

배열



- 배열 구조
- 1차원 배열
- 다차원 배열
- `System.arraycopy()`
- J2SE 5.0 확장

배열 구조

- 같은 자료형일 경우 변수를 여러 개 선언하면 비효율적이다.

```
int a1, a2, a3, a4;
```

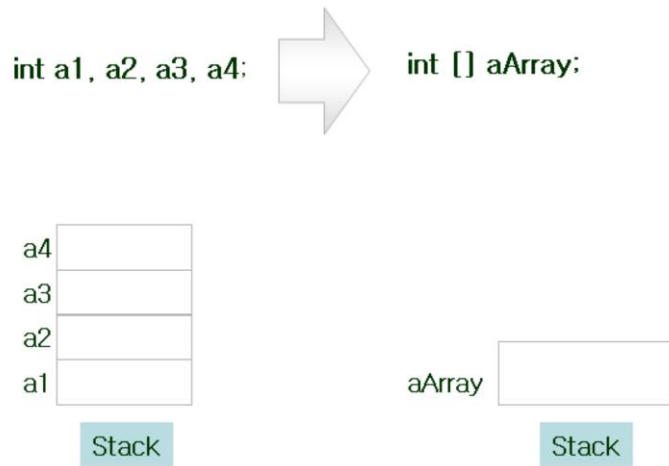
```
a1=1;  
a2=2;  
a3=3;  
a4=4;
```

a4	4
a3	3
a2	2
a1	1

Stack

배열 구조

- 배열은 연속적인 메모리 공간 같은 자료형을 여러 개 선언할 수 있다.



5-3

배열 구조

- 배열은 new를 사용하는 객체이다.
- 단 클래스가 없어서 []를 이용하여 배열 크기를 지정한다.

```
int a1, a2, a3, a4;
```

```
a1=1;
```

```
a2=2;
```

```
a3=3;
```

```
a4=4;
```



```
int [] aArray;
```

```
aArray = new int[4];
```



	0xa001
[0]	0
[1]	0
[2]	0
[3]	0

Heap

5-4

배열 구조

- 배열의 주소가 전달된다.

```
int a1, a2, a3, a4;
```

```
a1=1;
```

```
a2=2;
```

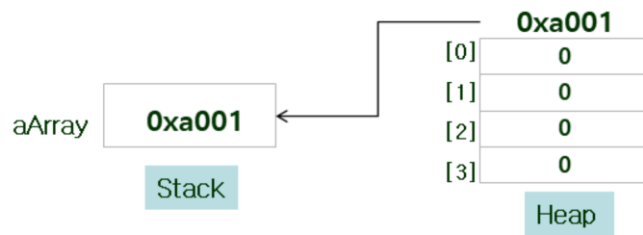
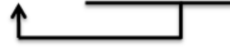
```
a3=3;
```

```
a4=4;
```



```
int [] aArray;
```

```
aArray = new int[4];
```



5-5

배열 구조

- 배열은 반복문을 사용하여 한꺼번에 초기화할 수 있다.

```
int a1, a2, a3, a4;
```

```
a1=1;
```

```
a2=2;
```

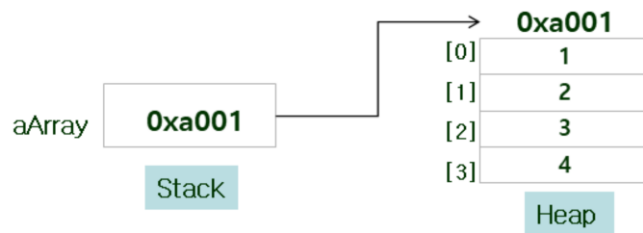
```
a3=3;
```

```
a4=4;
```



```
int [] aArray;  
aArray=new int[4];
```

```
for(int i=0; i<4; i++){  
    aArray[i]=i+1;  
}
```



5-6

배열의 정의

- 배열의 종류
 - 기본형 배열

```
int [] numbers; // array of ints
```

- 객체형 배열

```
Student [] studentList; //array of Students
```

1차열 배열

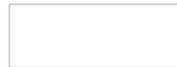
- 배열은 객체임으로 선언, 생성, 초기화 순으로 정의된다.
- 배열 선언은 클래스가 없으므로 []로 대신한다.

■ 선언

`int [] 변수명 or int 변수명 []`

ex) `int [] aArray`

aArray



Stack

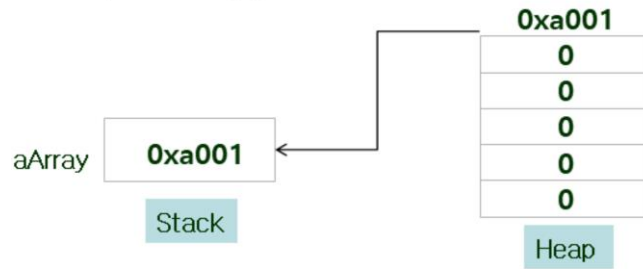
1차열 배열

- 배열 생성시 new 키워드 뒤에서 [배열 크기]를 지정한다.

- 생성

aArray=new int[length];

ex) aArray=new int[5]



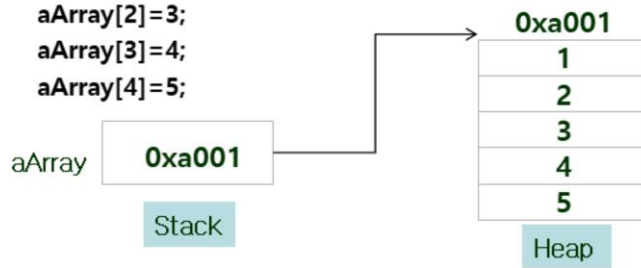
1차열 배열

- 배열의 초기화가 안될 경우 배열 자료형의 기본값이 설정된다.

- 초기화

```
aArray[0]=1; ...  
aArray[length-1]=5;
```

```
ex) aArray[0]=1;  
     aArray[1]=2;  
     aArray[2]=3;  
     aArray[3]=4;  
     aArray[4]=5;
```



5-10

1차열 배열

- 배열의 선언과 생성

```
int [] aArray = new int[length];
```

- 배열의 선언과 생성, 초기화

```
int [] aArray = {1, 2, 3, 4, 5};
```

- 배열은 클래스는 없지만 배열의 길이를 리턴하는 속성 제공
aArray.length

```
for(int i=0; i<aArray.length; i++){  
    aArray[i];  
}
```

1차열 배열

- 실습)

```
package com.oopsw.basic;
public class ArrayTest_1 {
    public static void main(String[] args) {

        int [] num;
        num=new int [6];
        num[0]=10;
        num[1]=20;

        for(int i=0; i<num.length; i++){
            System.out.println(num[i]);
        }
    }
}
```

5-12

1차열 배열

- 실습)

```
package com.oopsw.basic;
public class ArrayTest_2 {
    public static void main(String[] args) {
        //Array of Strings
        String [] names = {"전", "혜", "영"};

        for(int i=0; i<names.length; i++){
            System.out.println(names[i]);
        }
    }
}
```

5-13

다차열 배열

- 선언

`int[][] numbers` or `int numbers [][]`

- 생성

`numbers = new int[2][3]` or `numbers = new int[2][]`

- 초기화

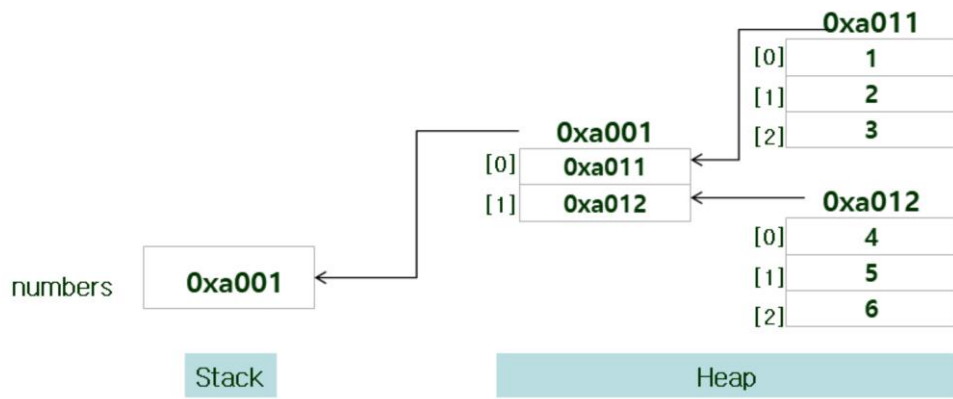
`numbers[0][0] = 1`

...

`numbers[1][2] = 6`

다차열 배열

- 다차열 배열의 메모리



5-15

다차열 배열

- 실습)

```
int[][] numbers = new int[3][3];
numbers[0][0]=1;
numbers[1][0]=2;
numbers[1][1]=2;
numbers[2][0]=3;
numbers[2][1]=3;
numbers[2][2]=3;
for(int i=0; i<3; i++){
    for(int j=0; j<3; j++){
        System.out.print(numbers[i][j] + " ");
    }
    System.out.println();
}
```


다차열 배열

- 실습)

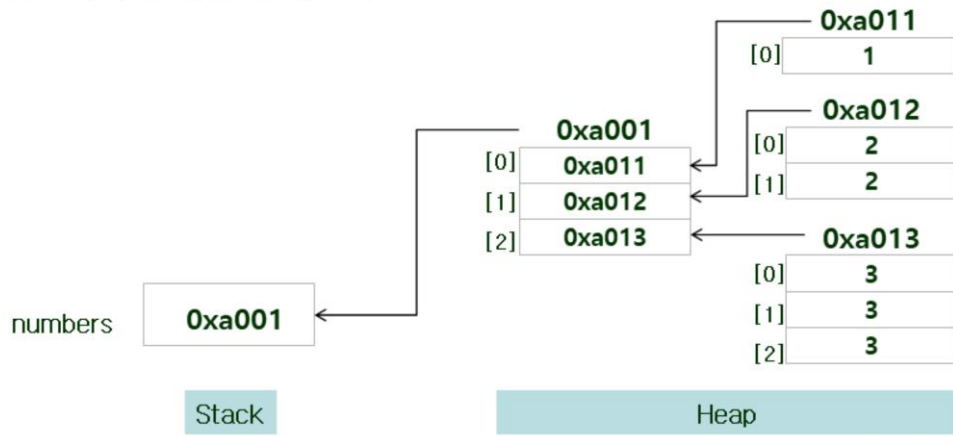
```
int[][] numbers = new int[3][];
numbers[0]=new int[1];
numbers[1]=new int[2];
numbers[2]=new int[3];

for(int i=0; i<numbers.length; i++){
    for(int j=0; j<numbers[i].length; j++){
        numbers[i][j]=i+1;
        System.out.print(numbers[i][j]+"wt");
    }
    System.out.println();
} //i
```

5-17

다차원 배열

- 다차원 배열의 메모리



System.arraycopy()

- 배열은 new 로 생성될 때 정의된 배열 사이즈를 고정
- 배열 크기를 변경할 수 없다.
- 굳이 변경하려면 새로운 배열을 생성 후 기존 배열 데이터를 새로운 배열에 복사한다.
- System.arraycopy() 활용

System.arraycopy()

- 실습)

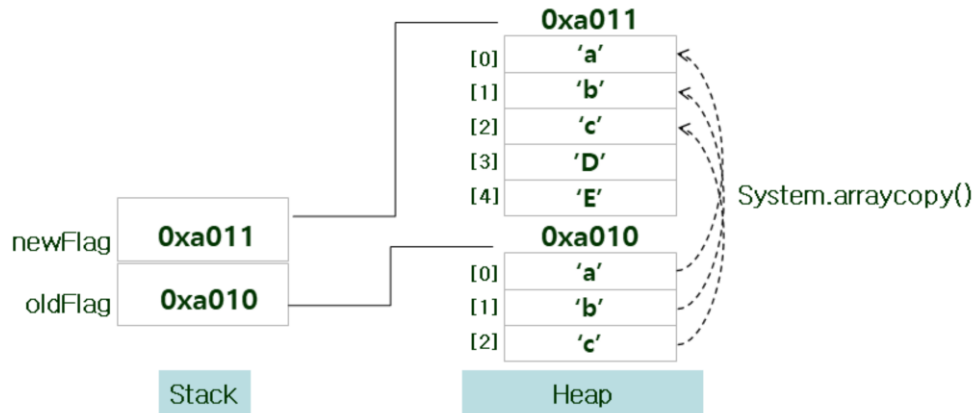
```
public static void main(String[] args)
{
    char [] oldFlag = {'a', 'b', 'c'};
    char [] newFlag = {'A', 'B', 'C', 'D', 'F'};

    System.out.println(oldFlag);
    System.out.println(newFlag);
    System.arraycopy(oldFlag, 0, newFlag, 0, oldFlag.length);
    System.out.println(newFlag);
}
```

5-20

System.arraycopy()

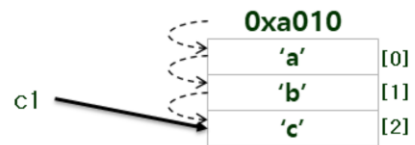
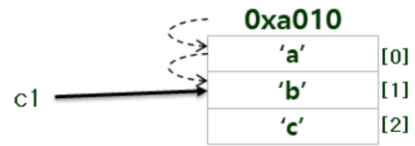
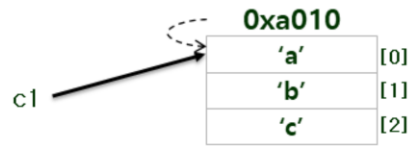
- 메모리



J2SE 5.0

● 배열 처리

```
char [] flag ={'a', 'b', 'c'};  
//기존방식  
for(int i=0; i<flag.length; i++){  
    System.out.println(flag[i]);  
}  
  
//JavaSE5  
for(char c1 : flag){  
    System.out.println(c1);  
}
```



enum

- 데이터중 한정된 값만 갖고 싶을 때
 - 상태 정보 : WAIT, RUN, STOP, CLOSE
 - 학생 종류 : 초등학생, 중학생, 고등학생, 대학생, 대학원생
 - 크기 : S, M, L
 - 사이즈 : 55, 66, 77
- 열거형 값은 상수로 취급함으로 대문자 권장

enum

- **Calendar 객체**
`Calendar c1 = Calendar.getInstance();`
`Int year = c1.get(Calendar.YEAR)`
- **사용자 정의 열거형 객체**

UserState.java

```
package com.oopsw.basic;  
public enum UserState {  
    WAIT,  
    RUN,  
    STOP,  
    CLOSE  
}  
UserState carState;  
carState.WAIT
```

5-24

Scanner

- 입력
 - 변수선언
 - 매개인자
 - JVM's `main(String [] args)`
 - Scanner객체 활용

LAB