1. 배열

1. int a1, int a2, int a3;

2. int a1, ..., int a100;

3. 배열은 같은 타입의 값들을 모아서 하나의 변수처럼 사용할 수 있는 자료형.

4. 배열은 순차적인 순번(인덱스)이 있고 지정된 같은 타입의 값만 저장할 수 있다.

5. 배열의 선언

- 자료형(타입)[] 배열명 = new 자료형[배열의 길이];

```
int[] intArr = new int[15];
```

- 컴퓨터는 0부터 센다. 인덱스는 0부터 시작. 마지막 인덱스는 배열의 길이 - 1.

6. 배열의 초기화

- 각 인덱스에 직접 접근해서 값을 입력해주는 방식

```
intArr[0] = 10;
intArr[1] = 20;
...
intArr[14] = 140;
```

- 반복문을 이용해서 초기화

```
for(int i = 0; i < intArr.length; i++) {
    intArr[i] = (i + 1) * 10;
}
```

- 배열을 생성할 때 동시에 초기화

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

7. 배열의 길이

- 배열의 길이 length 배열의 인스턴스 변수로 확인할 수 있다.
- String의 length는 메소드로 length()로 사용하지만 배열에서는 length 필드라서 괄호()없이 사용.

```
//String 클래스에 길이를 구하는 length라는 메소드로 구현되어 있다
String str = "aaa";
//문자열의 길이는 length() 메소드를 이용해서 구한다.
System.out.println(str.length());

//배열은 length라는 변수가 선언되어 있어 배열을 선언할 때 지정한
//length에 저장된다.
int[] a = new int[10];
//지정한 길이 10이 a의 length 변수에 저장된다.
System.out.println(a.length);

//List는 size() 메소드로 크기를 구한다.
List<String> strList = new ArrayList<String>();
System.out.println(strList.size());
```

- **예제 : chap05_array._01_createArray.java**
- **예제 : chap05_array._02_arrayExample.java**
- **예제 : chap05_array._03_arrayExample.java**
- **예제 : chap05_array._04_arrayExample.java**

8. 향상된 for

- 배열로 선언된 자료형을 통해 배열에 요소 하나씩 접근할 수 있는 반복문. 지정한 변수로 인덱스 순서대로 값을 하나씩 꺼내서 사용한다.

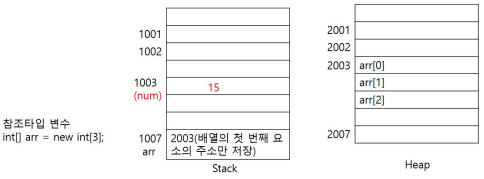
```
int[] intArr = {1,2,3,4,5};
for(int a : intArr) {
    System.out.println(a);
}
```

- **예제 : chap05_array._05_advancedFor.java**

2. 배열의 복사

1. 배열도 참조 타입의 변수라 heap메모리에 값이 저장.

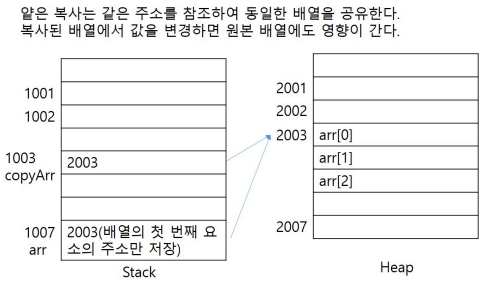
2.



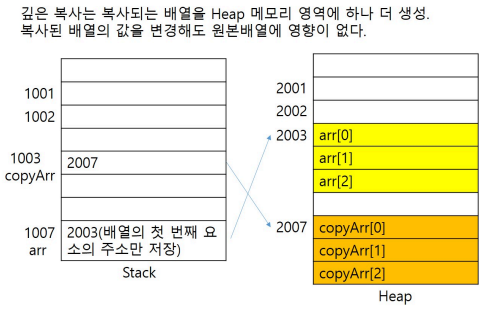
3. heap메모리에 메모리 한 칸당 하나씩 배열의 요소가 저장되고 stack에는 배열의 첫 번째 요소의 주소값이 저장된다.

4. 배열의 복사는 얕은 복사와 깊은 복사로 나눌 수 있다. 얕은 복사는 heap메모리에 생성된 배열을 함께 사용하는 복사이고 깊은 복사는 heap메모리에 생성된 배열을 heap메모리에 하나 더 생성해주는 복사.

5.



6.



7. 얕은 복사 방법

- 대입연산자를 이용해서 복사될 배열에 원본 배열을 대입한다.

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

- **예제 : chap05_array._06_shallowCopy.java**

8. 깊은 복사 방법

- clone 메소드 사용. clone 메소드 사용시 새로운 배열을 메모리에 저장하고 저장된 배열을 전달.

```
int[] c = a.clone();
```

- System.arraycopy() 메소드 사용.
- arraycopy(원본배열, 원복배열에서 복사를 시작하는 위치의 인덱스, 복사된 배열을 받을 배열, 복사된 배열에서 값을 넣을 위치의 인덱스, 복사할 배열의 길이)
- **예제 : chap05_array._07_deepCopy.java**