



2020년 2회 정보처리기사 실기 시험 100% 합격전략집

1
일차

2
일차

3
일차

4
일차

5
일차

6
일차

7
일차

8
일차

9
일차

10
일차

11
일차

12
일차

13
일차

14
일차

15
일차

16
일차

17
일차

18
일차

19
일차

20
일차

1장 프로그래밍 언어 활용

핵심 010 for문

핵심 011 기타 반복문

핵심 012 배열과 문자열

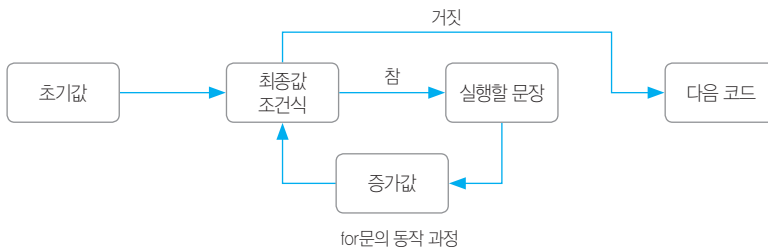
핵심 013 Java에서의 배열과 문자열



2020년 2회 정보처리기사 실기 대비용 핵심요약

[핵심010] for문

- 초기값, 최종값, 증가값을 지정하는 수식을 이용해 정해진 횟수를 반복하는 제어문이다.
- 초기값을 정한 다음 최종값에 대한 조건이 참이면 실행할 문장을 실행한 후 초기값을 증가값 만큼 증가시키면서 최종값에 대한 조건이 참인 동안 실행할 문장을 반복 수행한다.



- 형식

for(식1; 식2; 식3)

- for는 반복문을 의미하는 예약어로 그대로 입력한다.
- 식1 : 초기값을 지정할 수식을 입력한다.
- 식2 : 최종값을 지정할 수식을 입력한다.
- 식3 : 증가값으로 사용할 수식을 입력한다.

실행할 문장;

식2가 참일 동안 실행할 문장을 입력한다. 실행할 문장이 두 문장 이상일 경우 { }를 입력하고 그 사이에 처리할 문장들을 입력한다.

- 초기값, 최종값, 증가값 중 하나 이상을 생략할 수 있고, 각각의 요소에 여러 개의 수식을 지정할 수도 있다.
- for문은 처음부터 최종값에 대한 조건식을 만족하지 못하면 한 번도 수행하지 않는다.

2020년 2회 가능사 실기

- 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오.

```
#include <stdio.h>
main() {
    int i, j;
    for (i = 2; i <= 4; i++) {
        for (j = 5; j <= 7; j++) {
        }
    }
    printf("%d × %d = %d", j, i, i * j);
}
```

답 :



```
#include <stdio.h>
main() {
    int i, j;
    ❶ for (i = 2; i <= 4; i++) {
    ❷         for (j = 5; j <= 7; j++) {
    ❸             }
    ❹         }
    ❺ printf("%d × %d = %d", j, i, i * j);
}
```

- ❶ 반복 변수 i 가 2에서 시작하여 1씩 증가하면서 4보다 작거나 같은 동안 ❷번 문장을 반복하여 수행한다. 결국 ❷번을 3번 반복하여 수행한다.
- ❷ 반복 변수 j 가 5에서 시작하여 1씩 증가하면서 7보다 작거나 같은 동안 반복하는데 실행할 문장 없이 for문만 반복 수행한다.
- ❸ ❷번 반복문의 끝이다.
- ❹ ❶번 반복문의 끝이다.
- ❺ " $j \times i = j * i$ "의 형태로 변수의 값을 출력한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

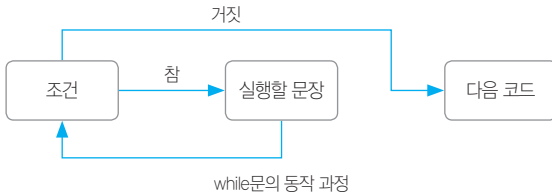
i	j
2	5
	6
	7
	8
3	5
	6
	7
	8
4	5
	6
	7
	8
5	

정답 1. $8 \times 5 = 40$

[핵심011] 기타 반복문

while문

- 조건이 참인 동안 실행할 문장을 반복 수행하는 제어문이다.
- while문은 조건이 참인 동안 실행할 문장을 반복 수행하다가 조건이 거짓이면 while문을 끝낸 후 다음 코드를 실행한다.
- while문은 조건이 처음부터 거짓이면 한 번도 수행하지 않는다.



- 형식

while(조건)

실행할 문장;

- while은 반복문에 사용되는 예약어로 그대로 입력한다.
 - (조건) : 참이나 거짓을 결과로 갖는 수식을 '조건'에 입력한다. 참()을 직접 입력할 수도 있다.
- 조건이 참인 동안 실행할 문장을 입력한다. 문장이 두 문장 이상인 경우 { }를 입력하고 그 사이에 처리할 문장들을 입력한다.

do~while문

- 조건이 참인 동안 정해진 문장을 반복 수행하다가 조건이 거짓이면 반복문을 벗어나는 while문과 같은 동작을 하는데, 다른 점은 do~while문은 실행할 문장을 무조건 한 번 실행한 다음 조건을 판단하여 탈출 여부를 결정한다는 것이다.
- do~while문은 실행할 문장을 우선 실행한 후 조건을 판별하여 조건이 참이면 실행할 문장을 계속 반복 수행하고, 거짓이면 do~while문을 끝낸 후 다음 코드를 실행한다.



- 형식

do

실행할 문장;

while(조건);

- do는 do~while문에 사용되는 예약어로, do~while의 시작 부분에 그대로 입력한다.
- 조건이 참인 동안 실행할 문장을 입력한다. 문장이 두 문장 이상인 경우 { }를 입력하고 그 사이에 실행할 문장들을 입력한다.
- while은 do~while문에 사용되는 예약어로, do~while의 끝 부분에 그대로 입력한다.
- (조건) : 참이나 거짓을 결과로 갖는 수식을 '조건'에 입력한다. 참()을 직접 입력할 수도 있다.

break, continue

- switch문이나 반복문의 실행을 제어하기 위해 사용되는 예약어이다.
- break : switch문이나 반복문 안에서 break가 나오면 블록을 벗어난다.

- continue
 - continue 이후의 문장을 실행하지 않고 제어를 반복문의 처음으로 옮긴다.
 - 반복문에서만 사용된다.

1. 다음 C언어로 구현된 프로그램을 실행했을 때 최종적으로 출력되는 결과를 쓰시오.

```
#include <stdio.h>
main() {
    int x = 100, y = 10;
    do {
        x -= y;
        printf("%d, %d\n", x, y++);
    } while (x > 30);
}
```

답 :

```
#include <stdio.h>
main() {
    ① int x = 100, y = 10;
    ② do {
    ③     x -= y;
    ④     printf("%d, %d\n", x, y++);
    ⑤ } while (x > 30);
}
```

- ① 정수형 변수 x와 y를 선언한 후 각각 100과 10으로 초기화한다.
- ② do~while 반복문의 시작점이다. ③, ④번 문장을 반복하여 수행한다.
- ③ 'x = x - y;'와 동일하다. x의 값에서 y의 값을 뺀 후 x에 저장한다.
- ④ x와 y를 정수형 10진수로 출력한 후 커서를 다음 줄 맨 처음으로 이동한다. y++로 인해 ④번 문장 종료 후 y의 값이 1 증가한다.
- ⑤ x가 30보다 큰 동안 ③, ④번 문장을 반복하여 수행한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

X	Y
100	10
90	11
79	12
67	13
54	14
40	15
25	16

정답 1, 25, 15

[핵심012] 배열과 문자열

배열의 개념

- 동일한 데이터 유형을 여러 개 사용해야 할 경우 이를 손쉽게 처리하기 위해 여러 개의 변수들을 조합해서 하나의 이름으로 정의해 사용하는 것을 말한다.
- 배열은 변수명 뒤에 대괄호 []를 붙이고 그 안에 사용할 개수를 지정한다.
- C언어에서 배열의 위치는 0부터 시작된다.
- C언어에서 배열 위치를 나타내는 첨자 없이 배열 이름을 사용하면 배열의 첫 번째 요소의 주소를 지정하는 것과 같다.

1차원 배열

- 변수들을 일직선상의 개념으로 조합한 배열이다.
- 형식

자료형 변수명[개수];

- 자료형 : 배열에 저장할 자료의 형을 지정한다.
- 변수명 : 사용할 배열의 이름으로 사용자가 임의로 지정한다.
- 개수 : 배열의 크기를 지정하는 것으로 생략할 수 있다.

예 int a[5] : 5개의 요소를 갖는 정수형 배열 a

	첫 번째	두 번째	세 번째	네 번째	다섯 번째
배열 a	a[0]	a[1]	a[2]	a[3]	a[4]

※ a[3] : a는 배열의 이름이고, 3은 첨자로서 배열 a에서의 위치를 나타낸다.

a[3]에 4를 저장시키려면 'a[3] = 4'와 같이 작성한다.

2차원 배열

- 변수들을 평면, 즉 행과 열로 조합한 배열이다.
- 형식

자료형 변수명[행개수][열개수]

- 자료형 : 배열에 저장할 자료의 형을 지정한다.
- 변수명 : 사용할 배열의 이름으로 사용자가 임의로 지정한다.
- 행개수 : 배열의 행 크기를 지정한다.
- 열개수 : 배열의 열 크기를 지정한다.

예 int b[3][3] : 3개의 행과 3개의 열을 갖는 정수형 배열 b

	열		
행	0, 0	0, 1	0, 2
	1, 0	1, 1	1, 2
	2, 0	2, 1	2, 2

※ b[0][2] : b는 배열의 이름이고, 0은 행 첨자, 2는 열 첨자로서 배열 b에서의 위치를 나타낸다.

배열 형태의 문자열 변수

- C언어에서는 큰따옴표(“ ”)로 묶인 글자는 글자 수에 관계없이 문자열로 처리된다.
- C언어에는 문자열을 저장하는 자료형이 없기 때문에 배열 또는 포인터를 이용하여 처리한다.
- 형식

```
char 배열이름[크기] = "문자열"
```

- 배열에 문자열을 저장할 때는 배열 선언 시 초기값으로 지정해야 하며, 이미 선언된 배열에는 문자열을 저장할 수 없다.
- 문자열 끝에 자동으로 널 문자('\0')가 삽입되므로, 널 문자까지 고려하여 배열 크기를 지정해야 한다.

예 char a[5] = "love" →

l	o	v	e	\0
---	---	---	---	----

1. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오.

```
#include <stdio.h>
main() {
    int exint[ ] = { 4,9,3,7,6,4,24,4,8,4,10 };
    int len = sizeof(exint) / sizeof(int);
    int value = 0;
    for (int i = 0; i < len; i++) {
        if (exint[i] == 4) {
            value++;
        }
    }
    printf("%d", value);
}
```

답 :

```
#include <stdio.h>
main() {
    ① int exint[] = { 4,9,3,7,6,4,24,4,8,4,10 };
    ② int len = sizeof(exint) / sizeof(int);
    ③ int value = 0;
    ④ for (int i = 0; i < len; i++) {
    ⑤     if (exint[i] == 4) {
    ⑥         value++;
    }
    }
    ⑦ printf("%d", value);
}
```

① 배열을 선언할 때 사용할 개수를 생략하고 초기값을 지정하면, 초기값으로 지정된 값의 수와 같은 크기의 배열이 선언된다.

exint[11] 4 9 3 7 6 4 24 4 8 4 10

② 정수형 변수 len을 선언하고 초기값으로 배열 exint의 크기에서 int 자료형의 크기를 나눈 값을 할당한다. int 자료형은 4Byte이고, exint는 int 자료형의 요소 11개를 가지고 있으므로 44Byte이다. 즉 len은 exint의 요소 개수인 11을 저장한다.

③ 정수형 변수 value를 선언하고 초기값으로 0을 할당한다.

④ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 len보다 작은 동안 ⑥번을 반복하여 수행한다.

⑤ exint 배열의 i번째 값이 4이면 ⑥번을 수행한다.

⑥ 'value = value + 1'과 동일하다. value의 값을 1씩 누적시킨다.

⑦ 결과 4

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

exint[11] 4 9 3 7 6 4 24 4 8 4 10

value	i	exint[i]	출력
0	0	4	4
1	1	9	
2	2	3	
3	3	7	
4	4	6	
	5	4	
	6	24	
	7	4	
	8	8	
	9	4	
	10	10	
	11		

정답 1. 4

[핵심013] Java에서의 배열과 문자열

Java에서의 배열 처리

- Java에서는 향상된 for문을 사용할 수 있는데, 향상된 for문은 객체를 대상으로만 가능하다.
- Java에서는 배열을 객체로 취급하며, 배열을 이용하여 작업할 때 필요할 만한 내용은 이미 API로 만들어 두었기 때문에 잘 골라서 사용하면 된다.

예제 다음은 Java에서 향상된 for문을 사용한 예제이다. 결과를 확인하시오.

```
public class Example {
    public static void main(String[] args) {
        ① int[] a = {90,100,80,70,60,50,30};
        ② int hap = 0;
        ③ float avg;
        ④ for (int i : a)
        ⑤     hap = hap + i;
        ⑥     avg = (float)hap / a.length;
        ⑦     System.out.printf("%d, %.2f", hap, avg);
    }
}
```

코드 해설

- ① 배열을 선언하면서 초기값을 지정한다. 개수를 지정하지 않으면 초기값으로 지정된 수만큼 배열의 요소가 만들어진다. 이건 C언어와 같다.

	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
배열 a	90	100	80	70	60	50	30

- ② 총점을 구하기 위해 정수형 변수 hap을 선언하고 초기값으로 0을 할당한다.
 ③ 평균을 구하기 위해 실수형 변수 avg를 선언한다.
 ④ 향상된 반복문이다. a 배열의 요소 수만큼 ⑤번을 반복 수행한다.
- int i : a 배열의 각 요소가 일시적으로 저장될 변수를 선언한다. a 배열과 형이 같아야 한다. 즉 a 배열이 정수면 정수, 문자면 문자여야 한다.
 - a : 배열의 이름을 입력한다. a 배열이 7개의 요소를 가지므로 각 요소를 i에 저장하면서 ⑤번을 7번 수행한다.
- ⑤ i의 값을 hap에 누적한다. i는 a 배열 각 요소의 값을 차례로 받는다.

변수의 변화는 다음과 같다.

첫 번째 수행 : a 배열의 첫 번째 값이 i를 거쳐 hap에 누적된다.

hap	i	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
90	90	90	100	80	70	60	50	30

두 번째 수행 : a 배열의 두 번째 값이 i를 거쳐 hap에 누적된다.

hap	i	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
190	100	90	100	80	70	60	50	30

세 번째 수행 : a 배열의 세 번째 값이 i를 거쳐 hap에 누적된다.

hap	i	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
270	80	90	100	80	70	60	50	30

이런 방식으로 a 배열의 요소 수만큼 반복한다.

⑥ 총점이 저장되어 있는 hap을 실수형으로 변환한 후 a 배열의 요소 수로 나눠 평균을 구한다.

- length : 클래스에는 클래스의 속성과 수행해야 할 메소드가 포함되어 있다. length는 배열 클래스의 속성으로 배열 요소의 개수가 저장되어 있다. a 배열은 7개의 요소를 가지므로 a.length는 7을 가지고 있다.
- a.length : 개체 변수의 이름과 메소드는 (마침표)로 연결하여 사용한다.

⑦ 결과 480, 68.57

Java의 문자열

Java에서는 문자열을 처리할 수 있도록 클래스를 제공하므로 그에 따른 속성과 메소드도 당연히 지원하는데 여기서는 문제 풀이에 꼭 필요한 속성과 메소드만 학습한다.

예제 다음은 문자열을 거꾸로 출력하는 Java 프로그램이다. 결과를 확인하시오.

```
public class Example {
    public static void main(String[] args){
        ① String str = "Information!";
        ② int n = str.length();
        ③ char[] st = new char[n];
        ④ n--;
        ⑤ for (int k = n; k >= 0; k--) {
        ⑥     st[n-k] = str.charAt(k);
        ⑦ }

        ⑧ for (char k : st) {
        ⑨     System.out.printf("%c", k);
        }
    }
}
```

코드 해설

- ① 문자열 변수 str을 선언하면서 초기값으로 "Information!"을 할당한다. 객체 변수를 생성할 때는 예약어 new를 입력해야 하지만 문자열 변수는 초기값을 이용해 new 없이 바로 생성할 수 있다.
- ② 문자열 클래스에서 length() 메소드는 해당 문자열의 길이를 반환한다. 즉 정수형 변수 n에 str에 저장된 문자열의 길이 12가 저장된다.
- ③ Java에서는 배열도 클래스이므로 생성할 때는 new를 사용해야 한다. n에 12가 저장되어 있으므로 st는 12개의 요소를 갖는 문자 배열로 생성된다.
 - char[] st : 문자 배열이고 배열 이름은 st이다. 'char st[]'처럼 입력해도 된다. st만 임의로 입력하고 나머지는 그대로 적어준다.
 - new : 객체 변수를 생성하는 예약어다. 그대로 입력한다.
 - char[n] : 문자 배열의 크기를 지정하므로 12개 요소를 갖는 문자 배열이 생성된다. n을 제외한 나머지는 항상 그대로 입력한다.
- ④ 12개짜리 배열이지만 배열의 첨자는 0부터 시작하여 11까지 사용하기 때문에 첨자로 사용할 변수의 값을 1 감소시킨다(n = 11).

st[0]	st[1]	st[2]	st[3]	st[4]	st[5]	st[6]	st[7]	st[8]	st[9]	st[10]	st[11]
st											

- ⑤ 정수형 변수 k를 반복 변수로 선언하면서 초기값으로 11을 갖고, 1씩 감소시키면서 0보다 크거나 같은 동안 ⑥번을 반복 실행한다. 실행할 문장이 한 개이므로 ⑤, ⑦번의 중괄호는 없어도 된다. 반복 변수 k는 for문 안에서 선언한 지역 변수이기 때문에 for문을 벗어나면 소멸된다.
- ⑥ charAt() 메소드는 해당 문자열에서 인수에 해당하는 위치의 문자열을 반환한다.

첫 번째 수행

k = 11, n = 110 |므로 str.charAt(k)는 '!'을 반환한다. '!'을 s[n-k]번째, 즉 s[0]번째에 저장한다.

s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]	s[7]	s[8]	s[9]	s[10]	s[11]
!											

두 번째 수행

k = 10, n = 110 |므로 str.charAt(k)는 'n'을 반환한다. 'n'을 s[1]에 저장한다.

s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]	s[7]	s[8]	s[9]	s[10]	s[11]
!	n										

위와 같은 작업을 반복수행하다가 k = 0일 때, '!'를 s[11]에 저장한 다음 k는 -1이 되어 반복문을 벗어난다.

s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]	s[7]	s[8]	s[9]	s[10]	s[11]
!	n	o	i	t	a	m	r	o	f	n	!

- ⑧ 항상된 반복문이다. s 배열의 각 요소를 처음부터 차례대로 문자 변수 k에 옮기면서 s 배열의 개수, 즉 ⑨번을 12회 반복 수행한다.
 - ⑨ k에는 s 배열 각 요소의 값이 할당되므로 12회 수행을 마치면 출력 결과는 다음과 같다. 서식 문자열에 '\n'이 없으므로 한 줄에 붙여 출력한다.
- 결과 !noitamrofni

2020년 1회 기사 실기

1. 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.)

```
public class Test {
    static int[] arr() {
        int a[] = new int[4];
        int b = a.length;
        for(int i = 0; i < b; i++)
            a[i] = i;
        return a;
    }

    public static void main(String[] args) {
        int a[] = arr();
        for(int i = 0; i < a.length; i++)
            System.out.print(a[i] + " ");
    }
}
```

답 :

모든 Java 프로그램은 반드시 main() 메소드부터 시작해야 한다.

```
public static void main(String[] args) {
    ❶ int a[] = arr();
        for(int i = 0; i < a.length; i++)
            System.out.print(a[i] + " ");
}
```

❶ 정수형 배열 a를 선언하고, 인수 없이 arr() 메소드를 호출한 다음 돌려받은 값을 배열 a에 저장한다.

```
2 static int[] arr() {
3     int a[] = new int[4];
4     int b = a.length;
5     for(int i = 0; i < b; i++)
6         a[i] = i;
7     return a;
}
```

❷ 메소드의 리턴값이 정수형 배열인 arr() 메소드의 시작점이다.

❸ 4개의 요소를 갖는 정수형 배열 a를 선언한다.

❹ 정수형 변수 b를 선언하고 배열 a의 길이인 4로 초기화한다.

• length : 배열 클래스의 속성으로 배열 요소의 개수가 저장되어 있다. a 배열은 4개의 요소를 가지므로 a.length는 4를 가지고 있다.

❺ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 b보다 작은 동안 ❹번 문장을 반복 수행한다.

❻ a[i]에 반복 변수 i의 값을 저장한다.

	a[0]	a[1]	a[2]	a[3]
배열 a	0	1	2	3

❼ 배열 a에 저장된 값들을 호출한 곳(main() 메소드)으로 반환한다.

```
public static void main(String[] args) {
    ❸ int a[] = arr();
    ❹ for(int i = 0; i < a.length; i++)
    ❺ System.out.print(a[i] + " ");
}
```

❸ arr() 메소드로부터 반환받은 값들을 main() 메소드의 배열 a에 저장한다.

❹ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 a 배열의 길이인 4보다 작은 동안 ❺번 문장을 반복 수행한다.

❺ a[i]의 값과 공백 한 칸을 출력한다.

정답 1.0123