

1. 배열(Array)

ex1-1) 각 선수의 정보(등번호, 이름, 포지션, 성적)를 표시

등번호	이름	포지션	성적
13	허경민	3루수	0.312
25	양의지	포수	0.425
36	오재일	1루수	0.352
34	린드블럼	투수	0.024(방어율)
37	박건우	중견수	0.347

DoosanBears.java

```
public static void main(String[] args) {
    Player p1 = new Player(13, "허경민", 5, 0.312);
    Player p2 = new Player(25, "양의지", 2, 0.425);
    Player p3 = new Player(36, "오재일", 3, 0.352);
    Player p4 = new Player(34, "린드블럼", 1, 0.024);
    Player p5 = new Player(37, "박건우", 8, 0.347);

    Player player[] = new Player[5]; // 길이가 5인 Player라는 배열 생성

    player[0] = p1;
    player[1] = p2;
    player[2] = p3;
    player[3] = p4;
    player[4] = p5;
    // player에 p1 - p5 까지 등록

    String positionStr[] = {"지명타자", "투수", "포수", "1루수", "2루수", "3루수", "유격수", "좌익수",
        "중견수", "우익수"};
    // 표기할 Position 배열 생성
    System.out.println("등번호\t이름\t포지션\t성적");
    System.out.println("-----");
    int len = player.length // len이라는 integer 변수 생성 후 player의 배열 길이를 값으로 한다.
        // 즉 5가 될 것이다.
    for (int i = 0; i < len; i++) { // i를 0부터 4까지 증가시키면서
        System.out.println(player[i].getNum() + "\t" + player[i].getName() +
            "\t" + positionStr[player[i].getPosition()] + "\t" + player[i].getScore() +
            (player[i].getPosition() == 1 ? "(방어율)" : ""));
        // player[] 배열의 등번호, 이름과, 성적을 받아오면서 중간에 player[] 배열에서 받아온 포지션의 값을
        // 받아와서 positionStr[] 배열로 입력해주면 포지션의 명칭이 출력될 것이며 player[] 배열에서 포지션의
        // 값이 1이면 성적 옆에 (방어율) 문자열 표시 1이 아니면 blank 처리해서 출력되지 않게 해주면
        // 위와 같이 출력 될 것이다.
    }
    ...
}
```

ex 1-2) 두 번째 문제 : 등 번호에 해당하는 선수 정보 표시 없으면 "00번에 해당하는 선수는 없습니다." 표시

```
박건우(37) : 중견수, 0.347
(int num= 39 대입하면.)
// 39번 선수는 없습니다.
```

DoosanBears.java

```
...
int num = 37; // 등번호 37번을 num라는 integer 변수로 생성
Player splayer = null; // Player 객체에서 나온 splayer라는 생성자를 만들면서
    null 값으로 초기화 하면서.
for (int i = 0; i < len; i++) {
    // i를 0부터 4까지 증가시키면서
    if(player[i].getNum() == num) {
        splayer = player[i];
        //player[] 배열의 num값을 가져와서 지정한 값과 일치 하면 해당 선수의 정보를
        // 생성자에 입력 후
        break; // 다른 문으로 빠져 나가라
    }
}

if(splayer != null) {
    System.out.println(splayer.getName() + "(" + splayer.getNum() +
        ") : " + positionStr[splayer.getPosition()] + ", " +
        splayer.getScore());
    // 그렇게 함으로써 생성자 splayer의 값이 비어있지 않으면 위와 같이 출력을 하고
} else {
    System.out.println(num + "선수는 없습니다."); // 아니면 "선수가 없다."라고 출력
}
...
```

ex 1-3) 해당 선수가 안타를 쳤다면 성적이 0.012 상승시켜서 선수정보 출력.

박건우(37) : 중견수, 0.359

DoosanBears.java

```
...
    System.out.println(num + " 선수가 안타.");
    splayer.setScore(splayer.getScore() + 0.012);
    // ex 1-2)에서 받아온 splayer생성자에서 점수를 받아온 값에 조건대로 0.012를 추가한 값을
    setScore() 메서드에 입력한다.
    System.out.println(splayer.getName() + "(" + splayer.getNum() +
        ") : " + positionStr[splayer.getPosition()] + ", " + splayer.getScore());
    // 위와 같은 조건대로 출력 한다.
...
```

Player.java

```
package com.kitri.array1;
public class Player {
    private int num;
    private String name;
    private int position;
    private double score;
    public Player(int num, String name, int position, double score) {
        this.num = num;
        this.name = name;
        this.position = position;
        this.score = score;
    }
    public int getNum() {
        return num;
    }
    public String getName() {
        return name;
    }
    public int getPosition() {
        return position;
    }
    public double getScore() {
        return score;
    }
    public void setScore(double score) {
        this.score = score;
    }
}
```

2. API Classes (sub - String Class)

- Wrapper Class : 기본 데이터 타입을 Class화 시킨 Class

기본 Reference Data type	Wrapper Class
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

ex) WrapperClassTest.java

```
public static void main(String[] args) {
    String s = "true";
    // String s = "treu"; → 오타가 나면 true를 제외한 나머지는 오타
    // boolean b= (boolean) s; → error : 같은 Data Type만 Casting 연산가능
    Boolean b1 = new Boolean(s);
    // 문자열 변수 s를 b1이라는 Boolean Type 클래스 생성자로 입력한다.
    boolean b = b1.booleanValue(); // b라는 boolean type 변수 생성 후 b1의 값을 booleanValue()
                                   메서드를 이용하여 boolean Type 변수로 반환한다.

    if(b) {
        System.out.println("1. 참입니다.");
    } else {
        System.out.println("1. 거짓입니다.");
    } // 출력 : 1. 참입니다.

    boolean b2 = Boolean.parseBoolean(s); // b2라는 boolean 타입 변수 생성 후 Boolean 클래스의
                                           parseBoolean() 메서드를 이용하여 s라는 문자열 변수를
                                           Boolean Type 값으로 변환 한다.

    if(b) {
        System.out.println("2. 참입니다.");
    } else {
        System.out.println("2. 거짓입니다.");
    } // 출력 : 2. 참입니다.

    String s2 = "123"; // 숫자만을 포함하는 문자열 변수 s2 생성
    Integer inte = Integer.parseInt(s2); // 클래스 Integer 생성 후 inte라는 이름으로 만든 후
                                           Integer클래스의 parseInt() 메서드를 이용하여 s2의 값을 숫자로
                                           반환한다.

    int total = inte + 10; // total = 123 + 10
    System.out.println(s2 + " + "+10+" = " +total); // 출력 : 123+10 = 133

    // autoboxing : 기본을 Ref에다가 Ref를 값으로 - JDK 1.4 버전 이하 버전은 안 됨.
    Integer it = new Integer("123");
    int y= it; // unboxing
    Integer it2 = y; // boxing
}
```

ex) Math 클래스를 이용한 예제 (Circle.java)

```
package com.kitri.lengtest;

public class Circle {
    int radius;
    public Circle(int radius) {
        super();
        this.radius = radius;
    }

    public double calcArea(){
        return Math.PI * Math.pow(radius, 2); // PI=3.14, pow(a,b) =  $a^b$ 
    }

    public double calcRound(){
        return 2 * Math.PI * radius;
    }
}
```

ex) String 클래스를 이용한 예제 1 (StringTest1.java)

```

... {

    String s1 = "java";
    String s2 = "java";
    String s3 = new String("java");
    String s4 = new String("java");
    if(s1 == s2) {
        System.out.println("s1와 s2는 주소값 같다");
    }
    if(s1 == s3) {
        System.out.println("s1와 s3는 주소값 같다");
    }
    if(s1 == s4) {
        System.out.println("s1와 s4는 주소값 같다");
    }
    if(s2 == s3) {
        System.out.println("s2와 s3는 주소값 같다");
    }
    if(s3 == s4) {
        System.out.println("s3와 s4는 주소값 같다");
    }
    // 출력 : s1와 s2는 주소값 같다.
    (이유 : 힙 - 문자열 저장소에 저장되고, 이후 s2 생성시 문자열을 확인하여 있으면, 주소값만
    참조한다. 객체와 객체 사이의 값 비교는 주소값 비교)

    System.out.println("=====");
    if(s1.equals(s2)) {
        System.out.println("s1와 s2는 문자열 같다");
    }
    if(s1.equals(s3)) {
        System.out.println("s1와 s3는 문자열 같다");
    }
    if(s1.equals(s4)) {
        System.out.println("s1와 s4는 문자열 같다");
    }
    if(s2.equals(s3)) {
        System.out.println("s2와 s3는 문자열 같다");
    }
    if(s2.equals(s4)) {
        System.out.println("s3와 s4는 문자열 같다");
    }
    if(s3.equals(s4)) {
        System.out.println("s3와 s4는 문자열 같다");
    }
    // 출력 : if 이하로 출력 다 된다.
    (이유 : 문자열 비교로 같다 / string 클래스가 가지고 있는 문자열은 Overriding)
}

```

ex) String 클래스를 이용한 예제 2 (StringTest2.java)

```

... {

    String s1 = new String();
    System.out.println("s1 == " + s1);
    System.out.println("s1 is Empty == " + s1.isEmpty());
    // isEmpty():문자열 길이가 0이면 true 생성
    System.out.println("s1 길이 == " + s1.length()); // length(): 길이 출력

    byte b[] = {97,98,99,100,101,102,103};
    String s2 = new String(b); // 숫자 배열을 String 객체에 입력하면
    System.out.println("s2 == "+s2); // ASCII 코드로 번역 되서 출력된다.
        // 출력 : abcdefg

    String s3 = new String(b,2,4); // 바이트 배열 b를 2번째 자리부터 시작해서 길이가 4인
        문자열 생성.
    System.out.println("s3 == "+s3); // 출력 : cdef

    char c[] = {'안','녕','하','세','요'};
    String s4 = new String(c); // 캐릭터 배열을 문자열로 변환
    System.out.println("s4 == "+s4); // 출력 : 안녕하세요

    String s5 = new String(c,2,3); // Character 배열 c를 2번째 자리부터 시작해서 길이가 3인
        문자열 생성.
    System.out.println("s5==" +s5); // 출력 : 하세요

}
}

```

ex) String 클래스를 이용한 예제 3 (StringTest3.java)

```

... public static void main(String[] args) {
    String str = "hello java !!!"; // 문자열 str 생성
    char c = str.charAt(4); // charAt(int) : int번째 자리의 글자를 반환
    System.out.println("c == "+c); // 출력 : c==o
    String jumin = "020408-4012324"; // 실제 주민등록번호 입력
    int gender = jumin.charAt(7) - 48; // 문자열 '4'에서 숫자 4로 변경해야 함.
        즉 문자열 4의 ASCII는 52이고 거기서 48을 빼면 4가 나온다.
    String result= gender%2==0 ? "여자" : "남자" ;// gender를 2로 나누었을 때 나머지가 0이면 여자 아니면 남자
    System.out.println("성별 : "+result); // 출력 : 여자

    String s1 = "jAVa";
    String s2 = "java";
    if(s1.toLowerCase().equals(s2.toLowerCase())) {
        System.out.println(s1+ "과 "+s2+"는 대소문자 구분 없이 같은 문자열 입니다.");
    } // 대소문자 구분하지 않는 방법 1 : s1의 문자열을 소문자(toLowerCase())로 한 다음에 s2를 소문자한 결과값을
        equals() 메서드를 사용해서 비교하는 방법

    if(s1.equalsIgnoreCase(s2)) {
        System.out.println(s1+ "과 "+s2+"는 대소문자 구분 없이 같은 문자열 입니다.");
    } // 대소문자 구분하지 않는 방법 2 : equalsIgnoreCase(string) : 대소문자 구분을 무시하고 안의 string과 비교
        즉 s1의 대소문자를 무시하고 s2와 비교한다.

    System.out.println(s1+ "을 대문자로 : "+s1.toUpperCase()); // toUpperCase() : 대문자 변경 출력: JAVA
    System.out.println(s1+ "을 소문자로 : "+s1.toLowerCase()); // toLowerCase() : 소문자 변경 출력 : java

}

```

ex) String 클래스를 이용한 예제 4 (StringTest4.java)

```
... public static void main(String[] args) {
    String str = "hello java !!!";
    byte b[] = str.getBytes(); // getBytes() : 문자열 클래스를 byte[] 배열 b로 가져온다.
    int len = b.length; // 배열 길이
    for (int i = 0; i < len; i++) {
        System.out.print(b[i] + " ");
    }
    System.out.println();
    // 출력 : 104 101 108 108 111 32 106 97 118 97 32 33 33 33
    char c[] = str.toCharArray(); // toCharArray () : str클래스를 Character 배열화 시킨다.
    len = c.length; // 배열 길이
    for (int i = 0; i < len; i++) {
        System.out.print(c[i] + " ");
    }
    // 출력 : h e l l o   j a v a   ! ! !
}
```

ex) String 클래스를 이용한 예제 5 (StringTest5.java)

```
... public static void main(String[] args) {
    String str = "hello";
    System.out.println(str.concat("java")); // concat(string) : 원래 있는 문자열에 괄호 안 string을 추가
    // 출력 : hellojava
    System.out.println(str); // 출력 : hello (이유 : 문자열은 불변하기 때문)

    if (str.startsWith("he")) {
        //startsWith(string) string이라는 문자열로 시작하는 지 여부를 묻는다.(True/False)로 결과 출력
        System.out.println(str + "은 he로 시작합니다."); // 출력 : hello은 he로 시작합니다.
    }
    if (str.endsWith("java")) {
        //endsWith(string) string이라는 문자열로 끝나는지 여부를 묻는다.(True/False)로 결과 출력
        System.out.println(str + "은 java로 끝납니다."); // 출력 : 출력값 없음
    }
}
```

ex) 반대로 된 글자를 뒤집기 - Reverse Character (StringReverseTest.java)

1) String str = "!!! avaJ olleH";을 Hello Java !!!로 바꾸시오 (방법 : 5가지)

```

... public static void main(String[] args) {
    String str = "!!! avaJ olleH";
    // 1. String Buffer 활용
    StringBuffer strBuffer = new StringBuffer();
    strBuffer.append(str);
    System.out.println(strBuffer.reverse());

    // 2. for문으로 돌려서 뒤집기(단, 문자 위치 추출은 불가능)
    for (int i = str.length() - 1; i >= 0; i--) {
        System.out.print(str.charAt(i));
    }
    System.out.println();

    // 3. for문으로 뒤집음과 동시에 문자열로 더해서 추출
    String rs = "";
    for (int i = str.length() - 1; i >= 0; i--) {
        rs += str.charAt(i);
    }
    System.out.println(rs);

    // 4. Byte 추출
    byte b[] = str.getBytes();
    int len = b.length;
    for (int i = len - 1; i >= 0; i--) {
        System.out.print((char) b[i]);
    }
    System.out.println();

    // 5. 문자열 배열 이용
    char c[] = str.toCharArray();
    int count = c.length;
    char rc[] = new char[count];
    int x=0;
    for (int i = count-1; i >= 0; i--) {
        rc[x++] = c[i];
    }
    String rs3 = new String(rc);
    System.out.println(rs3);
}
...

```


2) 뒤집은 문자열 중 'o'를 몇 번째에 있다고 출력하시오 (방법 : 2가지) [출력 : reverse한 문자열 중 'o'는 4번째에 있습니다.]

```
...  
  
    // 1. String Buffer 문자열이용.  
    StringBuffer sb = new StringBuffer(strBuffer);  
    String str2;  
    str2 = sb.toString();  
    System.out.println("reverse한 문자열중 'o'는 " + str2.indexOf('o') + "번째에 있습니다.");  
  
    // 2. 아까 만들어 뒀던 rs 추출  
    System.out.println("reverse한 문자열중 'o'는 " + rs.indexOf('o') + "번째에 있습니다.");  
}
```