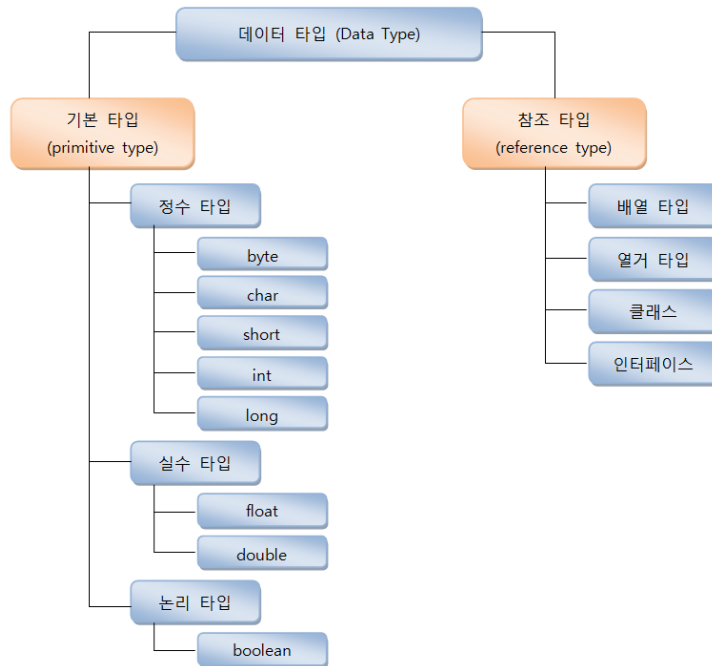




배열과 참조타입

데이터 타입 분류



변수의 메모리 사용

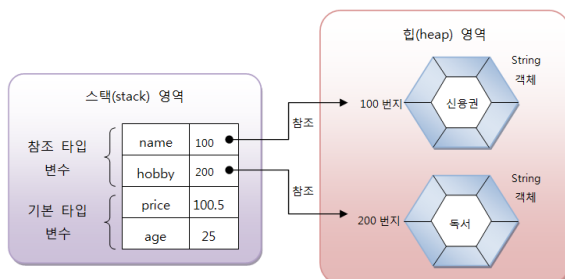
- 기본 타입 변수 : 실제 값을 변수 안에 저장
- 참조 타입 변수 : 주소를 통해 객체 참조

[기본 타입 변수]

```
int age = 25;  
double price = 100.5;
```

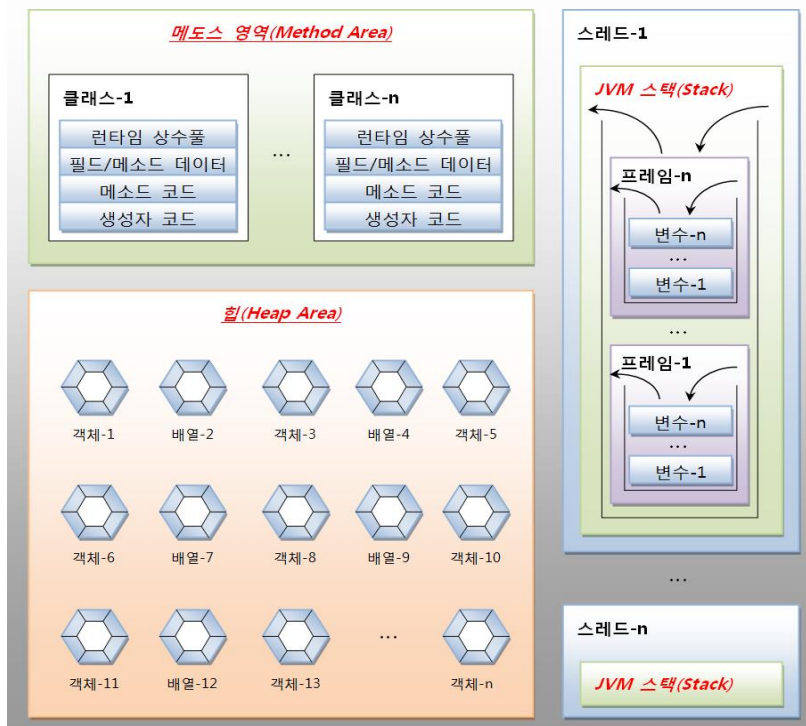
[참조 타입 변수]

```
String name = "신용권"  
String hobby = "독서";
```

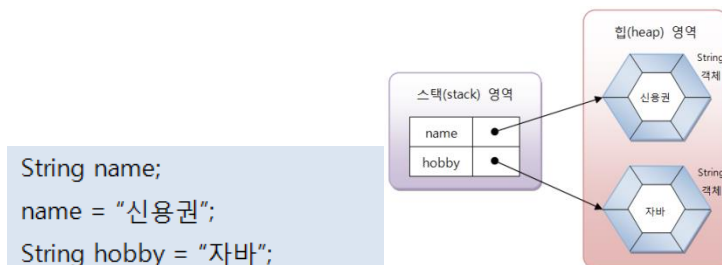


JVM이 사용하는 메모리 영역

Runtime Data Area



String 타입



배열이란?

같은 타입의 데이터를 연속된 공간에 저장하는 자료구조

배열 생성하는 방법

- 변수 선언과 동시에 초기화

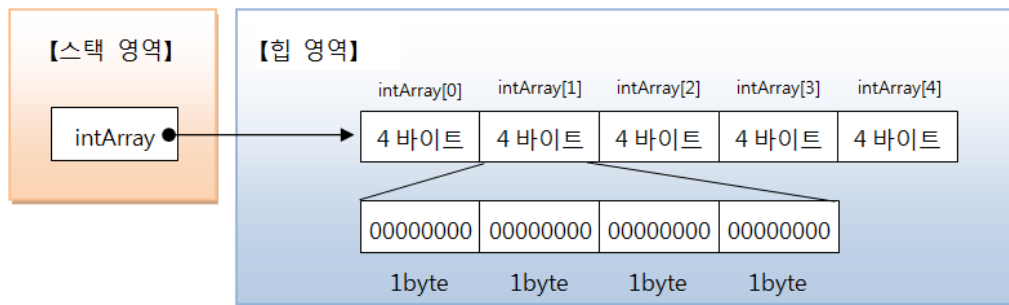
```
데이터타입[] 변수 = { 값 0, 값 1, 값 2, 값 3, ... };
```

- 변수 선언 후 값 목록 대입

```
데이터타입[] 변수;  
변수 = new 타입[] { 값 0, 값 1, 값 2, 값 3, ... };
```

new 연산자로 배열 생성

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



배열의 길이는 자동으로 구해준다

```
int[] scores = { 83, 90, 87 };

int sum = 0;
for(int i=0; i<scores.length; i++) {
    sum += scores[i];
}
System.out.println("총합 : " + sum);
```

아래코드 실습해 보세요

```
class StringArray {
    public static void main(String[] args) {
        String[] strArr = new String[3];
        strArr[0] = "사과";
        strArr[1] = "포도";
        strArr[2] = "배";

        for(int i = 0; i < strArr.length; i++)
            System.out.println("strArr[" + i + "]: " + strArr[i]);
    }
}
```

```
strArr[0]: 사과
strArr[1]: 포도
strArr[2]: 배
```

실습예제1

키보드에서 입력받은 양수 5개를 배열에 저장하고 제일 큰 수를 출력하는 프로그램을 코딩하시오.

```
import java.util.Scanner;
public class ArrayAccess {
    public static void main (String[] args) {
        Scanner in = new Scanner(System.in);
```

```

int intArray[] = new int[5]; // 배열의 선언과 생성
int max = 0;

for (int i = 0; i < 5; i++) {
    intArray[i] = in.nextInt(); // 키보드에서 입력받은 정수를 배열에 저장
    if (intArray[i] > max)      // 가장 큰 수와 비교하여 크면
        max = intArray[i]; // max에 저장
}
System.out.print("입력된 수에서 가장 큰 수는 " + max + "입니다.");
}
}

```

1
40
56
3
98
입력된 수에서 가장 큰 수는 98입니다.

실습예제2

5개의 정수를 입력받고 평균을 구하는 프로그램을 작성하시오

```

import java.util.Scanner;
public class ArrayLength {
    public static void main (String[] args) {
        Scanner in = new Scanner(System.in);
        int intArray[] = new int[5]; // 배열의 선언과 생성
        double sum = 0;
        for (int i = 0; i < intArray.length; i++)
            intArray[i] = in.nextInt(); // 키보드에서 입력받은 정수 저장
        for (int i = 0; i < intArray.length; i++) {
            sum += intArray[i]; // 배열에 저장된 정수 값을 더하기
        }
        System.out.print("배열 원소의 평균은 " + sum/intArray.length + "입니다.");
    }
}

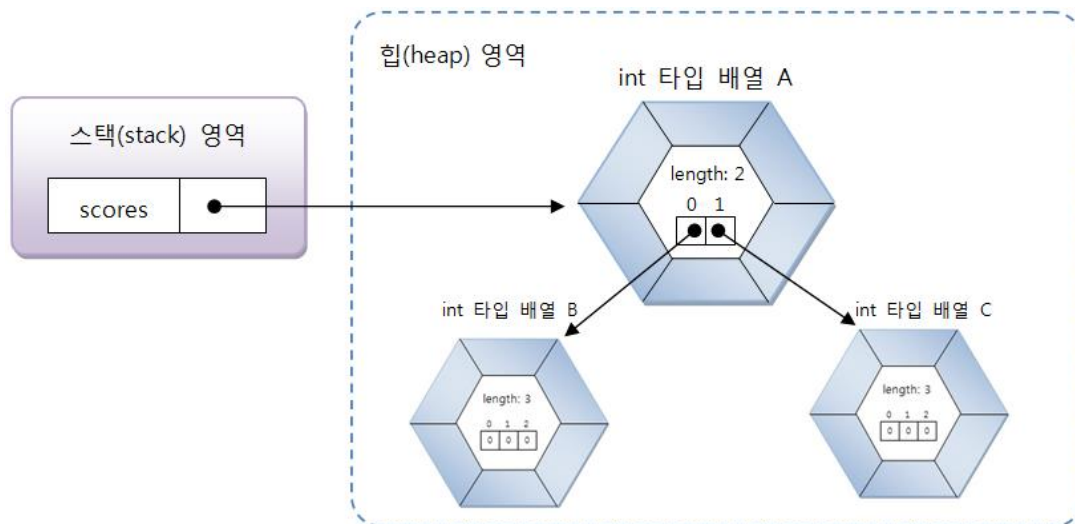
```

10 20 30 40 50
배열 원소의 평균은 30.0입니다.

다차원 배열

자바는 1차원 배열을 이용해 2차원 배열 구현

```
int[][] scores = new int[2][3];
```



```
class TwoDimenArray {
    public static void main(String[] args) {
        int[][] grades = {{55, 60, 65}, {85, 90, 95}};
        int high_test;

        high_test = getHighest(grades, 2, 3);

        System.out.println("가장 높은 점수는 " + high_test + "점 입니다.");
    }

    public static int getHighest(int[][] a, int row, int col)
    {
        int highest = a[0][0];

        for(int i=0; i<row; i++)
            for(int j=0; j<col; j++)
                if (a[i][j]>highest)
                    highest = a[i][j];

        return highest;
    }
}
```

결과:

가장 높은 점수는 95점 입니다.

코드로 돌아가, 3행을 보시면 2차원 배열 grades가 선언되어 있음을 확인할 수 있습니다. 선언과 동시에 초기화를 하고있죠. 이 경우에는 행이 3, 열이 2인 2차원 배열이 만들어집니다. 그림으로 표현하자면,

55 array[0][0]	60 array[0][1]	65 array[0][2]
85 array[1][0]	90 array[1][1]	95 array[1][2]

<배열 grades>

실습예제3

한 회사의 지난 3년간 분기별 매출의 총액과 연평균 매출을 구하는 프로그램을 코딩하시오.

```
public class SalesRevenue {
    public static void main (String[] args) {
        int intArray[][] = {{90, 90, 110, 110}, // 1차년도의 매출
                            {120, 110, 100, 110}, // 2차년도 매출
                            {120, 140, 130, 150}}; // 3차년도 매출
        double sum = 0;
        for (int i = 0; i < intArray.length; i++) // 년도에 대한 반복
            for (int j = 0; j < intArray[i].length; j++) // 분기에 대한 반복
                sum += intArray[i][j]; // 분기별 매출의 합

        System.out.println("지난 3년간 매출 총액은 " + sum + "이며 연평균 매출은 " + sum/intArray.length +
"입니다.");
    }
}
```

지난 3년간 매출 총액은 1380.0이며 연평균 매출은 460.0입니다.

불규칙한 2차원 배열을 가지는 예제

```
public class IrregularArray {
    public static void main (String[] args) {
        int intArray[][] = new int[4][]; // 각 행의 레퍼런스 배열 생성
        intArray[0] = new int[3]; // 첫째 행의 3개 정수 배열 생성
        intArray[1] = new int[2]; // 둘째 행의 2개 정수 배열 생성
        intArray[2] = new int[3]; // 셋째 행의 3개 정수 배열 생성
        intArray[3] = new int[2]; // 넷째 행의 2개 정수 배열 생성
    }
}
```

<pre> for (int i = 0; i < intArray.length; i++) // 행에 대한 반복 for (int j = 0; j < intArray[i].length; j++) // 열에 대한 반복 intArray[i][j] = (i+1)*10 + j; for (int i = 0; i < intArray.length; i++) { for (int j = 0; j < intArray[i].length; j++) System.out.print(intArray[i][j]+" "); System.out.println(); } } } </pre>	
<pre> 10 11 12 20 21 30 31 32 40 41 </pre>	

배열을 매개변수로 받는 메소드

```

class StringArray {
    public static void ArrayPrint(String[] ar)
    {
        for(int i=0; i<ar.length; i++)
            System.out.println("배열의 " + i + "번째 요소: " + ar[i]);
    }

    public static void main(String[] args) {
        String[] strArr = new String[3];
        strArr[0] = "사과";
        strArr[1] = "포도";
        strArr[2] = "배";

        ArrayPrint(strArr);
    }
}

```

배열의 0번째 요소: 사과
 배열의 1번째 요소: 포도
 배열의 2번째 요소: 배

```

class StringArray {
    public static int[] addNumArray(int[] ar)
    {
        for(int i=0; i<ar.length; i++)
            ar[i] += 10;

        return ar;
    }

    public static void main(String[] args) {
        int[] numArr1 = {10, 20, 30, 40, 50};
        int[] numArr2;

        numArr2=addNumArray(numArr1);

        for(int i=0; i<numArr2.length; i++)
            System.out.println("numArr2[" + i + "]: " + numArr2[i]);
    }
}

```

```

numArr2[0]: 20
numArr2[1]: 30
numArr2[2]: 40
numArr2[3]: 50
numArr2[4]: 60

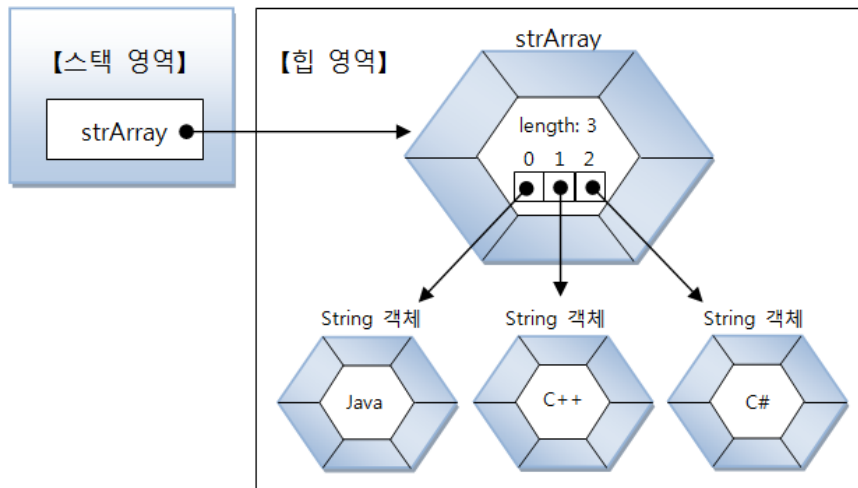
```

객체를 참조하는 배열

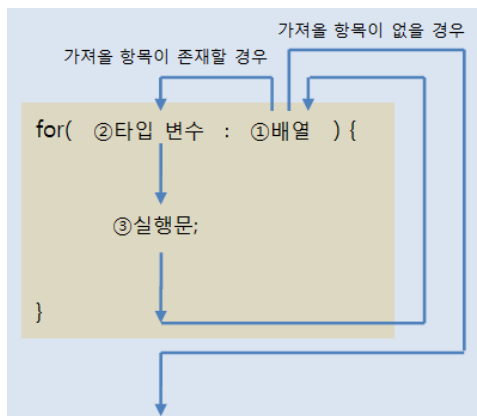
```

String[] strArray = new String[3];
strArray[0] = "Java";
strArray[1] = "C++";
strArray[2] = "C#";

```

향상된 for문



```
int[] scores = { 95, 71, 84, 93, 87 };
```

```
int sum = 0;
for (int score : scores) {
    sum = sum + score;
}
```

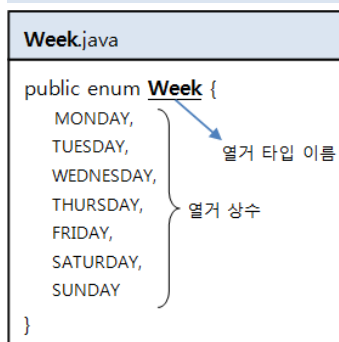
열거 타입

한정된 값만을 갖는 데이터 타입

```
public enum 열거타입이름 { ... }
```

```
public enum Week { MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, ... }
```

```
public enum LoginResult { LOGIN_SUCCESS, LOGIN_FAILED }
```



열거 타입 변수 선언

```
열거타입 변수;
```

```
Week today;
```

```
Week reservationDay;
```

```
Week today = Week.SUNDAY;
```

실습예제4

for-each문을 이용한 반복문 예제를 실습하시오

```
public class foreachEx {
    enum Week { 월, 화, 수, 목, 금, 토, 일 }

    public static void main(String[] args) {
        int[] num = { 1,2,3,4,5 };
        String names[] = { "사과", "배", "바나나", "체리", "딸기", "포도" };
        int sum = 0;

        // 아래 for-each에서 k는 num[0], num[1], ..., num[4]로 반복됨
        for (int k : num)
            sum += k;
        System.out.println("합은 " + sum);

        // 아래 for-each에서 s는 names[0], names[1], ..., names[5]로 반복됨
        for (String s : names)
            System.out.print(s + " ");
        System.out.println();

        // 아래 for-each에서 day는 월, 화, 수, 목, 금, 토, 일 값으로 반복됨
        for (Week day : Week.values())
            System.out.print(day + "요일 ");
        System.out.println();
    }
}
```

합은 15

사과 배 바나나 체리 딸기 포도

월요일 화요일 수요일 목요일 금요일 토요일 일요일

Main() 인자의 이용 예

여러 개의 실수를 main() 메소드 인자로 전달받아 평균값을 구하는 프로그램을 코딩하시오.

```
public class MainParameter {
    public static void main (String[] args) {
        double sum = 0.0;

        for (int i=0; i<args.length; i++) // 인자 개수만큼 반복
            sum += Double.parseDouble(args[i]); // 문자열을 실수로 변환하여 합산
        System.out.println("합계 : " + sum);
        System.out.println("평균 : " + sum/args.length);
    }
}
```

```
C:\workspace\제임스\bin>java MainParameter 77.5 89.6 100
학생번호 : 267.1
평균 : 89.03333333333335
```

실습예제5

```
< GenderType.java >
public enum GenderType {
    WOMAN("여자"), MAN("남자");

    private String string;
    private GenderType(String string) {
        this.string = string;
    }

    @Override
    public String toString() {
        return string;
    }
}
```

```
< Student.java >
public class Student {
    private String studentNo;
    private String name;
    private int age;
    private GenderType gender;

    public Student() {} // Default Constructor

    public Student(String studentNo, String name, int age, GenderType gender) {
        this.studentNo = studentNo;
        this.name = name;
        this.age = age;
        this.gender = gender;
    }

    public String getStudentNo() {
        return studentNo;
    }

    public void setStudentNo(String studentNo) {
        this.studentNo = studentNo;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }
}
```

```

    public void setAge(int age) {
        if (age > 0) {
            this.age = age;
        }
    }

    public GenderType getGender() {
        return gender;
    }

    public void setGender(GenderType gender) {
        this.gender = gender;
    }

    public void displayInformation() {
        System.out.println("학번 : " + studentNo);
        System.out.println("이름 : " + name);
        System.out.println("나이 : " + (age > 0 ? age : "미정"));
        System.out.println("성별 : " + gender.toString());
        System.out.println("-----");
    }
}

```

< ArrayEx1.java >

```

public class ArrayEx1 {
    public static void main(String[] args) {
        Student[] students; // 4바이트 참조변수 (stack area)
        students = new Student[5]; // 배열 생성 (heap area)

        students[0] = new Student(); // student[0] 또한 4바이트. 실제 객체를
        // 포인팅하는 주소를 담는 부분. 실제 객체 포인팅 해주기
        students[0].setStudentNo("S001");
        students[0].setName("HongGilDong");
        students[0].setAge(18);
        students[0].setGender(GenderType.MAN);

        students[1] = new Student("S002", "JiwonKim", 27,
        GenderType.WOMAN);

        for(Student student : students) {
            if (student != null) {
                student.displayInformation();
            }
        }
    }
}

```

```

학번 : S001
이름 : HongGilDong
나이 : 18
성별 : 남자
-----
학번 : S002
이름 : JiwonKim
나이 : 27
성별 : 여자
-----

```

실습예제6

상품 클래스를 3개의 객체를 생성하여 배열에 저장하고 출력 하시오 (클래스도 배열에 저장할 수 있다)

```
import java.util.Scanner;
public class GoodsArray {
    public static void main(String[] args) {
        Goods [] goodsArray;
        goodsArray = new Goods[3];

        for(int i=0; i<goodsArray.length; i++) {
            Scanner s = new Scanner(System.in);
            String name = s.next();      // 상품 이름
            int price = s.nextInt();      // 상품 가격
            int n = s.nextInt();          // 상품 재고
            int sold = s.nextInt();       // 팔린 수량
            goodsArray[i] = new Goods(name, price, n, sold); // 객체 생성
        }

        for(int i=0; i<goodsArray.length; i++) {
            System.out.print(goodsArray[i].getName()+" "); // 상품 이름 출력
            System.out.print(goodsArray[i].getPrice()+" "); // 상품 가격 출력
            System.out.print(goodsArray[i].getNumberOfStock()+" "); // 상품 재고 출력
            System.out.println(goodsArray[i].getSold()); // 팔린 수량 출력
        }
    }
}

class Goods {
    private String name;
    private int price;
    private int numberOfStock;
    private int sold;

    Goods(String name, int price, int numberOfStock, int sold) { // 생성자
        this.name = name;
        this.price = price;
        this.numberOfStock = numberOfStock;
        this.sold = sold;
    }

    String getName() {return name; } // 상품 이름 반환
    int getPrice() {return price; } // 상품 가격 반환
    int getNumberOfStock() {return numberOfStock; } // 상품 재고 반환
    int getSold() {return sold; } // 팔린 수량 반환
}
```

컴퓨터	1000	5	3
프린터	500	4	1
스캐너	300	1	0
컴퓨터	1000	5	3
프린터	500	4	1
스캐너	300	1	0

실습예제7

배열을 리턴 타입으로 가지는 경우의 예제

```
public class ReturnArray {
    static int[] makeArray() { // 정수형 일차원 배열 반환
        int temp[] = new int[4];
        for (int i=0; i<temp.length; i++)
            temp[i] = i;
        return temp;
    }
    public static void main (String[] args) {
        int intArray[]; // 배열 레퍼런스 변수 선언
        intArray = makeArray();
        for (int i = 0; i < intArray.length; i++)
            System.out.println(intArray[i]);
    }
}
```

0
1
2
3

실습예제8

문자열 배열을 이용하여 아래 게임 프로그램을 실습해 보세요

```
import java.util.Scanner;
public class Kawibawibo {
    public static void main (String[] args) {
        Scanner sin = new Scanner(System.in);
        final String s[] = {"가위", "바위", "보"};
        int userChoice, computerChoice;
        while (true) {
            System.out.print("가위(1), 바위(2), 보(3), 끝내기(4) 중 하나를 선택하세요>>");
            userChoice = sin.nextInt();
            if (userChoice == 4)
                break; // 게임 끝내기

            computerChoice = (int) Math.round(Math.random() * 2 + 1); // 1부터 3까지의 수
            if (userChoice < 1 || userChoice > 3) {
                System.out.println("잘못 입력하셨습니다.");
            } else {
                System.out.println("컴퓨터는 " + s[computerChoice-1] + "를 냈습니다.");
                int diff = userChoice - computerChoice;
                switch (diff) {
                    case 0: // 같은 것을 낸 경우
                        System.out.println("비겼습니다.");
                        break;
                    case -1: // 사용자가 가위, 컴퓨터가 바위 또는 사용자가 바위, 컴퓨터가 보
                    case 2: // 사용자가 보, 컴퓨터가 가위
                        System.out.println("당신이 졌습니다.");
                        break;
                    case 1: // 사용자가 바위, 컴퓨터가 가위 또는 사용자가 보, 컴퓨터가 바위
                    case -2: // 사용자가 가위, 컴퓨터가 보
                        System.out.println("당신이 이겼습니다.");
                }
            }
        }
    }
}
```

<pre>} }</pre>	
<p>가위(1), 바위(2), 보(3), 끝내기(4) 중 하나를 선택하세요>>3 컴퓨터는 보를 냈습니다. 비겼습니다. 가위(1), 바위(2), 보(3), 끝내기(4) 중 하나를 선택하세요>></p>	

수고했습니다. 배열과 참조타입을 잘 이해 합시다.

아래 문제를 풀어 보세요.

공통: 1402, 1403, 1407, 1409, 1410, 1411, 1412, 1425, 1430, 1093, 1096, 1501, 2007, 2008, 2601, 2602, 2603 실습문제 1번, 2번

THINKING CODING 