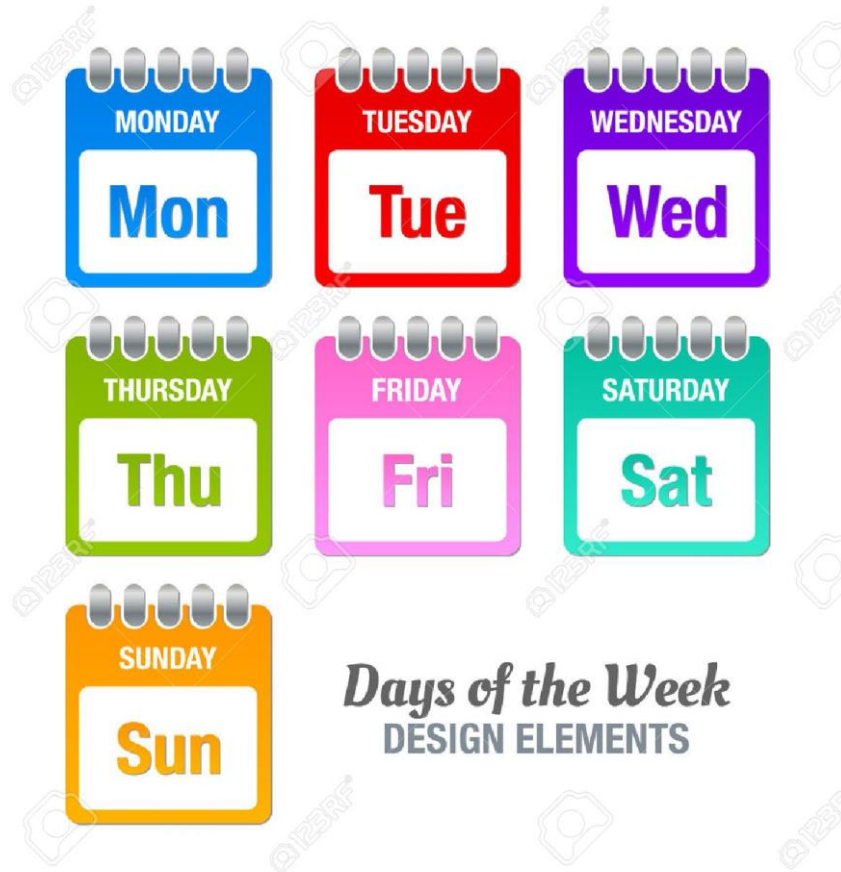
```
case 4:  
    result = "3등";  
    break;  
case 3:  
    result = "4등";  
    break;  
case 2:  
    result = "5등";  
    break;  
case 1:  
    result = "6등";  
    break;  
default:  
    result = "당첨되지 않았습니다.";  
}
```

Lotto 번호

```
System.out.print("생성된 로또 번호 : ");
for (i = 0; i < lotto.length; i++)
    System.out.print(" " + lotto[i]);
System.out.println();
System.out.print("구입한 로또 번호 : ");
for (i = 0; i < mylotto.length; i++)
    System.out.print(" " + mylotto[i]);
if (result.length() == 2)
    System.out.printf("\n%s에 당첨되었습니다.\n", result);
else
    System.out.printf("\n당첨되지 않았습니다.\n");
}
```

내가 좋아하는 요일

- 내가 좋아하는 요일을 출력하는 프로그램을 작성하여라

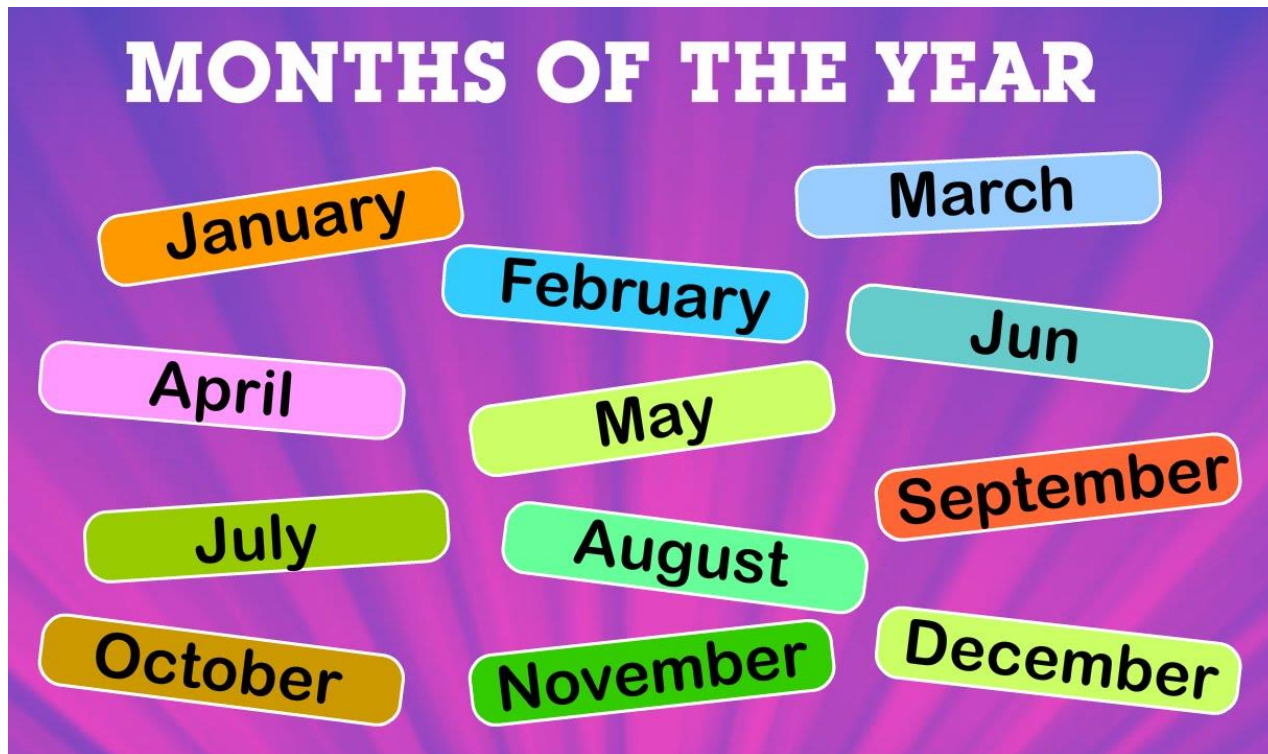


내가 좋아하는 요일

```
public static void main(String[] args) {  
    String week[] = new String[7];  
    week[0] = "월요일";  
    week[1] = "화요일";  
    week[2] = "수요일";  
    week[3] = "목요일";  
    week[4] = "금요일";  
    week[5] = "토요일";  
    week[6] = "일요일";  
  
    System.out.println("제일 좋아하는 요일은 " + week[5] + " 입니다.");  
}
```


월(month) 이름

- 1 ~ 12월을 입력 받아 이름을 영어로 출력하는 프로그램을 작성하여라.



월(month) 이름

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    int month;  
    String[] name = {"January", "February", "March", "April", "May",  
                    "Jun", "July", "August", "September", "October",  
                    "November", "December"};  
  
    do {  
        System.out.print(" 원하시는 달은 ? ");  
        month = keyboard.nextInt();  
        if (month >= 1 && month <= 12)  
            break;  
        System.out.println("ERROR : 달(month)의 범위를 넘었다.");  
    } while (true);  
    System.out.printf(" %d월은 영어로 %s입니다.\n", month,  
                      name[month - 1]);  
}
```

일주일(week)의 이름 초기화 방법

- 일주일의 이름을 초기화하고, 좋아하는 요일을(월요일은 1, 일요일은 7)을 숫자로 입력하면 요일 이름을 출력하는 프로그램을 작성하여라

일주일(week)의 이름 초기화 방법

```
import java.util.Scanner;
public class Week {
    public static void main(String args[]) {
        Scanner keyboard = new Scanner(System.in);
        String[ ] weekname = new String[7];
        weekname[0] = "월요일"; weekname[1] = "화요일";
        weekname[2] = "수요일"; weekname[3] = "목요일";
        weekname[4] = "금요일"; weekname[5] = "토요일";
        weekname[6] = "일요일";
        int test;
        do {
            System.out.printf("Wn 좋아하는 요일은 ? (월요일 = 1, 일요일 = 7) ");
            test = keyboard.nextInt( );
            if (test >= 1 && test <= 7)
                break;
            System.out.printf("Wn ERROR : 일주일(week)의 범위를 넘었다.");
        } while (true);
        System.out.println("제일 좋아하는 요일은 " + weekname[test - 1] + " 입니다.");
    }
}
```

일주일(week)의 이름 초기화 방법

```
import java.util.Scanner;

public class Week1 {
    public static void main(String args[]) {
        Scanner keyboard = new Scanner(System.in);
        String[ ] weekname = {"월요일", "화요일", "수요일", "목요일", "금요일", "토요일",
                               "일요일"};

        int test;
        do {
            System.out.printf("\n 좋아하는 요일은 ? (월요일 = 1, 일요일 = 7) ");
            test = keyboard.nextInt( );
            if (test >= 1 && test <= 7)
                break;
            System.out.printf("\n ERROR : 일주일(week)의 범위를 넘었다.");
        } while (true);
        System.out.println("제일 좋아하는 요일은 " + weekname[test - 1] + " 입니다.");
    }
}
```

월(month)의 날자 수 초기화 방법

- 1 ~ 12월의 날자 수를 배열에 기억시켜 두고 월(1 ~ 12)을 입력하여 그에 대응하는 월의 날자 수를 출력하는 프로그램 작성하여라. (단, 2월 달은 28일로 한다)

월(month)의 날짜 수 초기화 방법

```
import java.util.Scanner;
public class Month {
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        int test;
        int[] month = new int[12]; // 배열 선언
        month[0] = 31; month[1] = 28; month[2] = 31; // 배열 원소 초기화
        month[3] = 30; month[4] = 31; month[5] = 30;
        month[6] = 31; month[7] = 31; month[8] = 30;
        month[9] = 31; month[10] = 30; month[11] = 31;
        do {
            System.out.printf("\n 원하시는 달은 ? ");
            test = keyboard.nextInt( );
            if (test >= 1 && test <= 12)
                break;
            System.out.printf("\n ERROR : 달(month)의 범위를 넘었다.");
        } while (true);
        System.out.printf(" %d월은 %d일 입니다.\n", test, month[test - 1]);
    }
}
```

월(month)의 날짜 수 초기화 방법

```
import java.util.Scanner;
public class Month1 {
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        int test;
        int[] month = new int[12]; // 배열 선언
        month = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}; // 배열 원소 초기화
        do {
            System.out.printf("\n원하시는 달은 ? ");
            test = keyboard.nextInt( );
            if (test >= 1 && test <= 12)
                break;
            System.out.printf("\n ERROR : 달(month)의 범위를 넘었다.");
        } while (true);
        System.out.printf(" %d월은 %d일 입니다.\n", test, month[test - 1]);
    }
}
```

오류

월(month)의 날짜 수 초기화 방법

```
import java.util.Scanner;
public class Month1 {
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        int test;
        // 배열 선언과 초기화
        int[] month = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

        do {
            System.out.printf("Wn 원하시는 달은 ? ");
            test = keyboard.nextInt( );
            if (test >= 1 && test <= 12)
                break;
            System.out.printf("Wn ERROR : 달(month)의 범위를 넘었다.");
        } while (true);
        System.out.printf(" %d월은 %d일 입니다.Wn", test, month[test - 1]);
    }
}
```

월(month)의 날자 수 초기화 방법

```
import java.util.Scanner;
public class Month1 {
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        int test;
        int[12] month = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

        do {
            System.out.printf("\n 원하시는 달은 ? ");
            test = keyboard.nextInt( );
            if (test >= 1 && test <= 12)
                break;
            System.out.printf("\n ERROR : 달(month)의 범위를 넘었다.");
        } while (true);
        System.out.printf(" %d월은 %d일 입니다.\n", test, month[test - 1]);
    }
}
```

JAVA 성적 평균 계산하기

- 키보드로부터 학생 10명의 JAVA 성적을 입력 받아 합계와 평균, 최대값, 최소값을 구하는 프로그램을 단순 변수와 배열을 사용하는 경우로 나누어 작성해보자.

- 목적

- 배열의 편리성
- 배열 선언 방법
- 배열 초기화 방법
- 배열의 사용법 확인



JAVA 성적 평균

■ 문제 분석

■ 입력

■ 정수 배열 : `int score[10]` (정수형) : 0 ~ 100 사이 값

■ 출력

■ 합계 : `total` (정수형)

■ 평균 : `average` (float 형)

■ 최대값 : `max` (정수형)

■ 최소값 : `min` (정수형)

실행시

10개의 정수 데이터 입력

1 : _

2 : -

출력 결과

합계 :

평균 :

최대값 :

최소값 :

JAVA 성적 평균

■ 문제 분석

■ 계산 방법

■ 10개 정수들은 배열에 저장하여라

■ 정수 N개에 대한 정수형 배열 x[N]에 저장할 때,

$$\text{총점(sum)} = \sum_{i=0}^{N-1} x[i] \quad \text{평균(ave)} = \frac{1}{N} \sum_{i=0}^{N-1} x[i]$$

■ 최대값(max) = 최대 값을 찾는다

■ 최소값(min) = 최소 값을 찾는다

JAVA 성적 평균

1. $i = 0$ 부터 9까지 $score[i]$ 를 읽어 배열에 저장
2. $i = 0$ 부터 9까지 다음 식을 계산

$$total = \sum_{i=0}^9 score[i]$$

3. 평균을 구한다: $average = (float) total / 10;$
4. 최소값을 구한다
5. 최대값을 구한다
6. $total$, $average$, 최대값, 최소값을 출력한다

JAVA 성적 평균

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    final int STUDENT = 10;  
    int[ ] score = new int[STUDENT];  
    int total = 0;  
    float average;  
    int max = 0;  
    int min = 100;  
    int i = 0;
```

JAVA 성적 평균

```
while (i < score.length) {  
    System.out.printf(" %d번 학생의 JAVA 성적을 입력: ", i + 1);  
    score[i] = keyboard.nextInt();  
    if (score[i] >= 0 && score[i] <= 100)  
        i++;  
    else {  
        System.out.println("ERROR : 0점에서 100점 사이의 값 입력 바람");  
    }  
}  
  
for (i = 0; i < score.length; i++)  
    total += score[i];  
average = (float) total / STUDENT;
```


JAVA 성적 평균

```
for (i = 0; i < score.length; i++) {  
    if (max < score[i])  
        max = score[i];  
}
```

```
for (i = 0; i < score.length; i++) {  
    if (min > score[i])  
        min = score[i];  
}
```

```
for (i = 0; i < score.length; i++) {  
    System.out.println("score[" + i + "] = " + score[i]);  
}
```

```
System.out.println("총점 성적은 " + total + " 입니다");  
System.out.println("평균 성적은 " + average + " 입니다");  
System.out.println("최소값은 " + min + " 입니다");  
System.out.println("최대값은 " + max + " 입니다");
```

```
}
```

JAVA 성적 가중치 평균

- JAVA 성적의 중간, 기말, 퀴즈, 과제, 출석 점수가 모두 100점 만점으로 작성되어 있다. 각 부문별 점수를 입력 받아서 단순 평균 점수와 가중치 평균 점수를 구하는 프로그램을 작성하여라.
 - 각 점수의 반영 비율은 각각 중간 15%, 기말 15%, 퀴즈 10%, 과제 40%, 출석 20%이다
 - 가중치 점수는 각 점수에 가중치를 곱하여 더한 값이다



JAVA 성적 가중치 평균

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    String[] subject = {"중간고사", "기말고사", "퀴즈", "과제", "출석"};  
    int[] termscore = new int[subject.length];  
    float[] weight = {0.15f, 0.15f, 0.1f, 0.4f, 0.2f};  
    float[] score = new float[2];  
    int i = 0;  
  
    while (i < termscore.length) {  
        System.out.print(subject[i] + " 점수 입력 : ");  
        termscore[i] = keyboard.nextInt();  
        if (termscore[i] >= 0 && termscore[i] <= 100)  
            i++;  
        else  
            System.out.println("ERROR");  
    }  
}
```

JAVA 성적 가중치 평균

```
float temp = 0.0f;
for (i = 0; i < termscore.length; i++) {    /* 단순 평균 */
    temp += termscore[i];
    score[0] = temp / termscore.length;
}
```

```
for (i = 0; i < termscore.length; i++)    /* 가중치를 반영한 점수 계산 */
    score[1] += termscore[i] * weight[i];
```

```
System.out.printf(" 단순 평균 점수 = %.1f\n", score[0]);
System.out.printf(" 가중 평균 점수 = %.1f\n", score[1]);
```

```
}
}
```

JAVA 성적의 표준편차

- 한 학급(10명)의 JAVA 성적의 표준 편차를 구하여라.

JAVA 성적의 표준편차

- 문제 분석
 - 입력 변수
 - JAVA성적 배열 : score[10] (정수형)
 - 출력 변수
 - 총점 : total (정수형)
 - 평균 : ave (float 형)
 - 표준편차 : sd (float 형)

실행 시

10개의 JAVA점수 데이터 입력

학생 1 : _

학생 2 : -

출력 결과

총점 :

평균 :

표준편차 :

JAVA 성적의 표준편차

■ 문제 분석

■ 학생 N명에 대한 성적을 실수형 배열 x[N]에 저장할 때,

■ 총점(total) = $\sum_{i=0}^{N-1} x[i]$ 평균(ave) = $\frac{1}{N} \sum_{i=0}^{N-1} x[i]$

■ 분산 = $\frac{1}{N} \sum x_i^2 - ave^2 = \frac{1}{N} \sum (x_i - ave)^2$

■ 표준편차(sd) = $\frac{1}{N} \sum x_i^2 - 2ave \frac{1}{N} \sum x_i + ave^2 = \sqrt{\frac{1}{N} \sum x_i^2 - ave^2}$

JAVA 성적의 표준편차

1. i=0부터 9까지 score[i]를 읽어 배열에 저장
2. i=0부터 9까지 다음 식을 계산
 - 1) $total = \sum_{i=0}^9 score[i]$
 - 2) $total2 = \sum_{i=0}^9 (score[i] * score[i])$
3. 평균을 구함 : $ave = total / 10.0$
4. 표준편차를 구함 : $sd = \sqrt{total2/10.0 - ave * ave}$
5. total, ave, sd를 출력

JAVA 성적의 표준편차

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    int[] score = new int[10];  
    int total = 0, total2 = 0;  
    double average, sd;  
  
    System.out.println("학생 10명의 성적 입력");  
    for (int i = 0; i < score.length; i++) {  
        System.out.printf("학생 %d 점수 입력 : ", i + 1);  
        score[i] = keyboard.nextInt();  
    }  
  
    for (int i = 0; i < score.length; i++){  
        total += score[i];  
        total2 += score[i] * score[i];  
    }  
}
```

JAVA 성적의 표준편차

```
average = (double) total / score.length;  
sd = Math.sqrt( (double) total2 / score.length - average * average);  
  
System.out.printf("\n총점 : %,d\n", total);  
System.out.printf("평균 : %.2f\n", average);  
System.out.printf("표준편차 : %.2f\n", sd);  
}
```

Fibonacci 수열

- 배열을 이용하여 Fibonacci 수열을 10개만 구하시오.
- Fibonacci 수열
 - 이탈리아의 수학자 피보나치(Fibonacci, 1175~1250)는 아라비아에서 발전된 수학을 유럽에 소개하여 유럽 여러 나라의 수학을 발전시키는 데 크게 기여하였다.
 - 피사의 레오날도라고도 불리우는 그는 사라센제국의 회교도권 수학을 유럽의 그리스도교 국가로 소개한 공헌을 수학사에 남겼는데 피보나치는 1202년 자신의 저서 ‘산반서(Liber abaci)’에서 다음과 같은 토끼의 번식에 대한 문제를 제시하였다.

Fibonacci 수열

- 배열을 이용하여 피보나치 수열을 10개만 구하시오.
- 피보나치 수열
 - 갓 태어난 토끼 암수 한 쌍이 있다. 이 토끼 한 쌍은 태어난 지 두 달이 되는 달부터 매달 암수 한 쌍의 토끼를 낳으며, 새로 태어난 토끼 한 쌍도 태어난 지 두 달이 되는 달부터 매달 암수 한 쌍의 토끼를 낳는다. 일 년 후 토끼는 모두 몇 쌍이 될까? (단, 토끼는 중간에 죽지 않는다.)
 - 이를 수열로 나타내면,
 - 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, ...
 - 따라서 피보나치 수열이란 어떤 수열의 항이, 앞의 두 항의 합과 같은 수열을 말함.
 - 이 성질을 발견한 레오나르도 피보나치(Leonardo Fibonacci)의 이름을 따서 피보나치 수열이라 이름 붙임

Fibonacci 수열

- Fibonacci 수열의 신비(특징)
 - 수열의 구성을 살펴보면 앞의 두 수를 더하면 다음 숫자가 나오게 된다.
 - 어느 숫자이건 하나 건너 숫자로 나누면 그 몫은 2가 된다. (예: $8 \div 3 = 2 \cdots 2$; 몫2) 그리고 나머지 값은 그 직전의 값이 된다. (예: $13 \div 5 = 2 \cdots 3$; 5의 직전의 값 3)
 - 바로 앞의 숫자를 뒤의 숫자로 나누면 그 값은 1.6180이라는 숫자에 근접하게 됨. 이것을 황금비라고 함
 - 큰 수에서 하나 건너 작은 숫자로 나누면 그 값은 2.618이라는 숫자에 근접함
 - 인접한 두 항의 최대공약수는 유클리드 호제법으로 증명함

Fibonacci 수열

```
public static void main(String[] args) {  
    int[] num = new int[10];  
    num[0] = 1;  
    num[1] = 1;  
  
    for (int i = 2; i < num.length; i++)  
        num[i] = num[i-2] + num[i-1];  
  
    for (int i = 0; i < num.length; i++)  
        System.out.print(num[i] + ", ");  
  
    System.out.println("\n b b");  
}
```

제일 큰 수와 가장 작은 수 찾기

- Keyboard로부터 10개의 정수 Data를 입력 받아 그 중 제일 큰 값과 가장 작은 값을 찾아 출력하는 Program을 작성하라
- 목적
 - 배열
 - 배열 이름[개수]
 - 같은 타입의 모임
 - 10개의 정수 표현하기
 - 배열 선언 방법
 - 배열 초기화 방법
 - 배열의 사용법 확인

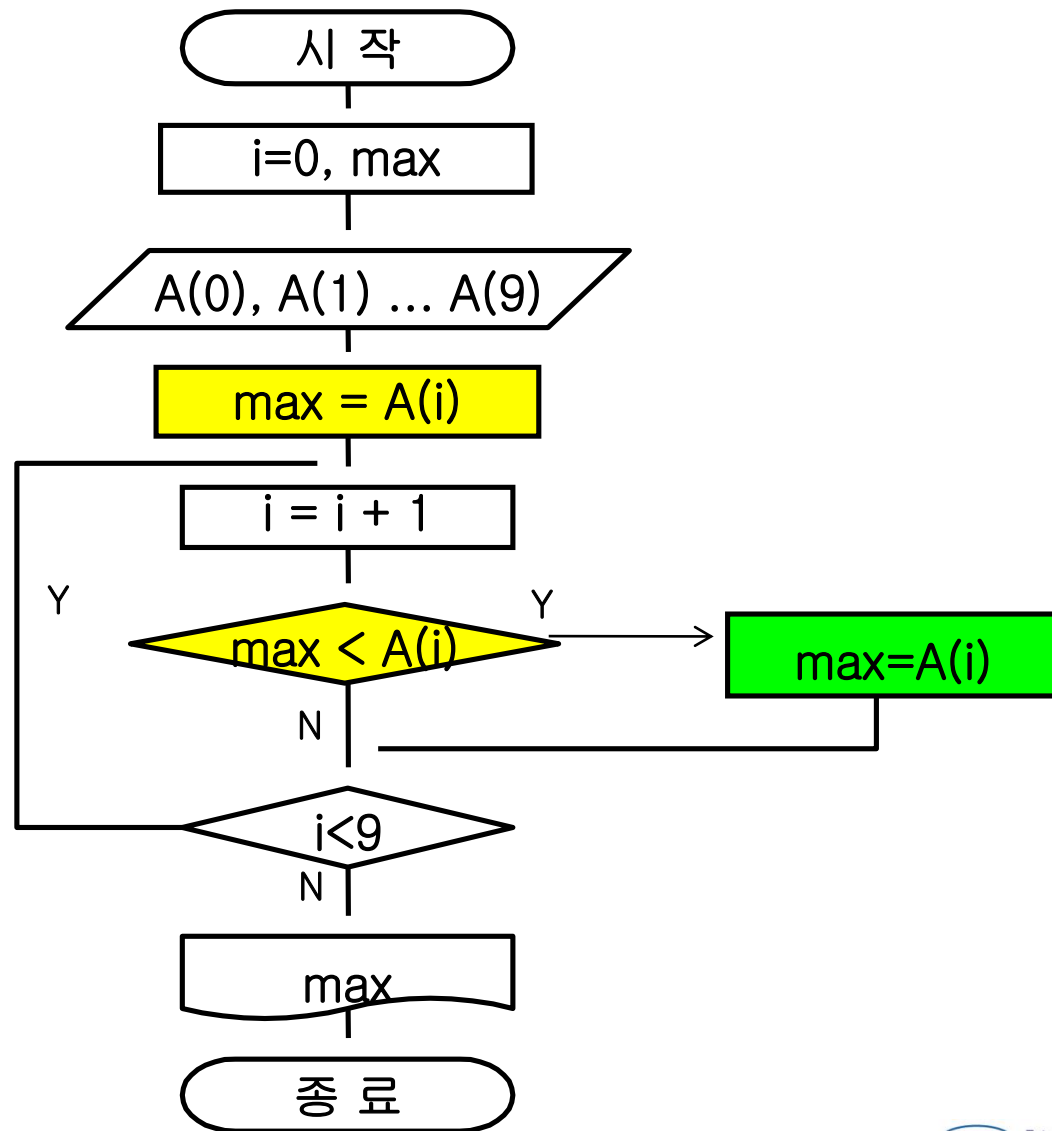
제일 큰 수와 가장 작은 수 찾기

■ 접근법

- 10개의 정수를 줄을 세워 들어오게 함
- 지금까지 읽은 수 중에 가장 큰 수 자리에 대기시킴
- 들어온 수와 가장 큰 수 자리와 비교 함
- 비교하여 들어온 수가 크면 가장 큰 수 자리에 대기시킴
- 모두 다 들어 올 때까지 반복

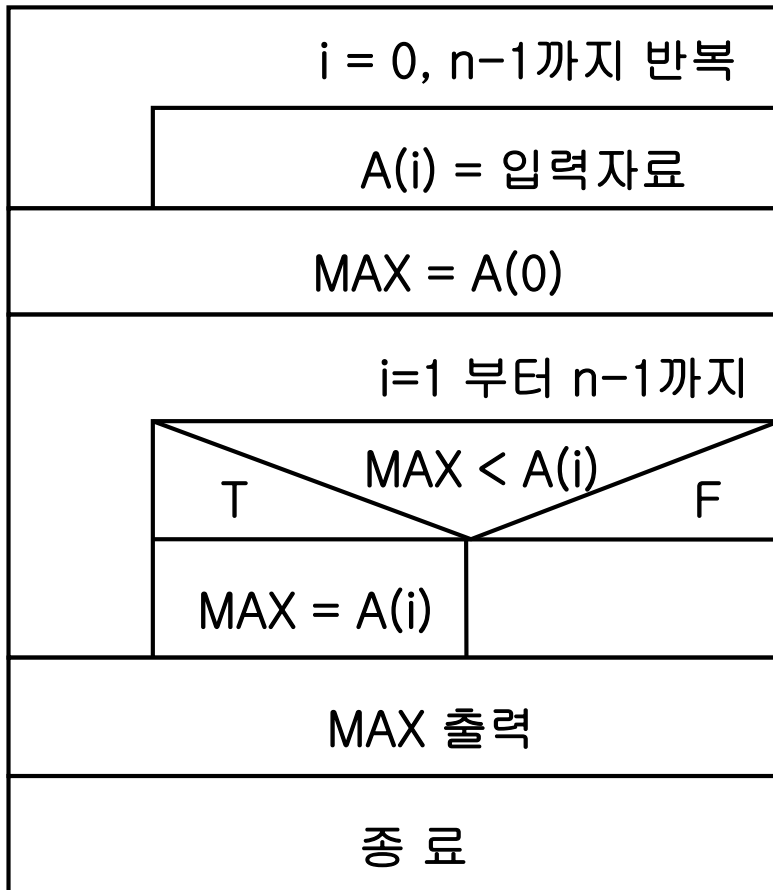


제일 큰 수와 가장 작은 수 찾기



제일 큰 수와 가장 작은 수 찾기

- n개의 자료를 읽어 들여 최대값을 구하는 NS chart 와 의사 코드가 다음과 같다



```
i=0  
  
WHILE (i<n)  
    A(i) = key_input  
ENDWHILE  
  
MAX = A(0)  
  
FOR i=2 n-1 DO  
    IF (MAX < A(i)) THEN  
        MAX = A(i)  
    ENDIF  
ENDFOR  
  
PRINT MAX
```

제일 큰 수와 가장 작은 수 찾기

```
public static void main (String[] args) {  
    Scanner in = new Scanner(System.in);  
    final int SIZE = 10;  
    int[] intArray = new int[SIZE];  
    int max = 0;  
    int min = 0;  
    for (int i = 0; i < SIZE; i++) {  
        System.out.printf(" %d/%d 번째 정수값 입력 : ", i + 1, SIZE);  
        intArray[i] = in.nextInt();  
    }  
    max = intArray[0];  
    min = intArray[0];  
    for (int i = 0; i < SIZE; i++) {  
        if (intArray[i] > max)  
            max = intArray[i];  
        if (intArray[i] < min)  
            min = intArray[i];  
    }  
    System.out.println("입력된 수에서 가장 큰 수는 " + max + "입니다.");  
    System.out.println("입력된 수에서 가장 작은 수는 " + min + "입니다.");  
}
```

학급 반대표 선거

- 어떤 학급의 반대표를 뽑는데 4명의 후보가 나왔다. 각각의 후보에게 번호가 1번부터 4까지 붙여졌을 때, 투표자의 수와 각 후보자의 득표 결과, 그리고 무효표의 수를 출력하는 Program을 작성하시오.

- 목표

- 1차원 배열의 인덱스(첨자) 사용에 관하여 공부

학급 반대표 선거

- 입후보자 개인마다 번호가 주어졌으므로, 투표자들은 각 후보자의 번호를 사용하여 투표한다고 가정한다.
(즉, 입력 데이터가 입후보자의 번호, 즉 정수로 들어온다고 가정)
- 1~4번 이외의 입력이 들어오면 무효 표로 간주
- 투표자의 수를 정확히 알 수 없으므로 더 이상의 입력이 없을 때까지 개표를 진행

학급 반대표 선거

■ 예) 투표 상황

투표자	투표결과
1	3
2	2
3	2
4	2
5	4
6	3
7	2
8	1
9	5
10	3
11	3
12	4
13	2
14	1
15	1
16	2

득표결과

후보	1	2	3	4	무효표
득표수	3	6	4	2	1

총투표인 수 : 16명



학급 반대표 선거

■ 문제 분석

- 1) 투표자들이 숫자를 입력하는 동안 계속해서 입력을 받음
- 2) 입력 받은 정수 값이 1 ~ 4 사이의 값이 아닌 경우에는 무효 표를 1 증가
- 3) 각 후보의 득표수를 저장하기 위하여 득표수 배열을 선언하되 각 후보의 번호를 인덱스로 사용
- 4) 입력 받은 정수 값이 1 ~ 4 사이의 값인 경우에는 입력 받은 정수를 인덱스로 하는 배열 원소 값을 1 증가
- 5) 더 이상 입력이 없으면 (즉 숫자 입력이 끝나면) 결과를 출력

학급 반대표 선거 : 입 출력 설계

■ 입력 데이터

- 입력 값 : xdata

■ 출력 데이터

- 각 후보의 득표수 : ip[4]

- 무효표 수 : nogood

■ 실행 시

- 1 ~ 4번 중 한 후보의 번호를 입력 _

출력

득표수

1번	:	3표
2번	:	6표
3번	:	4표
4번	:	2표
무효표	:	1표

총 16표

학급 반대표 선거 : 가상 언어 표현

1. 각 후보의 득표수 배열 $ip[i]$ 를 0으로 초기화
($i = 0 \sim 3$)
2. 데이터를 입력 받는다
3. 입력된 데이터가 번호인 동안
 - 3.1 총투표수(n)를 1 증가시킨다
 - 3.2 만일 입력 데이터가 1 ~ 4 사이의 값이면
입력된 후보의 득표수 배열 값을 1 증가
그렇지 않으면 무효 표 수를 1 증가
 - 3.3 데이터를 입력 받는다
4. 각 후보의 득표수, 무효 표 수, 총 투표수를 출력

학급 반대표 선거

```
public static void main(String[] args) throws IOException {
    Scanner keyboard = new Scanner(System.in);
    int NMAX;
    int Voter;
    int temp;
    int votedata;
    int count;
    int nogood = 0;
    char answer;

    // 투표자 누적 인원
    // 무효표 누적

    while (true) {
        System.out.print("Wn 후보자가 몇명 입니까 ? ");
        temp = keyboard.nextInt( );
        if (temp > 1) {
            NMAX = temp;
            break;
        } else {
            System.err.printf("Wn 입력 오류 입니다.");
            System.in.read( );
        }
    }
}
```

학급 반대표 선거

```
int[] man = new int[NMAX];    // 입후보자용 득표 현황 누적용
for (int i = 0 ; i < NMAX ; ++i)
    man[i] = 0;                // 입후보자 득표 현황 초기화

while (true) {
    System.out.print("\n 총 유권자가 몇명 입니까 ? ");
    temp = keyboard.nextInt( );
    if (temp > 1) {
        Voter = temp;
        break;
    } else {
        System.err.printf("\n 입력 오류 입니다.");
        System.in.read( );
    }
}
```

학급 반대표 선거

```
count = 0;
while (true) {
    System.out.print("\n 선거 시간이 종료 되었나요? (Y/N) ");
    answer = (char) System.in.read( );
    if (answer == 'Y' || answer == 'y')
        break;
    System.out.printf("\n 1 - %d번 중 한 후보의 번호를 입력하시오.", NMAX) ;
    votedata = keyboard.nextInt( );
    count++;
    if (votedata >= 1 && votedata <= NMAX)
        man[votedata - 1]++;
    else
        nogood++;
    if (count == Voter)
        break;
}
```

학급 반대표 선거

```
System.out.print("WnWnWn 득표수WnWn");  
for (int i = 0; i < NMAX; i++)  
    System.out.printf("%d번 : %d표Wn", i + 1, man[i]);  
System.out.printf("무효표 : %d표 Wn", nogood);  
System.out.printf("총투표수 : %d표 Wn", count);  
}
```

Data 오름차순 정렬(Bubble)

- 10개의 정수 Data를 입력 받아서 작은 값에서 큰 값의 순으로 정렬하는 Program을 작성하여라

양의 정수 입력 : 34

양의 정수 입력 : 56

양의 정수 입력 : 78

양의 정수 입력 : 90

양의 정수 입력 : 12

양의 정수 입력 : 32

양의 정수 입력 : 56

양의 정수 입력 : 34

양의 정수 입력 : 17

양의 정수 입력 : 85

Data 오름차순 정렬(Bubble)

원본 데이터 : 34, 56, 78, 90, 12, 32, 56, 34, 17, 85

Step 1 : 12, 56, 78, 90, 34, 32, 56, 34, 17, 85

Step 2 : 12, 17, 78, 90, 56, 34, 56, 34, 32, 85

Step 3 : 12, 17, 32, 90, 78, 56, 56, 34, 34, 85

Step 4 : 12, 17, 32, 34, 90, 78, 56, 56, 34, 85

Step 5 : 12, 17, 32, 34, 34, 90, 78, 56, 56, 85

Step 6 : 12, 17, 32, 34, 34, 56, 90, 78, 56, 85

Step 7 : 12, 17, 32, 34, 34, 56, 56, 90, 78, 85

Step 8 : 12, 17, 32, 34, 34, 56, 56, 78, 90, 85

Step 9 : 12, 17, 32, 34, 34, 56, 56, 78, 85, 90

정렬 데이터 : 12, 17, 32, 34, 34, 56, 56, 78, 85, 90

Data 오름차순 정렬(Bubble)

```
public static void main(String[] args) throws IOException {
    Scanner keyboard = new Scanner(System.in);
    int[] data = new int[10];
    int temp;

    for (int i = 0; i < data.length; i++) {
        System.out.print("양의 정수 입력 : ");
        temp = keyboard.nextInt();
        if (temp >= 1)
            data[i] = temp;
        else {
            System.err.print("입력 오류");
            System.in.read();
        }
    }

    System.out.print("\n원본 데이터 : ");
    for (int i = 0; i < data.length; i++)
        System.out.print(data[i] + ", ");
    System.out.println("\n\n");
}
```


Data 오름차순 정렬(Bubble)

```
for (int i = 0; i < data.length - 1; i++) {  
    for (int j = i + 1; j < data.length; j++) {  
        if (data[i] > data[j]) {  
            temp = data[j];  
            data[j] = data[i];  
            data[i] = temp;  
        }  
    }  
    System.out.print("  Step " + (i + 1) + " : ");  
    for (int k = 0; k < data.length; k++)  
        System.out.print(data[k] + ", ");  
    System.out.println("WbWb");  
}
```

```
System.out.print("\n정렬 데이터 : ");  
for (int i = 0; i < data.length; i++)  
    System.out.print(data[i] + ", ");  
System.out.println("WbWb");  
}
```

Data 오름차순 정렬(Insertion)

```
public static void main(String[] args) throws IOException {
    Scanner keyboard = new Scanner(System.in);
    int[] data = new int[10];
    int temp;

    for (int i = 0; i < data.length; i++) {
        System.out.print("양의 정수 입력 : ");
        temp = keyboard.nextInt();
        if (temp >= 1)
            data[i] = temp;
        else {
            System.err.print("입력 오류");
            System.in.read();
        }
    }

    System.out.print("\n원본 데이터 : ");
    for (int i = 0; i < data.length; i++)
        System.out.print(data[i] + ", ");
    System.out.println("\n\n");
}
```

Data 오름차순 정렬(Insertion)

```
for (int i = 1; i < data.length; i++) {  
    for (int j = i; j > 0; j--)  
        if (data[j - 1] > data[j]) {  
            temp = data[j - 1];  
            data[j - 1] = data[j];  
            data[j] = temp;  
        }  
  
    System.out.print("  Step " + i + " : ");  
    for (int k = 0; k < data.length; k++)  
        System.out.print(data[k] + ", ");  
    System.out.println("WbWb");  
}  
  
System.out.print("\n정렬 데이터 : ");  
for (int i = 0; i < data.length; i++)  
    System.out.print(data[i] + ", ");  
System.out.println("WbWb");  
}
```

Data 오름차순 정렬(Selection)

```
public static void main(String[] args) throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    int[] data = new int[10];  
    int temp;  
  
    for (int i = 0; i < data.length; i++) {  
        System.out.print("양의 정수 입력 : ");  
        temp = keyboard.nextInt();  
        if (temp >= 1)  
            data[i] = temp;  
        else {  
            System.err.print("입력 오류");  
            System.in.read();  
        }  
    }  
  
    System.out.print("\n원본 데이터 : ");  
    for (int i = 0; i < data.length; i++)  
        System.out.print(data[i] + ", ");  
    System.out.println("\n\n");  
}
```

Data 오름차순 정렬(Selection)

```
int indexMin;
for (int i = 0; i < data.length-1; i++) {           // 1.
    indexMin = i;
    for (int j = i + 1; j < data.length; j++) {      // 2.
        if (data[j] < data[indexMin]) {              // 3.
            indexMin = j;
        }
    }
    temp = data[indexMin];
    data[indexMin] = data[i];
    data[i] = temp;

    System.out.print("  Step " + (i + 1) + " : ");
    for (int k = 0; k < data.length; k++)
        System.out.print(data[k] + ", ");
    System.out.println("WbWb");
}
```

Data 오름차순 정렬(Selection)

```
System.out.print("\n정렬 데이터 : ");  
for (int i = 0; i < data.length; i++)  
    System.out.print(data[i] + ", ");  
System.out.println("\b\b");  
}
```

오늘은 그 해의 몇 번째 날인가 ?

- 오늘의 날짜를 입력하면 오늘이 그해의 몇 번째 날인지를 계산하는 프로그램을 작성하여라.



오늘은 그 해의 몇 번째 날인가 ?

```
public class Main {  
    static int days[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        int year, day, month;  
        int totalday = 0;  
  
        do {  
            System.out.print("연도를 입력하세요 (1900 ~ ? ) : ");  
            year = scanner.nextInt();  
        } while (year < 1900);  
  
        if (((year % 4 == 0) && (year % 100 != 0)) || year % 400 == 0)  
            days[1] = 29;
```



Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY

오늘은 그 해의 몇 번째 날인가 ?

```
do {  
    System.out.print("월을 입력하세요 (1 ~ 12) : ");  
    month = scanner.nextInt();  
} while (month < 1 || month > 12);  
  
do {  
    System.out.printf("일을 입력하세요 (1 ~ %d) : ", days[month-1]);  
    day = scanner.nextInt();  
} while (day < 1 || day > days[month-1]);  
  
totalday += day;  
System.out.printf(" %d년 %d월 %d일은 그해의 ", year, month, day);  
while (month - 1 >= 1){  
    totalday += days[month - 1];  
    month--;  
}  
System.out.printf("%d번째 날입니다.", totalday);  
}
```

나의 띠

- 태어난 년도를 입력 받아 띠를 계산하는 프로그램을 작성하라.



나의 띠

- 문제 분석

- 입력

- 출생년도(year) - int

- 출력

- 띠(result) - String

- 계산 방법

- 자(子, 쥐), 축(丑, 소), 인(寅, 호랑이), 묘(卯, 토끼), 진(辰, 용), 사(巳, 뱀), 오(午, 말), 미(未, 양), 신(申, 원숭이), 유(酉, 닭), 술(戌, 개), 해(亥, 돼지)의 순서로 구성

- 출생 년도를 12로 나누어 나머지가 0 -> 신(원숭이)

나의 띠

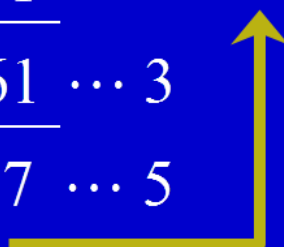
```
public static void main(String[] args) {
    Scanner keyboard = new Scanner(System.in);
    String[] animal = {"원숭이", "닭", "개", "돼지", "쥐", "소",
                      "호랑이", "토끼", "용", "뱀", "용", "양"};

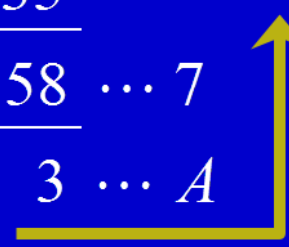
    int year;

    while (true) {
        System.out.print("당신이 태어난 년도를 입력 : ");
        year = keyboard.nextInt();
        if (year > 0)
            break;
        else
            System.out.println("ERROR");
    }
    int index = year % animal.length;
    System.out.println("당신은 " + animal[index] + "띠 입니다");
}
```

진법(2 ~ 16진법) 변환

- 자연수를 입력 받아 원하는 진법(2~16진법)의 수로 변경하는 프로그램을 작성하여라.

$$\begin{array}{r} 8 \overline{) 491} \\ 8 \overline{) 61} \dots 3 \\ \quad 7 \dots 5 \end{array}$$

$$491_{(10)} = 753_{(8)}$$

$$\begin{array}{r} 16 \overline{) 935} \\ 16 \overline{) 58} \dots 7 \\ \quad 3 \dots A \end{array}$$

$$935_{(10)} = 3A7_{(16)}$$

진법(2 ~ 16진법) 변환

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    char[] digit = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9',  
                    'A', 'B', 'C', 'D', 'E'};  
  
    int value, decimal, base;  
    String result = "";  
  
    while (true) {  
        System.out.print(" 자연수 입력 : ");  
        value = keyboard.nextInt();  
        System.out.print(" 원하는 진법 : ");  
        base = keyboard.nextInt();  
        if (value > 0 && (base >= 2 && base <= 16))  
            break;  
        else  
            System.out.println("ERROR : 다시 입력해 주세요");  
    }  
}
```

진법(2 ~ 16진법) 변환

```
decimal = value;
```

```
while (value != 0) {  
    int index = value % base;  
    result = Character.toString(digit[index]) + result;  
    value /= base;  
}
```

```
System.out.printf(" %d의 %d 진법의 값은 %s 입니다\n",  
                    decimal, base, result);  
}
```

60 갑자

- 원하는 년도를 입력하면 60갑자의 간지(干支)를 출력하는 프로그램을 작성하여라



60 갑자

■ 60갑자

- 10간(干)과 12지(支)를 결합하여 만든 60개의 간지(干支)를 말함

■ 간(干)

- 하늘의 시간을 나타내는 10개의 간(干)으로 구성된다.
- 갑(甲), 을(乙), 병(丙), 정(丁), 무(戊), 기(己), 경(庚), 신(辛), 임(壬), 계(癸)의 순서로 구성

■ 지(支)

- 땅을 지키는 12개의 동물로 12지지(地支)라고 함
- 자(子, 쥐), 축(丑, 소), 인(寅, 호랑이), 묘(卯, 토끼), 진(辰, 용), 사(巳, 뱀), 오(午, 말), 미(未, 양), 신(申, 원숭이), 유(酉, 닭), 술(戌, 개), 해(亥, 돼지)의 순서로 구성

60갑자

■ 문제 분석

■ 입력

■ 년도(year) – int

■ 출력

■ 60갑자(result) – String

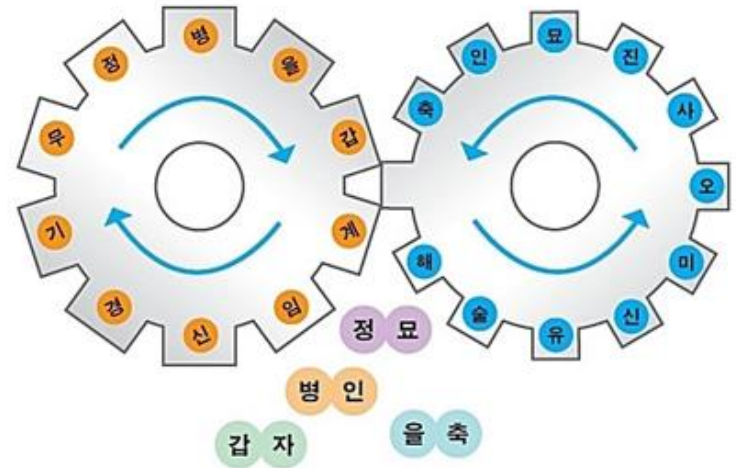
■ 계산 방법

■ 10간은 10년마다 반복되므로 년도의 끝자리 수를 보면 쉽게 이해할 수 있다

■ 12지도 12년마다 반복이 된다.

■ 년도를 12로 나눈다. 그리고 나머지가 얼마인지 알면 된다.

■ 10간에서 '갑'이 4부터 시작하고, 12지도 '자'가 4부터 시작한다



60갑자

■ 예

■ 1592년(임진년)

년도의 끝자리 수 2 → 10간은 '임'이 됨

$1592 / 12 = 132 + 8$ (나머지)

년도를 12로 나눈 나머지 8 → 12지는 '진'이 됨

연도의 끝자리 수와 10간(十干)										
연도의 끝자리 수	4	5	6	7	8	9	0	1	2	3
십간(十干)	갑	을	병	정	무	기	경	신	임	계

년도를 12로 나누었을 때 나머지와 십이지(十二支)												
연도/12, 나머지 수	4	5	6	7	8	9	10	11	0	1	2	3
십이지(十二支)	자	축	인	묘	진	사	오	미	신	유	술	해

60 갑자

```
public static void main(String[] args) {
    Scanner keyboard = new Scanner(System.in);
    char gan[ ] = {'경', '신', '임', '계', '갑', '을', '병', '정', '무', '기'};
    char ji[ ] = {'신', '유', '술', '해', '자', '축', '인', '묘', '진', '사', '오', '미'};
    int year, i, j;

    while (true) {
        System.out.print(" 연도 입력 : ");
        year = keyboard.nextInt();
        if (year > 0)
            break;
        else
            System.out.printf("\n ERROR : 입력 오류");
    }
}
```

60 갑자

```
i = year % 10;  
j = year % 12;
```

```
System.out.printf(" %d 년은 %c%c년 입니다.\n", year, gan[i], ji[j]);  
}
```

암호표

- 다음은 알파벳과 숫자를 아래에 주어진 암호표로 암호화하는 프로그램을 작성하여라.

원문	a	b	c	d	e	f	g	h	i	j
암호	'	~	!	@	#	\$	%	^	&	*

원문	k	l	m	n	o	p	q	r	s	t
암호	()	-	_	+	=		[]	{

원문	u	v	w	x	y	z
암호	}	;	:	,	.	/

원문	0	1	2	3	4	5	6	7	8	9
암호	q	w	e	r	t	y	u	i	o	p

암호표

■ 실행 결과

영어문장 입력 : Kyungbok 123 Welcome

원문 : kyungbok 123 welcome

암호 : (.}_%~+(wer :#)!+-#

해독 : kyungbok 123 welcome

암호표

```
public static int linearSearch(char[] arr, char key) {  
    for (int i = 0; i < arr.length; i++) {  
        if (arr[i] == key) {  
            return i;  
        }  
    }  
    return -1;  
}
```



Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY

암호표

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    char[] alphabetCode = {'\'', '~', '!', '@', '#', '$', '%', '^', '&', '*',  
                           '(', ')', '-', '_', '+', '=', '|', '[', ']', '{',  
                           '}', ';', ':', ',', '.', '/'};  
    char[] numCode = {'q', 'w', 'e', 'r', 't', 'y', 'u', 'i', 'o', 'p'};  
    String src;  
    String result = "";  
    String string = "";  
  
    System.out.print("영어문장 입력 : ");  
    src = keyboard.nextLine();  
}
```

암호표

```
if (src.equals("")) {  
    System.out.println("문자열이 비었습니다");  
} else {  
    src = src.toLowerCase();  
    for (int i = 0; i < src.length(); i++) {  
        char ch = src.charAt(i);  
        if (ch >= 'a' && ch <= 'z') {  
            result += alphabetCode[ch - 'a'];  
        } else if (ch >= '0' && ch <= '9') {  
            result += numCode[ch - '0'];  
        } else if (ch == ' ') {  
            result += ' ';  
        }  
    }  
}
```

암호표

```
for (int i = 0; i < result.length(); i++) {  
    char ch = result.charAt(i);  
    if (ch >= 'a' && ch <= 'z') {  
        string += linearSearch(numCode, ch);  
    } else if (ch == ' ') {  
        string += ' ';  
    } else {  
        string += (char) (linearSearch(alphabetCode, ch) + 'a');  
    }  
}  
System.out.println("원문 : " + src);  
System.out.println("암호 : " + result);  
System.out.println("해독 : " + string);  
}
```

암호문

- 암호 코드 배열 `alphabetCode`에는 문자 'a'부터 시작해서 문자 'z'까지의 암호 코드가 순서대로 저장되어 있기 때문에 문자 'a'의 암호 코드는 `alphabetCode[0]`이고, 문자 'c'의 암호 코드는 `alphabetCode[2]`이다
- 즉, 영어 소문자 `ch`의 암호 코드는 `alphabetCode[ch-'a']`로 표현할 수 있다
- 만일 문자 `ch`가 'c'였다면, 조건식 `'a' <= ch && ch <= 'z'`가 `true`가 되어 `result += alphabetCode[ch-'a'];`가 수행
- 알파벳이나 숫자는 문자코드가 연속적으로 할당되어 있기 때문에, 'c'에서 'a'를 빼면 2를 결과로 얻는다

$$'c' - 'a' \rightarrow 99 - 97 \rightarrow 2$$

중위수(median) 구하기

- 100개의 이하의 정수를 받아들이어 중위수를 구하는 프로그램을 작성하여라.
- 처리 조건
 - num1이 9999이면 처리를 끝낸다.
 - 음수도 처리한다.
 - 가능한 비교회수를 적게 한다.

중위수(median) 구하기

- 중위수(median)는 자료 n 개를 크기 순서로 정리해 놓은 후에 가운데에 위치하는 관측 값을 의미
 - n 이 홀수인 경우 : $(n + 1) / 2$ 번째 값이 중위수
 - n 이 짝수인 경우 : $n / 2$ 번째 순서 값과 $n / 2 + 1$ 번째 순서 값의 평균을 중위수
- 예) 10개의 15, 16, 17, 49, 21, 18, 23, 22, 37, 22
차례로 나열 : 15, 16, 17, 18, 21, 22, 22, 23, 37, 49
중위수 위치 = $(10 + 1) / 2 = 5.5$
따라서 5번째 수와 6번째 수의 평균값
중위수 = $(21 + 22) / 2 = 21.5$

중위수(median) 구하기

- 입 출력 설계
- 출력 데이터
 - 중위수(median) : XX (float 형)
- 입력 데이터
 - 자연수 n개 입력
num[n] (정수 형)

중위수(median) 구하기

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    final int MAX = 100;  
    int count;  
    float median;  
  
    do {  
        System.out.printf(" 몇개의 데이터를 입력 하겠습니까 ?  
                           (최대 %d 개)", MAX);  
  
        count = keyboard.nextInt();  
    } while (!(count < MAX && count > 0));  
    int[] data = new int[count];  
    for (int i = 0; i < data.length; i++) {  
        System.out.printf(" %d / %d 번째 데이터를 입력 : ", i + 1,  
                           data.length);  
  
        data[i] = keyboard.nextInt();  
    }  
}
```


중위수(median) 구하기

```
for (int i = 0; i < data.length - 1; i++)  
    for (int j = i + 1; j < data.length; j++)  
        if (data[i] > data[j]) {  
            int temp = data[j];  
            data[j] = data[i];  
            data[i] = temp;  
        }  
  
if (count % 2 == 0)  
    median = (data[data.length / 2 - 1] +  
              data[data.length / 2 + 1 - 1]) / 2.0f;  
else  
    median = data[data.length / 2];  
  
for (int i = 0; i < data.length; i++)  
    System.out.printf("%3d", data[i]);
```

중위수(median) 구하기

```
if (data.length % 2 == 0)
    System.out.printf("Wn median = %.2f", median);
else
    System.out.printf("Wn median = %.0f", median);
}
```

ISBN Checker

- ISBN(International Standard Book Number)의 마지막 한 자리수의 Check 기호를 계산하는 Program을 작성하여라.



ISBN Checker

- ISBN(International Standard Book Number)의 약자로 국제표준 연속 간행물 번호



ISBN Checker

■ 체크 기호 계산하기(모듈 10 알고리즘)

	접두부			국별 번호		발행자번호						서명 식별 번호	체크 기호	비고
1단계	9	7	8	8	9	9	6	4	3	2	1	0		
	×	×	×	×	×	×	×	×	×	×	×	×		
2단계	1	3	1	3	1	3	1	3	1	3	1	3		
3단계	9	21	8	24	9	27	5	12	3	6	1	0		125
4단계														125÷10 5(나머지)
5단계													5	체크기호 = 10-5
6단계	유효한 ISBN인지 확인 : $125 + 5 = 130 \rightarrow 130 \div 10 = 13 \rightarrow$ 나머지 0													
7단계	978 - 89 - 954321 - 0 - 5													

ISBN Checker

- Check 기호 계산하기
 - 1단계 ISBN
 - 2단계 ISBN 처음 12자리 숫자에 가중치 1과 3을 번갈아가며 기입
 - 3단계 각 가중치를 곱한 후 곱한 값의 합을 계산
 - 4단계 가중치의 합을 10으로 나누어 나머지를 구함
 - 5단계
 - 체크 기호 = $10 - \text{나머지}$
 - 나머지가 0일 경우 Check 기호는 0이다
 - 6단계
 - 처음 12자리에 대한 가중치의 합과 Check 기호를 합쳐서 10으로 나누어 나머지가 없어야만 유효한 ISBN이 됨
 - 7단계 하이픈으로 연결된 완성형을 만듦

ISBN Checker

```
public static void main(String[] args) throws IOException {  
    int[] isbn = new int[13];
```

```
    System.out.print("ISBN(ex.111-11-111111-1-1) = ");  
    for (int i = 0; i < isbn.length; i++) {  
        isbn[i] = System.in.read() - 48;  
    }
```

```
    int hap = 0;  
    for (int i = 0; i < isbn.length - 1; i++) {  
        if (i % 2 == 0) {  
            hap += isbn[i];  
        } else {  
            hap += isbn[i] * 3;  
        }  
    }
```

ISBN Checker

```
int temp = 10 - hap % 10;
int check = temp == 10 ? 0 : temp;

if (check != isbn[12]) {
    System.out.println("ISBN이 올바르지 않습니다.");
} else {
    System.out.println("ISBN이 올바르다");
}
}
```


거스름 돈

- 다음은 거스름돈을 몇 개의 동전으로 지불할 수 있는지를 계산하는 문제로 변수 money의 금액을 동전으로 바꾸었을 때 각각 몇 개의 동전이 필요한지 계산해서 출력하라.
- 단, 가능한 한 적은 수의 동전으로 거슬러 주어야 함
- 500원, 100원, 50원, 10원 5원, 1원
- [Hint] 나눗셈 연산자와 나머지 연산자를 사용

거스름 돈

```
public static void main(String[] args) {
    Scanner keyboard = new Scanner(System.in);
    int[] coinunit = {500, 100, 50, 10, 5, 1};
    int money;
    int count;
    int total = 0;

    System.out.print("\n 거스름 돈을 입력 : ");
    money = keyboard.nextInt();

    for (int i = 0; i < coinunit.length; i++) {
        count = money/coinunit[i];
        System.out.printf(" %8d 원 : %2d개\n", coinunit[i], count);
        money = money % coinunit[i];
        total += count;
    }
    System.out.printf(" 동전 총 개수 : %2d개\n", total);
}
```

거스름 돈(II)

- 거스름에 사용할 동전을 사전에 각 10개씩 보유한 동전의 개수로 거스름돈을 지불할 수 없으면, ‘거스름 돈이 부족합니다.’라고 출력하고 종료한다.
- 지불할 돈이 충분히 있으면, 거스름돈을 지불한 만큼 가진 돈에서 빼고 남은 동전의 개수를 화면에 출력한다.

거스름 돈(II)

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    int[] coinunit = {500, 100, 50, 10, 5, 1}; // 동전의 단위  
    int[] coin = {5, 5, 5, 5, 5, 5}; // 단위별 동전의 개수  
    int money;  
    int total;  
  
    System.out.print("\n 거스름 돈을 입력 : ");  
    money = keyboard.nextInt();  
  
    System.out.println("money = " + money);  
    for (int i = 0; i < coinunit.length; i++) {  
        total = money / coinunit[i];  
        if (coin[i] >= total) {  
            coin[i] -= total;  
        } else {  
            total = coin[i];  
            coin[i] = 0;  
        }  
    }  
}
```

거스름 돈(II)

```
    money -= total * coinunit[i];
    System.out.println(coinunit[i] + "원 : " + total);
}
if (money > 0) {
    System.out.println("거스름돈이 부족합니다.");
    System.exit(0);
}
System.out.println("=남은 동전의 개수 =");
for (int i = 0; i < coinunit.length; i++) {
    System.out.println(coinunit[i] + "원 : " + coin[i]);
}
}
```

막대 그래프 그리기

- 다음과 같이 10점 만점의 5명의 JAVA 성적을 받아 수평 막대 그래프를 그리는 Program을 작성하여라

1번째 학생 JAVA 성적 : 8

2번째 학생 JAVA 성적 : 7

3번째 학생 JAVA 성적 : 5

4번째 학생 JAVA 성적 : 6

5번째 학생 JAVA 성적 : 9

1번 (8개) : ********

2번 (7개) : *******

3번 (5개) : *****

4번 (6개) : *******

5번 (9개) : ***********

막대 그래프 그리기

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    int[] test = new int[5];  
    int i = 0;  
  
    while(i < test.length){  
        System.out.printf(" %d번째 학생 JAVA 성적 : ", i+1);  
        test[i] = keyboard.nextInt();  
        if (test[i] >= 0 && test[i] <= 10)  
            i++;  
        else  
            System.out.println("ERROR : 0 ~ 10점 사이의 값 입력");  
    }  
}
```

막대 그래프 그리기

```
for(i = 0; i < test.length; i++) {  
    System.out.printf("%d번 (%d개) : ", i + 1, test[i]);  
    for(int j = 0; j < test[i]; j++) {  
        System.out.printf("*");  
    }  
    System.out.println();  
}  
}
```


오늘은 무슨 요일인가요 ?

■ 요일 계산

- 1990년 1월 1일 : 월요일부터 시작
- 원하는 날짜를 7로 나누면 나머지가 0 : 일요일 ~ 6 : 토요일로 떨어지게 됨
- 현재시점 : 2018.12.01
- 2017.12.31까지의 일 수
 - $(y - 1) * 365 + \text{윤년의 수}(2월 29일)$
 - 2017년 * 365일 + 윤년의 수(2월 29일) + 올해 전월까지의 날수(31일+28일+31일+30일...11.30까지의 달마다 일수) + 이번 달 일수(1일이라면 +1) = 현재시점까지의 날 수
 - 날수 / 7을 하면 나오는 나머지 활용(날수 % 7 == 0 이면 일요일, 0~6사이의 값)
 - 2018년 12월 1일은 나눴을 때 6이므로 토요일

오늘은 무슨 요일인가요 ?

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int[] days = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    int year, day, month;
    String dayOfWeek = "";
    int totalDay = 0;

    do {
        System.out.print("연도를 입력하세요 (1900 ~ ? ) : ");
        year = scanner.nextInt();
    } while (year < 1900);

    if (((year % 4 == 0) && (year % 100 != 0)) || year % 400 == 0)
        days[1] = 29;

    do {
        System.out.print("월을 입력하세요 (1 ~ 12) : ");
        month = scanner.nextInt();
    } while (month < 1 || month > 12);
```

오늘은 무슨 요일인가요 ?

```
do {
    System.out.printf("일을 입력하세요 (1 ~ %d) : ", days[month-1]);
    day = scanner.nextInt();
} while (day < 1 || day > days[month-1]);

for(int i = 1900; i <= year; i++) {
    if (i < year) {
        if (((i % 4 == 0) && (i % 100 != 0)) || i % 400 == 0)
            totalDay += 366;
        else
            totalDay += 365;
    } else {
        if (month > 1) {
            for (int j = 1; j < month; j++) {
                totalDay += days[j - 1];
            }
        }
    }
}
```

오늘은 무슨 요일인가요 ?

```
totalDay += day;
```

```
switch (totalDay % 7) {  
  case 0:  
    dayOfWeek = "일";  
    break;  
  case 1:  
    dayOfWeek = "월";  
    break;  
  case 2:  
    dayOfWeek = "화";  
    break;  
  case 3:  
    dayOfWeek = "수";  
    break;  
  case 4:  
    dayOfWeek = "목";  
    break;  
}
```



오늘은 무슨 요일인가요 ?

```
case 5:
    dayOfWeek = "금";
    break;
case 6:
    dayOfWeek = "토";
    break;
}
System.out.println(year + "년 " + month + "월 " + day + "일은 " +
                    dayOfWeek + "요일 입니다." );
}
```

현재 날짜 구하기

- System의 날짜 및 시각을 읽을 수 있도록 java.util Package의 Date 클래스와 Calendar 클래스를 제공
- Date는 JDK1.0, Calendar는 JDK1.1부터 제공
- JDK1.8부터 java.time Package로 Calendar와 Date의 단점을 개선한 클래스들이 추가됨
- Calendar는 추상 클래스라서 객체를 직접 생성할 수 없고, 메소드를 통해서 완전히 구현된 클래스의 Instance를 얻어야 함
- Calendar가 추가되면서 Date는 대부분의 메소드가 deprecated 됨
- Calendar 객체를 얻은 후 get() 메소드를 사용해서 날짜와 시간에 대한 정보를 얻을 수 있음

현재 날짜 구하기

- System의 밀리초 구하기(국제표준시각(UTC, GMT) 1970/1/1/0/0/0 으로부터 경과한 시각)

```
long time = System.currentTimeMillis( );  
System.out.println( time.toString( ) );
```

- 밀리초 단위(1000은 1초)
- 음수이면 이전 시각

현재 날짜 구하기

- JAVA의 기본 날짜/시간 Format으로 간단하게 출력

```
Date date = new Date();
```

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

- Date() 생성자는 Computer의 현재 날짜를 읽어 Date 객체로 만듦

현재 날짜 구하기

■ Calendar와 Date 간 변환

//1. Calendar를 Date로 변환

```
Calendar cal = Calendar.getInstance();
```

.....

```
Date date = new Date(cal.getTimeInMillis());
```

//2. Date를 Calendar로 변환

```
Date date = new Date();
```

.....

```
Calendar cal = Calendar.getInstance();
```

```
cal.setTime(date);
```

현재 날짜 구하기

■ 현재 년, 월, 일, 시, 분, 초 구하기

```
Calendar cal = Calendar.getInstance();
```

```
//현재 년도, 월, 일
```

```
int year = cal.get(cal.YEAR);
```

```
int month = cal.get(cal.MONTH) + 1 ;
```

```
int date = cal.get(cal.DATE) ;
```

```
//현재 (시,분,초)
```

```
int hour = cal.get(cal.HOUR_OF_DAY) ;
```

```
int min = cal.get(cal.MINUTE);
```

```
int sec = cal.get(cal.SECOND);
```

■ get(Calendar.MONTH)로 얻어오는 값은 1 ~ 12가 아니라 0 ~ 11임. 그래서 0이 1월 11이 12월을 의미함

현재 날짜 구하기

- 형식에 맞추어 날짜 출력하는 방법
 - SimpleDateFormat을 이용해 형식을 지정

```
SimpleDateFormat mSimpleDateFormat = new  
SimpleDateFormat("yyyy.MM.dd HH:mm:ss", Locale.KOREA );
```

```
Date currentTime = new Date();  
String mTime = mSimpleDateFormat.format(currentTime );  
System.out.println ( mTime );
```