

JAVA 배열 (3일차)

3장 피드백 Tip

 String 덧셈 연산은 지양하는 것이 좋다.

```
String result = "";

for (int i = 0; i < 3; i++) {
    result = result + (i + ","); // "0,"
                                // "0, 1,"
                                // "0, 1, 2,"
}

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



String에 더하기 연산을 할 경우, 각각 더해진 만큼의 불필요한 오버헤드가 발생한다.

- `String`은 `final`로 구현되어있음.
- 최종 결과만 `result`에 대입 된다.

위 코드의 결과는 정상적으로 출력 되지만, 더해진 모든 경우에 대해 서로 다른 String 객체가 메모리 상에 각각 생성되어 overhead가 발생한다.

따라서, 일반 String 객체에 값을 더하는 방법이 아닌, **StringBuilder**를 사용하여 **append()**를 사용하는 것이 좋다.

nextLine() 사용 문제

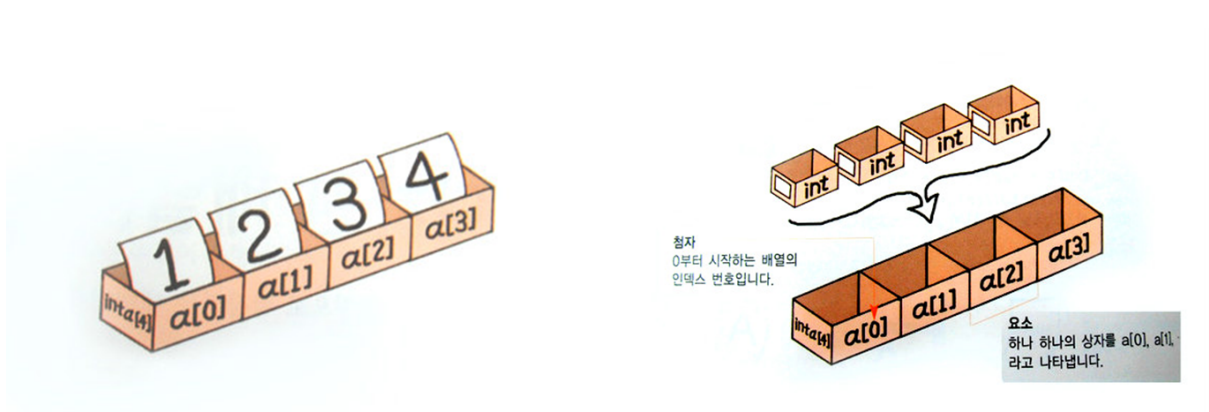
```
int input = sc.nextInt();
sc.nextLine(); // 입력버퍼를 비워주는 역할을 한다.
System.out.println("이름>");
name = sc.nextLine();
```

nextLine() 특징

- 한 줄 단위로 입력한다.
- 처음 Enter 입력의 원하지 않는 개행 문자 까지 읽는다.
- 따라서 한번 더 `nextLine()`을 앞전에 사용함으로써 개행 문자를 제거할 수 있다.

4장 배열

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



Java 배열(Array) 작성

```
int [] scores;          // 배열변수의 선언
scores = new int[5];    // 요소의 작성과 대입
```

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

배열 길이 구하기

```
int[] scores = new int[5];
int count = scores.length;
System.out.println("배열의 길이: " + count);
```

배열 요소 값 대입

```
int[] scores = new int[5];
for (int i = 0; i < scores.length; i++){
    scores[i] = i; // 배열 값 대입
}
```

```
int [] scores1 = new int[] {10, 20, 30, 40, 50};
int [] scores2 = {1, 2, 3, 4, 5};
```

배열의 초기화

```
1 public class Main {
2     public static void main(String[] args) {
3         // 변수를 사용하기 전에는 반드시 초기화를 해야 한다
4         int x;
5         System.out.println(x); // 컴파일 에러
6     }
7 }
```

```
1 public class Main {
2     public static void main(String[] args) {
3         // 배열의 요소는 자동으로 초기화된다
4         // 5개의 요소가 전부 0으로 초기화
5         int[] score = new int[5];
6         System.out.println(score[0]); // 에러가 아님
7     }
8 }
```



배열은 선언과 동시에 모든 요소가 0로 초기화 된다.

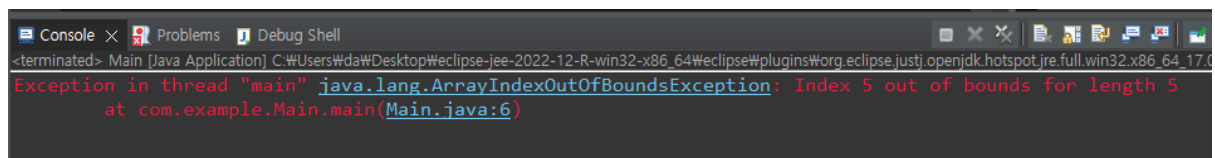
IndexOutOfBoundsException

- 배열 범위 벗어난 요소 이용할 때 **예외(exception)** 발생

```
package com.example;

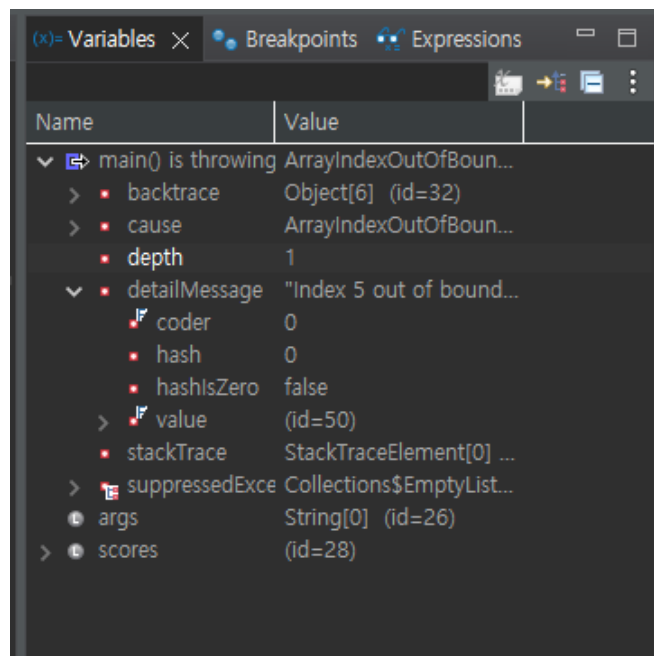
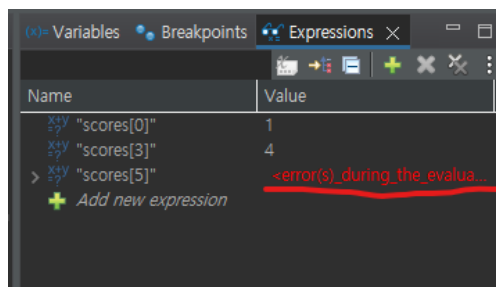
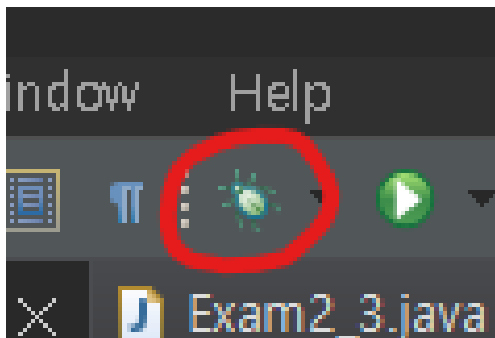
public class Main {
    public static void main(String[] args) {
        int[] scores = {1, 2, 3, 4, 5};
        // index 5 범위를 벗어남
        int sum = scores[0] + scores[1] + scores[2] + scores[3] + scores[5];
        double average = sum / 5;

        System.out.println(sum);
        System.out.println(average);
    }
}
```



→ console에 표시된 예외를 보고 디버깅 할 수 있다.

Debug Tool 사용



디버깅 모드에서, 오류가 발생한 것 같은 지점에 **break** 포인트를 적절히 사용하여 버그를 찾는다.

- **F6** 한줄씩 실행
- **F8** 전체 실행

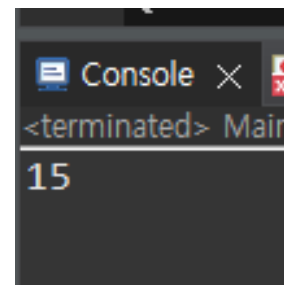
for each 문 (확장 for 문)

```
7      // 일반 for 문
8      for (int i = 0; i < scores.length; i++) {
9          sum += scores[i];
10     }
11     // for - each
12     sum = 0;
13     for (int score : scores) {
14         sum += score;
15     }
```

```
package com.example;

public class Main {
    public static void main(String[] args) {
        int[] scores = {1, 2, 3, 4, 5};
        int sum = 0;

        for (int score : scores) { // for - each
            sum += score;
        }
        System.out.println(sum);
    }
}
```



```
Console X
<terminated> Main
15
```

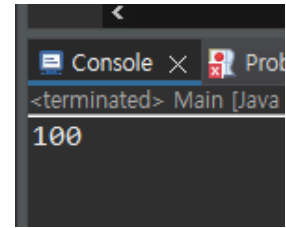
- 배열 요소에 대해 반복을 쉽게 할 수 있다.

메모리와 변수

```
// 배열 변수 a는 주소(reference)가 저장됨
```

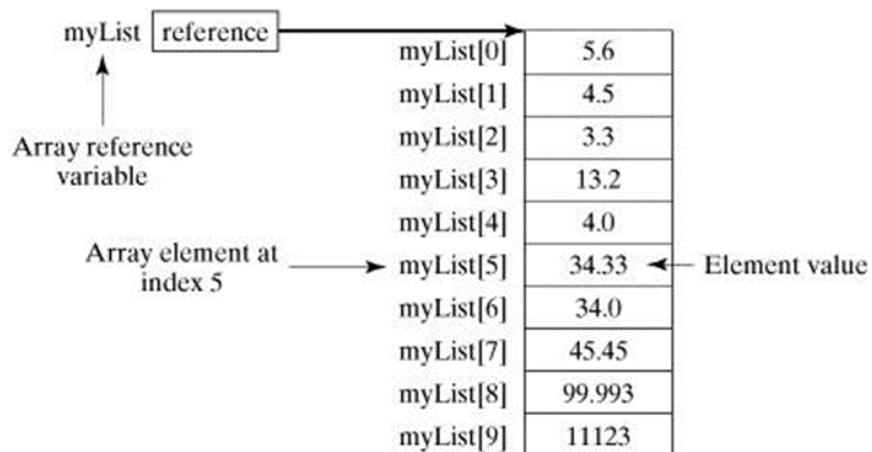
```
int[] a = { 1, 2, 3 };
int[] b;

b = a; // a 배열의 시작 주소가 b에 저장됨
b[0] = 100;
System.out.println(a[0]);
// b[0]에 100을 넣었으나, a[0]도 100이 출력됨.
```



배열 변수에는 각 요소가 들어 있는 것이 아닌, 최초의 요소 주소(reference)가 대입 된다.

참조 (reference)



참조(reference) : 변수 명 지정 시, 그 값이 아닌 주소를 가리키는 것



참조 형(reference type) 변수 : 주소를 가리키는 변수

그 외 int, double, boolean 등의 **기본형(primitive type)**변수와 구별 된다.

가비지 컬렉션 (garbage collection)

일반적으로 **함수 호출과 관련된 지역, 매개변수는 메모리의 stack 영역에 저장**된다. 또한 stack 영역은 함수 호출과 함께 할당되고, 함수 호출이 종료되면 pop되어 소멸한다.

하지만, **new 로 사용자에게 의해 동적으로 생성된 객체 요소들은 메모리의 heap 영역에 저장**된다.



따라서, **new** 로 생성된 객체 요소들은 해당 블록이 끝나도 수명이 다하지 않고, 메모리 상에 존재한다.



이렇게 사용하지 않게 된 메모리는 **Java에서는 가비지 컬렉션 (GC, Garbage Collection)에 의해 더 이상 사용되지 않는 메모리를 정리해 준다.**

▼ ChatGPT 설명

Java의 GC는 JVM (Java Virtual Machine)에 의해 수행됩니다. JVM은 실행 중인 Java 애플리케이션의 메모리 사용을 지속적으로 모니터링하며, 사용되지 않은 메모리를 식별하여 자동으로 해제합니다. 이러한 과정에서, Java 애플리케이션의 성능이 향상되며, 개발자는 메모리 관리에 더 이상 신경쓰지 않아도 됩니다.



java GC에 대해 알려줘



Java에서의 Garbage Collection (GC)은 동적 메모리 할당 및 해제를 자동으로 처리하는 기술입니다. Java는 개발자가 직접 메모리 할당 및 해제를 관리할 필요가 없도록 하며, 메모리 누수나 잘못된 메모리 관리로 인한 오류를 방지합니다.

Java의 GC는 JVM (Java Virtual Machine)에 의해 수행됩니다. JVM은 실행 중인 Java 애플리케이션의 메모리 사용을 지속적으로 모니터링하며, 사용되지 않는 메모리를 식별하여 자동으로 해제합니다. 이러한 과정에서, Java 애플리케이션의 성능이 향상되며, 개발자는 메모리 관리에 더 이상 신경쓰지 않아도 됩니다.

Java의 GC는 크게 세 가지 유형으로 구분됩니다.

1. Mark and Sweep GC

- 모든 객체를 순회하며 사용되는 객체와 사용되지 않는 객체를 식별하고, 사용되지 않는 객체를 삭제합니다.

2. Copying GC

- 힙을 두 개의 영역으로 나누고, 한 영역에서 사용되는 객체를 다른 영역으로 복사하며 사용되지 않는 객체를 삭제합니다.



메모리 구조

코딩교육 티씨피스쿨

4차산업혁명, 코딩교육, 소프트웨어교육, 코딩기초, SW코딩, 기초코딩부터 자바 파이썬 등

TCPSchool

T http://www.tcpschool.com/c/c_memory_structure


```
package com.example;

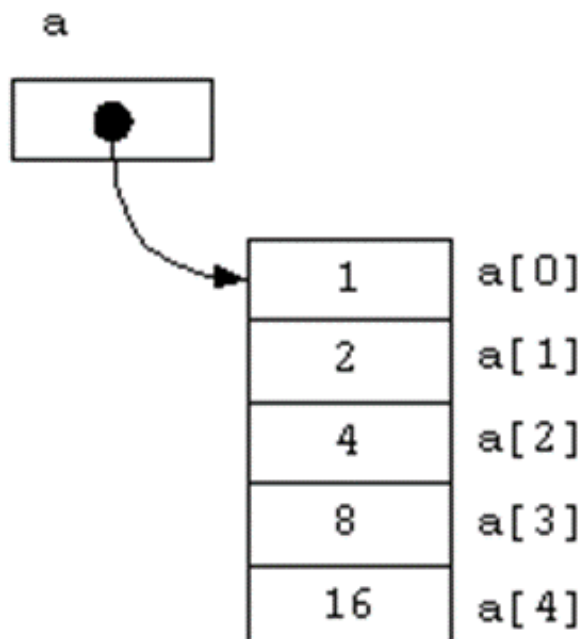
public class Main {
    public static void main(String[] args) {
        boolean b = true;
        if(b == true) {
            int[] nums = new int[] { 1, 2, 3 };
        }
        System.out.println(nums[1]); // 컴파일 에러
    }
}
```

- “nums” 변수는 블록 내부 선언된 변수이므로 사용 불가하다.
- 하지만 new로서 생성된 { 1, 2, 3 } 값은 메모리 내에 존재한다.
- GC에 의해 동정 메모리 해제가 자동으로 된다.



C 에서 프로그래머가 직접 malloc() 동적메모리 할당과 free() 메모리 정리를 Java에서는 자동으로 정리해주는 것이다.

null



null의 참조 제거 기능



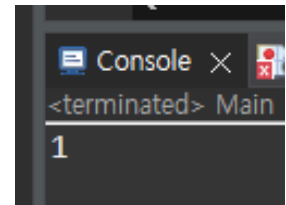
참조형 변수(reference type)에 null을 대입하여 아무 것도 참조하지 않게 할 때 사용한다.



기본형 변수(primitive type)에는 null을 대입할 수 없다.

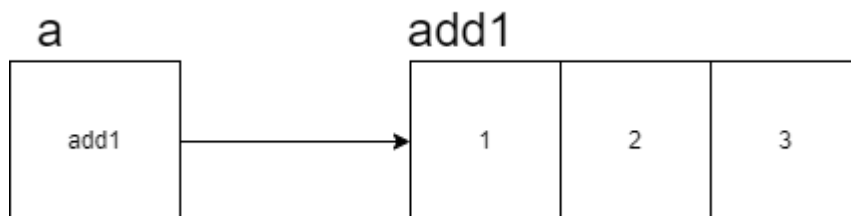
100 참조형 변수의 이해

```
public class Main {  
    public static void main(String[] args) {  
        int[] a = { 1, 2, 3 };  
        int[] b;  
        b = a;  
        a = null;  
        System.out.println(b[0]);  
    }  
}
```

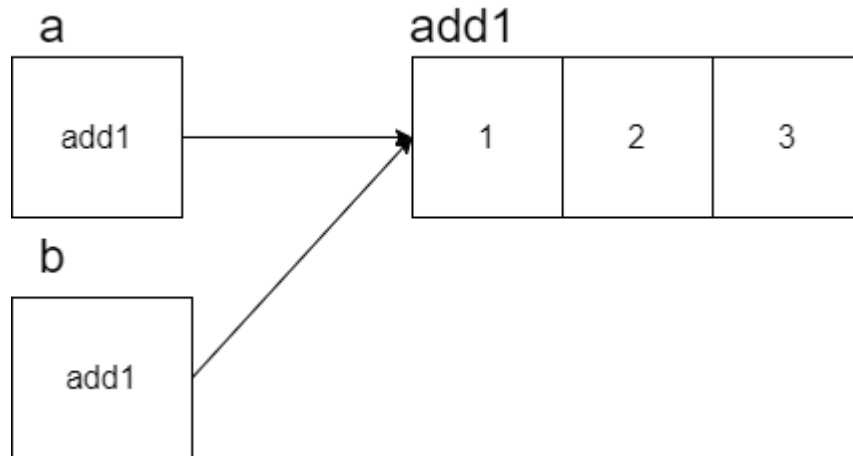


- 배열 b에 배열 a 참조형 변수가 가리키는 주소를 넣는다.
- 이후 a를 null로 초기화 하고 b를 출력하면, 원래 a가 가리키던 값이 출력된다.

위 코드에 대한 그림 설명

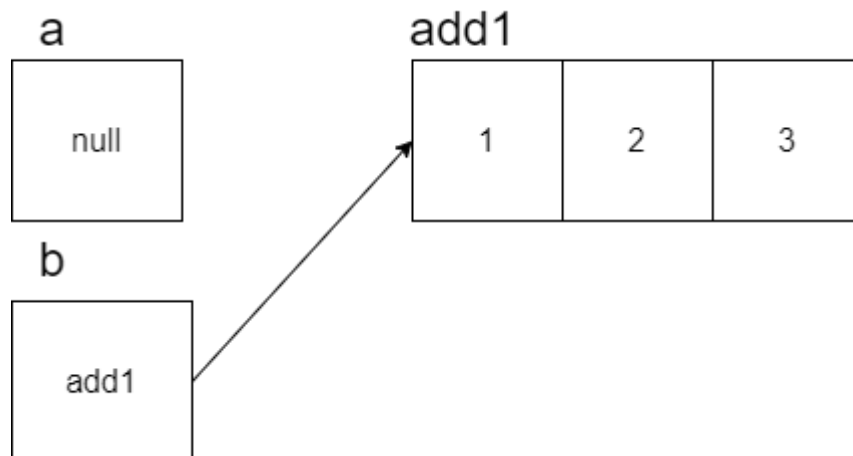


```
int[] a = { 1, 2, 3 };
```



💡 `int[] b;`

💡 `b = a;`



💡 `a = null;`

💡 a의 참조 연결을 끊어도, b는 연결되어 있으므로 b 출력 시 정상 출력된다.

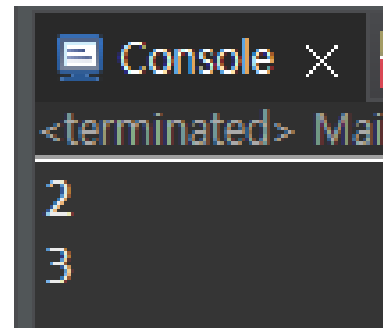
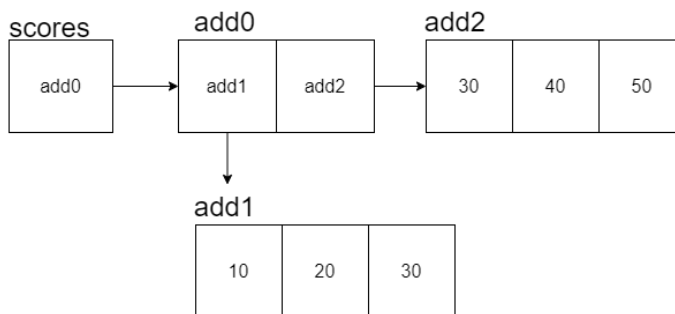
다차원 배열

| | | |
|----------------|----------------|----------------|
| arr[0][0] 1 | arr[0][1] 2 | arr[0][2] 3 |
| arr[1][0] 4 | arr[1][1] 5 | arr[1][2] 6 |
| arr[2][0] 7 | arr[2][1] 8 | arr[2][2] 9 |

```
package com.example;

public class Main {
    public static void main(String[] args) {
        int[][] scores = {{10, 20, 30}, {30, 40, 50}};
        System.out.println(scores.length);
        System.out.println(scores[0].length);
    }
}
```

- 다차원 배열 구조



scores는 {10, 20, 30}, {30, 40, 50}이 각각 저장된 주소를 가리키는 add0을 가리키고 있으므로, **scores의 length는 2이다.**



또한 scores[0]은 add1을 가리키고 있으며, add1은 10, 20, 30의 3개의 원소를 가지고 있으므로, **scores의 length는 3이다.**

연습 문제

4-1

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

- int 형 값을 4개 담을 수 있는 배열 points
- double 형 값을 5개 담을 수 있는 배열 weights
- boolean 형 값을 3개 담을 수 있는 배열 answers
- String 형 값을 3개 담을 수 있는 배열 names

```
package com.example;

public class Exam4_1 {
    public static void main(String[] args) {
        int[] points = new int[4];
        double[] weights = new double[5];
        boolean[] answers = new boolean[3];
        String[] names = new String[3];
    }
}
```

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 Exam4_2 {
    public static void main(String[] args) {
        int[] moneyList = {121902, 8302, 55100};

        // 일반 for문을 통한 배열 요소 가져오기
        for (int i = 0; i < moneyList.length; i++) {
            System.out.println(moneyList[i]);
        }
        // for - each 문을 통한 배열 요소 가져오기
        for (int money : moneyList) {
            System.out.println(money);
        }
    }
}
```

4-3

```
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 }
```

답

```
Exception in thread "main" java.lang.NullPointerException: Cannot load from int array because "counts" is null
at com.example.Exam4_3.main(Exam4_3.java:8)
```

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 2 out of bounds for length 2
at com.example.Exam4_3.main(Exam4_3.java:9)
```

- 예외 1: java.lang.NullPointerException
- 예외 2: java.lang.ArrayIndexOutOfBoundsException

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. 다음 코드를 사용해 키보드로부터 숫자를 입력 받아, 변수 input 에 대입합니다
input값이 3, 4, 9 중 하나와 같다면 “정답!” 이라고 표시합니다.

```
int input = new java.util.Scanner(System.in).nextInt();
```

4. input값이 3, 4, 9 중 하나와 같다면 “정답!” 이라고 표시합니다.

```
package com.example;

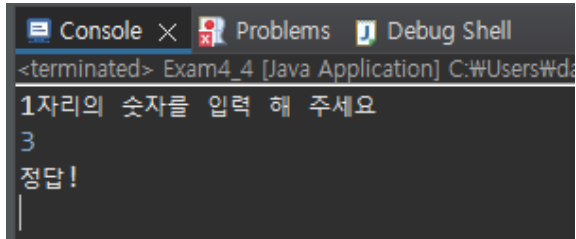
public class Exam4_4 {
    public static void main(String[] args) {
        int[] numbers = { 3, 4, 9 };
    }
}
```

```

System.out.println("1자리의 숫자를 입력 해 주세요");
int input = new java.util.Scanner(System.in).nextInt();

if (input == 3 || input == 4 || input == 9) {
    System.out.println("정답!");
}
}
}

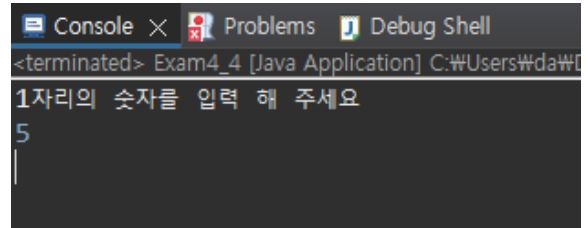
```



```

<terminated> Exam4_4 [Java Application] C:\Users\da\Wd
1자리의 숫자를 입력 해 주세요
3
정답!
|

```



```

<terminated> Exam4_4 [Java Application] C:\Users\da\Wd
1자리의 숫자를 입력 해 주세요
5
|

```

퀴즈

★ 문자열 결합 - 배열로 풀기

입력 받을 문자열 수와 문자열을 입력받고 출력 포맷에 맞게 출력하세요.

입력 포맷

다음 포맷으로 입력

```

n // 입력 받을 문자열 수
s_1 // 문자열
s_2
..
s_n

```

출력 포맷

```

Hello s_1,s_2,..s_n.

```

입력 예1

```

2
Java
Gino

```

출력 예1

```

Hello Java,Gino.

```

```

import java.util.Scanner;

```

```

import java.util.Arrays;

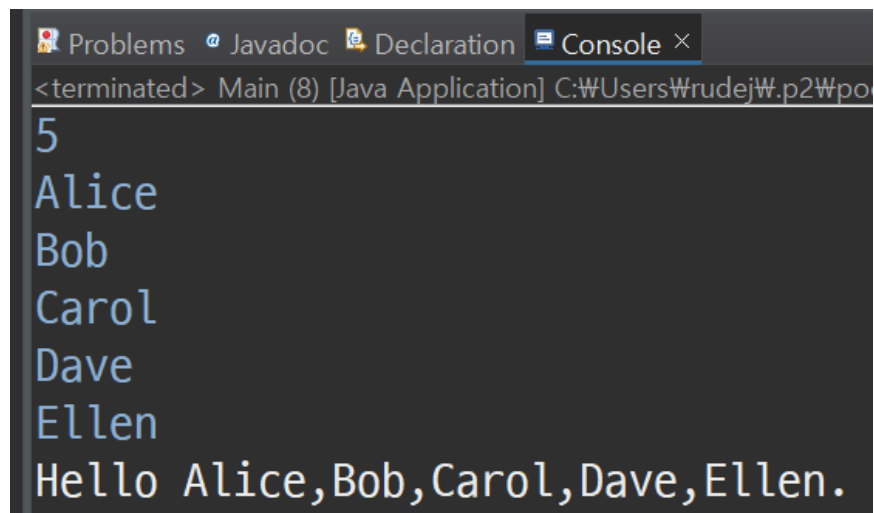
public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        String[] names = new String[n + 1];

        names[0] = "Hello ";
        for (int i = 1; i < n + 1; i++) {
            String input = sc.next();

            if (i == n) {
                names[i] = input + ".";
            } else {
                names[i] = input + ",";
            }
        }
        for (String name : names) {
            System.out.print(name);
        }
    }
}

```



```

Problems Javadoc Declaration Console ×
<terminated> Main (8) [Java Application] C:\Users\Wrudej\p2\Wpo
5
Alice
Bob
Carol
Dave
Ellen
Hello Alice,Bob,Carol,Dave,Ellen.

```



퀴즈01을 배열을 이용해서 해결했다. $n+1$ 의 크기로 배열을 초기화 한 후 Hello를 포함한 추후 입력 값들을 처리하여 배열에 넣은 후, 배열의 전체 원소들을 출력하는 방식으로 문제를 해결했다.