

23.03.08 3강

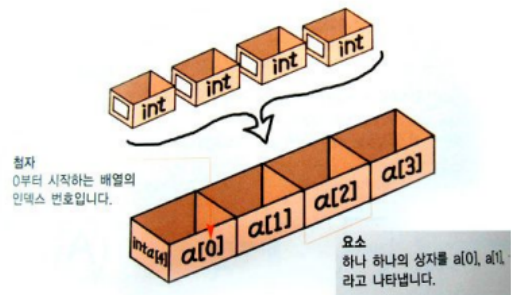
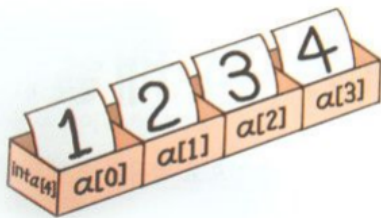
배열

프로그램이 커질 수록 많은 변수를 사용해야 함.
배열은, 변수를 보다 편리하게 사용하기 위한 방법
배열을 사용함으로써 한번에 많은 변수를 처리할 수 있음.

배열 (array)

배열(array)이란 동일 종류의 복수 데이터를 순서대로 저장하는 데이터 구조

배열의 최초의 요소는 0번이다



★ index : 0부터 시작함

배열의 선언

```
// 이렇게 하는 건 비추
int [] numbers;
numbers = new int [5];
// 이게 표준 선언식 자동으로 요소가 모두 0으로 초기화된다
int [] numbers = new int [5];
// 초기화도 가능
int [] numbers = {3,4,9};
```

배열의 길이

```
numbers.length;
```

배열 요소에 값 대입

인덱싱을 통해 값 대입

```
numbers[3] = 60;
```

예외(Exception)

```
package com.example;

import java.util.*;

public class Main {

    public static void main(String[] args) {
        int[] scores = {20, 30, 50, 80, 90};
        int sum = scores[1] + scores[2] + scores[3] + scores[4] + scores[5];
        int avg = sum / 5;
        System.out.println("총점 " + sum);
        System.out.println("평균 " + avg);
    }
}
```

Problems @ Javadoc Declaration Console ×
<terminated> Main [Java Application] C:\Users\#kopo\Desktop\#eclipse\#plugins\#org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933
Exception in thread "main" [java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5](#)
at com.example.Main.main([Main.java:9](#))

배열에서 가장 많이 실수하는 범위 오류 - 인덱스에 주의하자

```
package com.example;

import java.util.*;

public class Main {

    public static void main(String[] args) {
        int[] scores = {20, 30, 50, 80, 90};
        int sum = scores[0] + scores[1] + scores[2] + scores[3] + scores[4];
        int avg = sum / 5;
        System.out.println("총점 " + sum);
        System.out.println("평균 " + avg);
    }
}
```

자료형에 주의하자

```
package com.example;

import java.util.*;

public class Main {
```

```

public static void main(String[] args) {
    int[] scores = {20, 30, 50, 83, 90};
    int sum = scores[0] + scores[1] + scores[2] + scores[3] + scores[4];
    double avg = sum / 5;
    System.out.println("총점 " + sum);
    System.out.println("평균 " + avg);
}
}

```

```

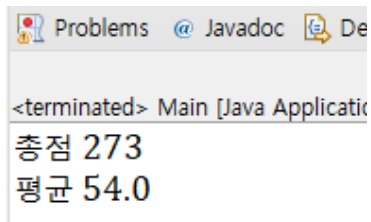
package com.example;

import java.util.*;

public class Main {

    public static void main(String[] args) {
        int[] scores = {20, 30, 50, 83, 90};
        int sum = scores[0] + scores[1] + scores[2] + scores[3] + scores[4];
        double avg = sum / 5.0;
        System.out.println("총점 " + sum);
        System.out.println("평균 " + avg);
    }
}

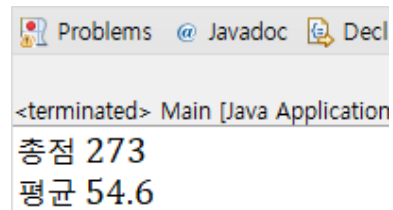
```



```

<terminated> Main [Java Application]
총점 273
평균 54.0

```



```

<terminated> Main [Java Application]
총점 273
평균 54.6

```

for문과 length, foreach를 적극 사용하자

```

package com.example;

import java.util.*;

public class Main {

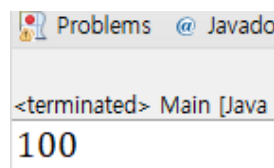
    public static void main(String[] args) {
        int[] scores = {20, 30, 50, 83, 90};
        int sum = 0;
        // for (int i = 0; i < scores.length; i++) {
        //     sum += scores[i];
        // }
        for (int value : scores) {
            sum += value;
        }
        double avg = sum / (double) scores.length;
        System.out.println("총점 " + sum);
        System.out.println("평균 " + avg);
    }
}

```

```
}  
}
```

주의할 것 -메모리와 변수

```
package com.example;  
  
import java.util.*;  
  
public class Main {  
  
    public static void main(String[] args) {  
        int[] scores = {20, 30, 50, 83, 90};  
        int[] sum = scores;  
        // for (int i = 0; i < scores.length; i++) {  
        //     sum += scores[i];  
        // }  
        sum[0] = 100;  
        System.out.println(scores[0]);  
    }  
}
```



변수를 $b = a$ 와 같이 선언하면, 같은 배열을 복사하는 것이 아님 원본 배열의 메모리를 참조하므로, 하나를 수정하면 다른 것도 수정되는 문제가 발생한다.

참조

배열처럼 변수명을 지정 했을 때, 그 값이 아니라 주소를 가리키는 것을 참조 (reference)라고 한다.

그리고 그 변수를 참조형 (reference type) 변수라고 한다.

int 나 boolean 같은 “기본형”(primitive type) 변수와 구별 된다.

가비지 컬렉션

new로 확보된 요소들은 보통의 변수와 다르기 때문에, 블록이 끝나도 수명이 다하지 않는다

블록 내에서 생성된 배열은 이 후 어떤 방법으로도 읽거나 쓸 수 없고 메모리를 차지하여 메모리 내의 쓰레기(garbage)가 된다.

```

1 public class Main {
2     public static void main(String[] args) {
3         boolean b = true;
4         if (b == true) {
5             int[] i = { 1, 2, 3 };
6         }
7     }
8 }

```

위 그림에서 i는 garbage라고 할 수 있다.

Java는 가비지 컬렉션 (GC, garbage collection) 이라는 장치가 더 이상 사용되지 않는 메모리를 정리 해 준다.

null

참조형 변수에 대입하면, 이 변수는 아무것도 참조하지 않게 됨
참조를 끊어낼 때 사용

```

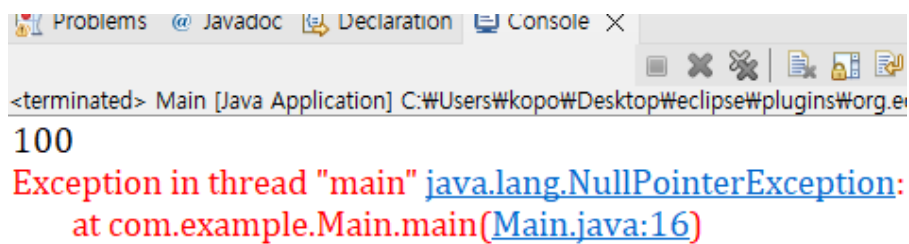
package com.example;

import java.util.*;

public class Main {

    public static void main(String[] args) {
        int[] scores = {20, 30, 50, 83, 90};
        int[] sum = scores;
        scores = null;
        // for (int i = 0; i < scores.length; i++) {
        //     sum += scores[i];
        // }
        sum[0] = 100;
        System.out.println(sum[0]);
        System.out.println(scores[0]);
    }
}

```



The screenshot shows the Eclipse IDE's console window. It displays the output of a Java application. The first line is "100", which is the value of sum[0] after it was set to 100. The second line is an exception message: "Exception in thread 'main' java.lang.NullPointerException: at com.example.Main.main(Main.java:16)". This indicates that the program crashed because it tried to access an array element (scores[0]) that had been set to null.

java.lang.NullPointerException ⇒ 잘못된 참조

문자열의 길이 length()

String의 length() 는 한글, 영문, 공백 관계없이 1문자로 카운트
배열의 length 와 비슷하지만 () 를 붙여야 한다

```
1 public class Main {  
2     public static void main(String[] args) {  
3         String s = "Java로 개발";  
4         System.out.println(s.length());  
5     }  
6 }
```

다차원 배열

2차원 배열

```
int[][] scores = new int[2][3];  
scores[0][0] = 30;  
  
int[][] scores = {{10, 20, 30}, {40, 50, 60}};  
  
System.out.println(scores.length);      // 2  
System.out.println(scores[0].length);    // 3
```

연습문제

연습문제 4-1

다음 조건에 맞는 각 배열을 준비하는 프로그램을 작성하시오. 값의 초기화는 필요 없음.

1. int형 값을 4개 담을 수 있는 배열 points
2. double 형 값을 5개 담을 수 있는 배열 weights
3. boolean 형 값을 3개 담을 수 있는 배열 answers
4. String형 값을 3개 담을 수 있는 배열 names

```
int [] points = new int [4];  
double [] weights = new double [5];  
boolean [] answers = new boolean [3];  
String [] names = new String [3];
```

연습문제 4-2

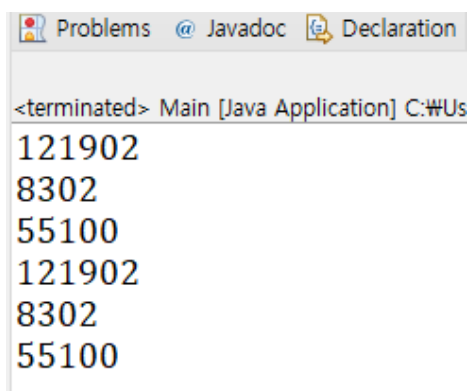
다음 조건에 맞는 프로그램을 작성하시오.

1. 3개의 계좌 잔액 "121902", "8302", "55100" 이 담겨 있는 int 형 배열 moneyList 를 선언하시오
2. 그 배열의 요소를 1개씩 for 문으로 꺼내서 화면에 표시하시오
3. 같은 배열 요소를 foreach 문으로 1개씩 꺼내서 화면에 표시하시오

```
package com.example;

public class Main {

    public static void main(String[] args) {
        // 1. 3개의 계좌 잔액 "121902", "8302", "55100" 이 담겨 있는 int 형 배열 moneyList 를 선언하시오
        int[] moneyList = {121902, 8302, 55100};
        // 2. 그 배열의 요소를 1개씩 for 문으로 꺼내서 화면에 표시하시오
        for (int i = 0; i < moneyList.length; i++) {
            System.out.println(moneyList[i]);
        }
        // 3. 같은 배열 요소를 foreach 문으로 1개씩 꺼내서 화면에 표시하시오
        for (int money : moneyList) {
            System.out.println(money);
        }
    }
}
```



```
<terminated> Main [Java Application] C:\WUs
121902
8302
55100
121902
8302
55100
```

foreach : 파이썬의 for 문에서 리스트의 요소 반복과 유사하다.

```

1 // 각 라인에서 발생하는 예외의 이름을 답하시오
2 public class Main {
3     public static void main(String[] args) {
4         int[] counts = null;
5         float[] heights = { 171.3F, 175.0F };
6         System.out.println(counts[1]); // 예외 발생
7         System.out.println(heights[2]); // 예외 발생
8     }
9 }

```

```

package com.example;

public class Main {

    public static void main(String[] args) {
        int[] counts = null;
        float[] heights = {171.3f, 175.0f};
        System.out.println(counts[1]);
        System.out.println(heights[2]);
    }
}

```

Line 8 : java.lang.NullPointerException

null로 인해 참조가 끊어져 잘못된 참조로 인해 오류가 발생함

Line 9 : java.lang.ArrayIndexOutOfBoundsException

배열의 2번째 요소에 접근하려면 최소 3개의 요소가 필요하나, 배열의 길이가 2이다. 파이썬의 IndexError : out of range와 유사하다.

연습문제 4-4

다음 4개의 조건에 맞는 “숫자 맞추기 퀴즈” 프로그램을 작성 하시오.

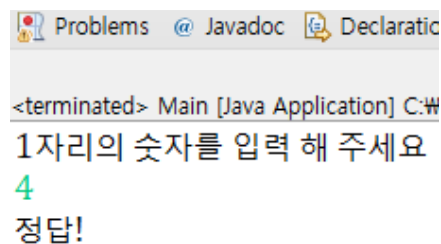
1. 3개짜리 int형 배열 numbers를 준비하시오. 이 때 초기화는 각각 3, 4, 9 로 합니다.
2. 화면에 “1자리의 숫자를 입력 해 주세요” 라고 표시합니다
3. `int input = new java.util.Scanner(System.in).nextInt();` 대입합니다
4. input값이 3, 4, 9 중 하나와 같다면 “정답!” 이라고 표시합니다.

```
package com.example;

import java.util.*;

public class Main {
    public static boolean contains(final int[] array, final int key) {
        for (final int i : array) {
            if (i == key) {
                return true;
            }
        }
        return false;
    }

    public static void main(String[] args) {
        // 1. 3개짜리 int형 배열 numbers를 준비하시오. 이 때 초기화는 각각 3, 4, 9 로 합니다.
        int[] numbers = {3, 4, 9};
        // 2. 화면에 “1자리의 숫자를 입력 해 주세요” 라고 표시합니다
        System.out.println("1자리의 숫자를 입력 해 주세요");
        // 3. 다음 코드를 사용해 키보드로부터 숫자를 입력 받아, 변수 input에 대입합니다
        int input = new Scanner(System.in).nextInt();
        // 4. input값이 3, 4, 9 중 하나와 같다면 “정답!” 이라고 표시합니다
        if (contains(numbers, input)) {
            System.out.println("정답!");
        }
    }
}
```



```
<terminated> Main [Java Application] C:\
1자리의 숫자를 입력 해 주세요
4
정답!
```