

객체 배열

▶ 객체배열 → 같은 클래스의 여러 객체를 하나로 묶어 관리한다.

객체를 저장하는 배열로 배열의 자료형을 클래스명(사용자 정의 자료형)으로 지정하여 활용

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

intArr	0	0	0	0	0
--------	---	---	---	---	---

```
Example[] exArr = new Example[5];
```

exArr	Example 객체	Example 객체	Example 객체	Example 객체	Example 객체
-------	---------------	---------------	---------------	---------------	---------------

▶ 객체배열 선언과 할당

✓ 선언

클래스명[] 배열명;

클래스명 배열명[] ;

ex) Academy[] arr;
Academy arr[];

✓ 할당

배열명 = new 클래스명[배열크기] ;

ex) arr = new Academy[5];

✓ 선언과 동시에 할당

클래스명 배열명[] = new 클래스명[배열크기];

ex) Academy[] arr = new Academy[5];



▶ 객체배열 초기화

✓ 인덱스를 이용한 초기화

배열명[i] = new 클래스명();

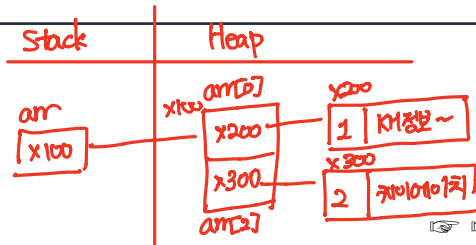
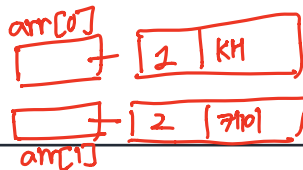
ex) arr[0] = new Academy(1, "KH정보교육원");
arr[1] = new Academy(2, "케이에이치");



✓ 선언과 동시에 할당 및 초기화

클래스명 배열명[]
= {new 클래스명(), new 클래스명()};

ex) Academy arr[] = {
new Academy(1, "KH정보교육원"),
new Academy(2, "케이에이치") };

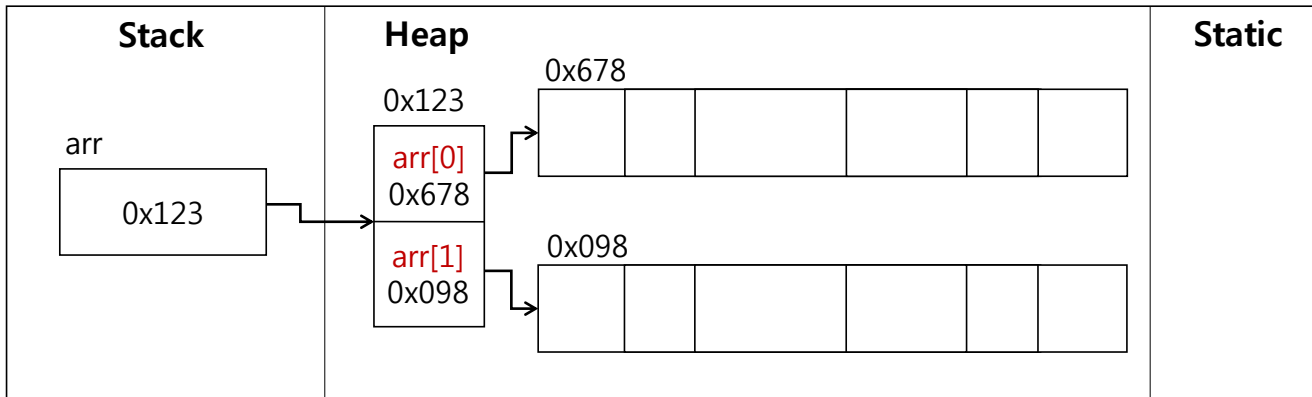


▶ 객체배열 구조

Example `arr[] = new Example[2];`

`arr[0] = new Example();` // Example 객체를 생성해서 주소값을 넣는다

`arr[1] = new Example();`



* `arr[index]`이 하나의 참조형 변수라고 생각

MTest (객체 배열 실행, 객체 배열을 사용하지 않은 코드) / Book (테스트 클래스)

```

1 package com.test01.model;
2
3 public class Book {
4
5     //필드
6     private String title;
7     private String writer;
8     private String publisher;
9     private int price;
10
11     //생성자
12     //기본생성자
13     public Book() {}
14
15     //매개변수 생성자
16     public Book(String title, String writer, String publisher, int price) {
17         this.title = title;
18         this.writer = writer;
19         this.publisher = publisher;
20         this.price = price;
21     }
22
23     //getter&setter
24     public String getTitle() {
25         return title;
26     }
27
28     public void setTitle(String title) {
29         this.title = title;
30     }
31
32     public String getWriter() {
33         return writer;
34     }
35
36     public void setWriter(String writer) {
37         this.writer = writer;
38     }
39
40     public String getPublisher() {
41         return publisher;
42     }
43
44     public int getPrice() {
45         return price;
46     }
47
48     public void setPrice(int price) {
49         this.price = price;
50     }
51
52     //한번에 보기쉽게 필드값들을 보여주는 출력 메소드
53     public String prn() {
54         return "title="+title+", writer="+writer+", publisher="+publisher+", price="+price;
55     }
56
57 }

```

```

1 package com.test01.run;
2
3 import java.util.Scanner;
4 import com.test01.model.Book;
5
6 public class MTest {
7
8     public static void main(String[] args) {
9
10         //객체배열을 사용하지 않은 코드
11         Book bk1 = new Book("자바의 정석", "홍길동", "나무", 10000);
12         Book bk2 = new Book("C언어의 정석", "이창진", "KH", 20000);
13         Book bk3 = new Book("프로그래밍", "나나나", "에스클래스", 30000);
14
15         //출력
16         System.out.println(bk1.prn());
17         System.out.println(bk2.prn());
18         System.out.println(bk3.prn());
19
20         //도서관색
21         Scanner sc = new Scanner(System.in);
22         System.out.print("검색할 책 제목: ");
23         String searchTitle = sc.nextLine();
24
25         if(bk1.getTitle().equals(searchTitle)) {
26             System.out.println(bk1.prn());
27         } else if(bk2.getTitle().equals(searchTitle)) {
28             System.out.println(bk2.prn());
29         } else if(bk3.getTitle().equals(searchTitle)) {
30             System.out.println(bk3.prn());
31         }
32
33     }
34
35 }

```

```

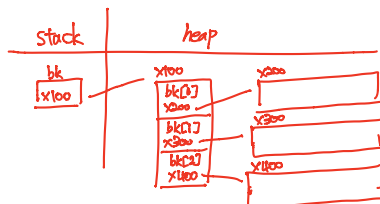
<terminated> MTest [Java Application] C:\Program Files\Amazon Corretto\jdk1.8.0_282\bin\javaw.exe (2021. 3. 12 오전 12:1
title=자바의 정석, writer=홍길동, publisher=나무, price=10000
title=C언어의 정석, writer=이창진, publisher=KH, price=20000
title=프로그래밍, writer=나나나, publisher=에스클래스, price=30000
검색할 책 제목: 자바의 정석
title=자바의 정석, writer=홍길동, publisher=나무, price=10000

```



Object Array Test (객체 배열 사용)

```
1 package com.test01.run;
2
3 import java.util.Scanner;
4
5
6 public class ObjectArrayTest {
7
8     public static void main(String[] args) {
9
10         //1. 기본자료형 배열
11         int[] arr = new int[3];
12         arr[0] = 1;
13         arr[1] = 2;
14         arr[2] = 3;
15
16         //자동완성 단축키(ctrl+space) ex) syso -> System.out.println(); ,
17
18         System.out.println(arr); //참조형 변수이기 때문에 주소값이 들어가 있다.
19         System.out.println(arr[0]); //index가 0인값의 방에 찾아가서 값을 가지고 온다.
20
21         //2. 객체배열
22         //선언 및 할당
23         Book[] bk = new Book[3];
24
25         //초기화
26         bk[0] = new Book("자바의 정석", "홍길동", "나무", 10000);
27         bk[1] = new Book("C언어의 정석", "이창진", "KH", 20000);
28         bk[2] = new Book("프로그래밍", "나나나", "케이페이지", 30000);
29
30         System.out.println(bk);
31         System.out.println(bk[0]);
32         System.out.println(bk[0].getPrice());
33
34         //bk                =>Book[]      (주소값)
35         //bk[index]         =>Book 객체   (주소값)
36         //bk[index].getPrice() =>Book객체의 필드 (실제값)
37
38         System.out.println("도서 갯수: " + bk.length);
39
40         //도서를 출력
41         System.out.println(bk[0].prn());
42         System.out.println(bk[1].prn());
43         System.out.println(bk[2].prn());
44
45         for(int i=0; i<bk.length; i++) {
46             System.out.println(bk[i].prn());
47         }
48
49         //도서 검색
50         Scanner sc = new Scanner(System.in);
51         System.out.print("검색할 책 제목: ");
52         String searchTitle = sc.nextLine(); //책 제목에 띄어쓰기가 있기 때문에 공백까지 입력 받기 위해
53
54         for(int i=0; i<bk.length; i++) {
55             if(bk[i].getTitle().equals(searchTitle)) {
56                 System.out.println(bk[i].prn());
57             }
58         }
59     }
60 }
61
62
63
64
```



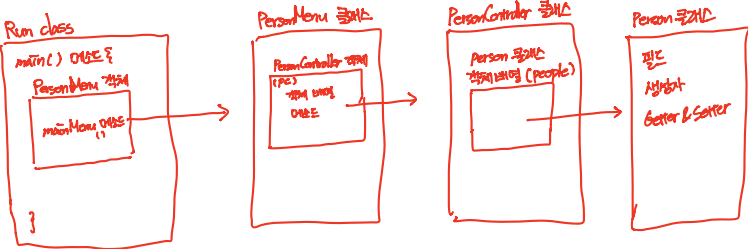
↳ 객체 배열의 양 = 객체의 수

(회원관리 프로그램 실습)

```

1 package com.test02.run;
2
3 import com.test02.view.PersonMenu;
4
5 public class Run {
6     public static void main(String[] args) {
7
8         //기본생성자 이용 mainMenu메소드 사용
9         new PersonMenu().mainMenu();
10
11     }
12
13 }
14
15 }
16

```



```

Run [Java Application] C:\Program Files\Amazon Corretto\jdk1.8.0_282\bin\javaw.exe (2021. 3. 12 오전 1:50:26)

=== 메뉴 ===
1. 회원 추가
2. 회원 전체 조회
3. 평균 재산 조회
4. 회원 이름으로 검색
9. 프로그램 종료
=====
메뉴입력: 1
=== 회원 정보 입력 ===
이름: 이창진
나이: 20
재산: 100000
=== 회원 정보 입력 ===
이름: 나나나
나이: 22
재산: 100000000
=== 메뉴 ===
1. 회원 추가
2. 회원 전체 조회
3. 평균 재산 조회
4. 회원 이름으로 검색
9. 프로그램 종료
=====
메뉴입력: 2
name=이창진, age=20, wealth=100000
name=나나나, age=22, wealth=100000000
=== 메뉴 ===
1. 회원 추가
2. 회원 전체 조회
3. 평균 재산 조회
4. 회원 이름으로 검색
9. 프로그램 종료
=====
메뉴입력: 3
평균 재산: 50050000

```

Person Menu (메뉴화면 구성)

```

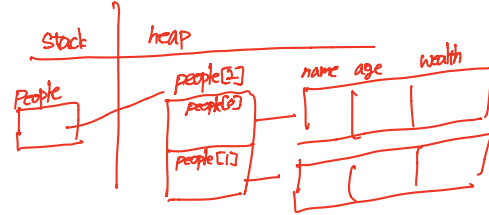
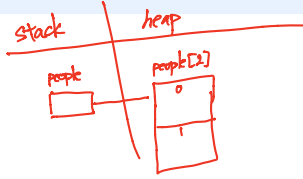
1 package com.test02.view;
2
3 import java.util.Scanner;
4
5 public class PersonMenu {
6
7     public void mainMenu() {
8
9         Scanner sc = new Scanner(System.in);
10
11         PersonController pc = new PersonController(); //PersonController 클래스 타입을 가지는 pc라는 이름의 field.
12         //pc라는 필드에 PersonController 객체를 담아준다.
13         //메뉴에서 실행할 수 있는 기능들을 사용하기 위해 객체를 만들고 객체배열도 생성된다.
14         //mainMenu메소드가 종료되기 전까지 자원(ex. 객체배열)들이 사용된다. 종료 되는 순간 사라진다.
15
16         while(true) {
17             System.out.println("=== 메뉴 ===");
18             System.out.println("1. 회원 추가");
19             System.out.println("2. 회원 전체 조회");
20             System.out.println("3. 평균 재산 조회");
21             System.out.println("4. 회원 이름으로 검색");
22             System.out.println("9. 프로그램 종료");
23             System.out.println("=====");
24             System.out.print("메뉴입력: ");
25             int menu = sc.nextInt();
26
27             switch(menu) {
28                 case 1:
29                     pc.insertPerson();
30                     break;
31                 case 2:
32                     pc.printPerson();
33                     break;
34                 case 3:
35                     pc.avgWealth();
36                     break;
37                 case 4:
38                     pc.searchPerson();
39                     break;
40                 case 9:
41                     System.out.println("프로그램을 종료합니다.");
42                     return;
43                 default:
44                     System.out.println("다시입력!!!");
45                     break;
46             }
47         }
48     }
49 }
50
51 }
52
53 }
54

```


PersonController (회원 정보 관리)

PersonMenu.java Run.java Person.java PersonController.java

```
1 package com.test02.controller;
2
3 import java.util.Scanner;
4
5 public class PersonController {
6
7     private Person[] people = new Person[2];
8
9     Scanner sc = new Scanner(System.in);
10
11     //기본생성자
12     public PersonController() {}
13
14     //메소드
15     //회원추가
16     public void insertPerson() {
17
18         for(int i=0; i<people.length; i++) {
19             System.out.println("=== 회원 정보 입력 ===");
20             System.out.print("이름: ");
21             String name = sc.nextLine();
22             System.out.print("나이: ");
23             int age = sc.nextInt();
24             System.out.print("재산: ");
25             int wealth = sc.nextInt();
26             sc.nextLine();
27
28             people[i] = new Person(name, age, wealth);
29
30         }
31
32     //회원 전체 조회
33     public void printPerson() {
34
35         for(int i=0; i<people.length; i++) {
36             System.out.println(people[i].info());
37         }
38
39     //평균 재산 조회
40     public void avgWealth() {
41         //people배열에 저장되어 있는 people 객체들의 wealth 평균값 구하여 출력
42
43         int sum = 0;
44
45         for(int i=0; i<people.length; i++) {
46             sum += people[i].getWealth();
47         }
48
49         System.out.println("평균 재산: " + (sum/people.length));
50
51     }
52
53     //회원 이름으로 검색
54     public void searchPerson() {
55         //이름을 입력받아 people배열에 저장되어있는 회원을 찾아 출력
56
57         System.out.print("검색할 이름: ");
58         String searchName = sc.nextLine();
59
60         int cnt = 0;
61
62         for(int i=0; i<people.length; i++) {
63             if(people[i].getName().equals(searchName)) {
64                 System.out.println(people[i].info());
65                 cnt++;
66             }
67         }
68
69         if(cnt==0) {
70             System.out.println("회원 없음");
71         }
72     }
73 }
74
75
76
77
78
79
80
81
82
83 }
```



Person (VO , dto 클래스)

```
1 package com.test02.model.dto;
2
3 public class Person {
4
5     //필드
6     private String name;
7     private int age;
8     private int wealth;
9
10    //생성자
11    public Person() {}
12
13    public Person(String name, int age, int wealth) {
14        this.name = name;
15        this.age = age;
16        this.wealth = wealth;
17    }
18
19    //getter&setter
20    public String getName() {
21        return name;
22    }
23
24    public void setName(String name) {
25        this.name = name;
26    }
27
28    public int getAge() {
29        return age;
30    }
31
32    public void setAge(int age) {
33        this.age = age;
34    }
35
36    public int getWealth() {
37        return wealth;
38    }
39
40    public void setWealth(int wealth) {
41        this.wealth = wealth;
42    }
43
44
45    //출력 메소드
46    public String info() {
47        return "name="+name+", age="+age+", wealth="+wealth;
48    }
49
50 }
```