

# 배열과 객체지향

## 배열

배열의 선언과 생성

배열 복사

배열 정렬

1차원

2차원

문제 1

문제2

내 풀이

강사님 풀이

## 객체 지향

문제 1

문제 2

문제 3

## 복습

## 배열

### 배열의 선언과 생성

- 크기에 제약이 있다.

```
int[] a; // 배열의 선언
a = new int[100]; // 배열의 생성

int[] a = new int[100]; // 배열의 선언과 생성 동시에
int[] months = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
```

- 여기서 a는 참조형 변수
  - 주소값을 가지고 있다.

### 배열 복사

```
int[] arr = { 1, 2, 3 };
int[] brr = arr;
System.out.println(brr[0]); // 1

brr[0] = 10;
System.out.println(arr[0]); // 10
```

- 주소값에 의해 brr과 arr이 같은 값을 참조하게 된다.

# 배열 정렬

## 1차원

```
int[] crr = { 3, 2, 4 };
System.out.println(Arrays.toString(crr)); // [3, 2, 4]
Arrays.sort(crr); // 오름차순 정렬
System.out.println(Arrays.toString(crr)); // [2, 3, 4]

for (int i = 0; i < crr.length; i++) {
    for (int j = 0; j < crr.length - 1; j++) {
        int tmp = 0;
        if (crr[j] < crr[j + 1]) {
            tmp = crr[j + 1];
            crr[j + 1] = crr[j];
            crr[j] = tmp;
        }
    }
}
System.out.println(Arrays.toString(crr));

// 어이없네.. 난 열심히 바꾸기까지 했는데 그냥 거꾸로 출력하래..
for(int i = 0; i < crr.length; i++)
    System.out.println(crr[crr.length-i-1]);
```

## 2차원

```
int[][] drr = { { 2, 90 }, { 1, 60 }, { 3, 20 } };
Arrays.sort(drr, new Comparator<int[]>() {

    @Override
    public int compare(int[] o1, int[] o2) {
        // 정렬 기준 작성
        // 1차원 배열 안 첫 번째 요소 기준 오름차순
        //     if (o1[0] < o2[0]) {
        //         return -1;
        //     } else if (o1[0] > o2[0]) {
        //         return 1;
        //     } else {
        //         return 0;
        //     }
        return o1[0] - o2[0];
    }
});

System.out.println(Arrays.deepToString(drr));
```

## 문제 1

- 여러 명의 성적을 관리할 수 있는 프로그램을 구현해 보자.
- 키보드로부터 각 과목별로 점수를 입력받아 구현하자.

```
public class Mission02 {
    public static void main(String[] args) {
        int[][] score = { { 1, 50, 34, 23 }, { 2, 56, 77, 89 } };
    }
}
```

```

int[][] ans = new int[2][6];
System.out.println("번호 국어 영어 수학 총점 평균");
for(int i = 0; i < ans.length; i++) {
    for(int j = 0 ; j < 4; j++) {
        ans[i][j] = score[i][j];
    }
}
for(int i = 0; i < ans.length; i++) {
    for(int j = 0; j < 4; j++) {
        ans[i][4] += score[i][j];
    }
}
ans[1][5] = ans[1][4] / 3;
ans[0][5] = ans[0][4] / 3;

for(int i = 0; i < ans.length; i++) {
    for(int j = 0; j < ans[0].length; j++) {
        System.out.print(ans[i][j] + "   ");
    }
    System.out.println();
}
}
}

```

- 이게 맞아.....?

## 문제2

- 배열에 문자열을 추가, 출력, 검색, 수정, 삭제 가능하도록 하기.

## 내 풀이

```

import java.util.Scanner;

public class Mission03 {
    public static void main(String[] args) {
        // 배열에 문자열을 추가, 출력, 검색, 수정, 삭제
        Scanner sc = new Scanner(System.in);
        String[] str = new String[1000];

        while (true) {
            System.out.println("1. 추가  2. 출력  3. 검색  4. 수정  5. 삭제  6. 종료");
            System.out.println("메뉴 입력: ");
            String menu = sc.nextLine();
            // 사례가 몇 개 등장하기 때문에 case 사용
            outer: switch (menu) {
                case "1":
                    System.out.println("문자열을 추가합니다.");
                    System.out.println("추가할 문자열을 입력해 주세요.: ");
                    String add = sc.nextLine();
                    for (int i = 0; i < str.length; i++) {
                        if (str[i] == null) {
                            str[i] = add;
                            break outer;
                        } else {
                            continue;
                        }
                    }
                    break;
                case "2":
                    System.out.println("문자열을 출력합니다.");

```

```

        for (int i = 0; i < str.length; i++) {
            if (str[i] != null)
                System.out.println(str[i] + " ");
        }
        break;

    case "3":
        System.out.println("문자열을 검색합니다.");
        System.out.println("검색할 문자열을 입력해 주세요.: ");
        String keyword = sc.nextLine();
        for (int i = 0; i < str.length; i++) {
            if (str[i].contains(keyword)) {
                System.out.println(str[i]);
                break outer;
            }
        }
        break;

    case "4":
        System.out.println("문자열을 수정합니다.");
        System.out.println("수정할 문자열의 번호를 입력해 주세요.");
        int number = Integer.parseInt(sc.nextLine());
        System.out.println("수정하고 싶은 내용을 입력해 주세요.");
        String edit = sc.nextLine();
        str[number - 1] = edit;
        break;

    case "5":
        System.out.println("문자열을 수정합니다.");
        System.out.println("삭제할 문자열의 번호를 입력해 주세요.");
        int delnum = Integer.parseInt(sc.nextLine());
        for (int i = delnum; i < str.length; i++) {
            str[delnum - 1] = str[delnum];
            if (str[i] == null) {
                str[i+1] = "";
                break outer;
            }
        }
        break;

    case "6":
        System.out.println("프로그램 종료");
        return;
    }
}
}
}
}

```

## 강사님 풀이

```

import java.util.Scanner;

public class Mission03_ans {
    // static 변수로 정의
    static int count = 0;
    static Scanner sc = new Scanner(System.in);

    // 메서드로 따로 쓰기
    // 1. 추가
    public static void add(String[] str) {
        System.out.println("이름: ");
    }
}

```

```

    str[count++] = sc.nextLine();
}

// 2. 출력
public static void print(String[] str) {
    for (int i = 0; i < count; i++) {
        System.out.print(str[i] + " ");
    }
    System.out.println();
}

// 3. 검색
public static void search(String[] str) {
    System.out.print("검색할 이름: ");
    String name = sc.nextLine();
    int idx = -1;

    for (int i = 0; i < count; i++) {
        if (name.equals(str[i])) {
            idx = i;
            break;
        }
    }
    if (idx == -1)
        System.out.println("해당 이름이 없습니다.");
    else
        System.out.println((idx + 1) + "번에 있습니다.");
}

// 4. 수정 => 찾은 후에 고치기
public static void update(String[] str) {
    System.out.print("수정할 이름: ");
    String name = sc.nextLine();
    int idx = -1;

    for (int i = 0; i < count; i++) {
        if (name.equals(str[i])) {
            idx = i;
            break;
        }
    }
    if (idx == -1)
        System.out.println("해당 이름이 없습니다.");
    else {
        System.out.print("수정할 이름: ");
        str[idx] = sc.nextLine();
    }
}

// 5. 삭제 => 없앤 값 앞으로 땡겨 주기 (접근법은 똑같았다 ㅎㅎ)
public static void delete(String[] str) {
    System.out.print("삭제할 이름: ");
    String name = sc.nextLine();
    int idx = -1;

    for (int i = 0; i < count; i++) {
        if (name.equals(str[i])) {
            idx = i;
            break;
        }
    }
    if (idx == -1)
        System.out.println("해당 이름이 없습니다.");
    else {
        for(int i = idx; i < count; i++) {
            str[i] = str[i+1];

```

```

    }
    count--;
}
}

public static void main(String[] args) {
    // 배열에 문자열을 추가, 출력, 검색, 수정, 삭제

    String[] str = new String[1000];

    while (true) {
        System.out.println("1. 추가  2. 출력  3. 검색  4. 수정  5. 삭제  6. 종료");
        System.out.println("메뉴 입력: ");
        String menu = sc.nextLine();
        // 사례가 몇 개 등장하기 때문에 case 사용
        switch (menu) {
            case "1":
                add(str);
                break;

            case "2":
                print(str);
                break;

            case "3":
                search(str);
                break;

            case "4":
                update(str);
                break;

            case "5":
                delete(str);
                break;

            case "6":
                System.out.println("프로그램 종료");
                return;
        }
    }
}
}

```

## 객체 지향

- 객체 지향 프로그래밍: 현실 세계를 모방해서 만든 것
- 키워드: **확장성**
  - 상속 관계, 인터페이스 등을 통해 새로운 소프트웨어 창출 가능
- 클래스: 객체들이 가진 **공통된** 데이터 속성과 기능들로 만드는 것
  - 상태(속성) + 행위(메서드)
- 인스턴스: 클래스로부터 생성된 객체
- 속성(Attribute)

- 객체가 가지는 변수, 객체의 상태
- 메소드(Method)
  - 데이터를 조작하는 행위
  - 메소드를 호출해 객체의 상태를 변경하거나 내부의 상태 값 가져옴

## 문제 1

- 계좌 객체 생성
- 멤버 변수: 계좌 번호, 계좌 이름, 잔액
- 기능: 출금, 조회

```
package kosa.oop;

public class Account {
    // 계좌 객체

    // 상태(필드): 계좌번호, 계좌주, 잔액 => 멤버 변수
    // 캡슐화 통한 정보 은닉
    private String accountNo;
    private String ownerName;
    private int balance;

    // 기능(메서드): 입금하다, 출금하다
    public void deposit(int amount) {
        balance += amount;
    }

    // 기본 생성자 만들어 주는 습관 들이기~
    public Account() { // 기본 생성자
    }

    // 새로운 객체가 생성될 때마다 생성자 호출
    public Account(String accountNo, String ownerName, int balance) {
        super();
        this.accountNo = accountNo;
        this.ownerName = ownerName;
        this.balance = balance;
    }

    public int withdraw(int amount) {
        if (balance < amount)
            return 0;
        balance -= amount;
        return amount;
    }

    public void printAccount() {
        System.out.println("계좌 번호: " + accountNo);
        System.out.println("계좌 주인: " + ownerName);
        System.out.println("잔액: " + balance);
    }

    public String getAccountNo() {
        return accountNo;
    }

    public void setAccountNo(String accountNo) {
```

```

        this.accountNo = accountNo;
    }

    public String getOwnerName() {
        return ownerName;
    }

    public void setOwnerName(String ownerName) {
        this.ownerName = ownerName;
    }

    public int getBalance() {
        return balance;
    }

    public void setBalance(int balance) {
        this.balance = balance;
    }
}

```

```

package kosa.oop;

public class AccountMain {

    public static void main(String[] args) {
        Account ac = new Account("1111-1111", "홍길동", 20000);
        Account ac2 = new Account("2222-2222", "홍길동", 50000);

        ac.withdraw(3000);
        ac.printAccount();
        // 계좌번호: 1111-1111
        // 계좌 주인: 홍길동
        // 잔액: 17000

        ac2.setAccountNo("3333-3333");
        ac2.getBalance();
        ac2.printAccount();
        // 계좌번호: 3333-3333
        // 계좌 주인: 김길동
        // 잔액: 50000
    }
}

```

## 문제 2

- 휴대 전화 객체 생성
- 멤버 변수: 이름, 전화번호, 생년월일
- 기능: 전화 내역 출력, 이름 추가, 이름 출력, 이름 검색
- phoneInfo.java

```

package kosa.phone;

public class PhoneInfo {
    // 상태: 이름, 전화번호, 생년월일(name, phoneNo, birth)
}

```



```

private String name;
private String phoneNo;
private int birth;

// 기본 생성자
public PhoneInfo() {
}

public PhoneInfo(String name, String phoneNo, int birth) {
    super();
    this.name = name;
    this.phoneNo = phoneNo;
    this.birth = birth;
}

// 기능: 1인 전화 내역 출력(show)
public void show() {
    System.out.println("이름: " + name);
    System.out.println("번호: " + phoneNo);
    System.out.println("생일: " + birth);
    System.out.println();
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getPhoneNo() {
    return phoneNo;
}

public void setPhoneNo(String phoneNo) {
    this.phoneNo = phoneNo;
}

public int getBirth() {
    return birth;
}

public void setBirth(int birth) {
    this.birth = birth;
}

}

```

- Manager.java

```

package kosa.phone;

import java.util.Scanner;

public class Manager {
    // 안에 PhoneInfo 객체가 들어간다 ㅎㅎ
    PhoneInfo[] arr = new PhoneInfo[10];
    static Scanner sc = new Scanner(System.in);
    // 멤버 변수는 초기화할 필요가 없다, 리터럴 값으로 기본 값을 가지고 있다
    int idx;
}

```

```

// 1. 추가
public void addPhoneInfo() {
    System.out.println("이름: ");
    String name = sc.nextLine();
    System.out.println("번호: ");
    String phoneNo = sc.nextLine();
    System.out.println("생년월일: ");
    int birth = Integer.parseInt(sc.nextLine());

    arr[idx++] = new PhoneInfo(name, phoneNo, birth);
}

// 2. 출력
public void listPhoneInfo() {
    for (int i = 0; i < idx; i++) {
        arr[i].show();
    }
}

// 3. 검색
public void findPhoneInfo() {
    System.out.println("검색할 이름: ");
    String name = sc.nextLine();
    int cnt = -1;
    for (int i = 0; i < idx; i++) {
        if (arr[i].getName().equals(name)) {
            arr[i].show();
            cnt = i;
            break;
        }
    }
    if (cnt == -1)
        System.out.println("해당 이름이 전화번호부에 없습니다.");
    else
        System.out.println("전화번호부 " + (cnt + 1) + "번에 있습니다.");
}
}

```

- Main.java

```

package kosa.phone;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        // 1. 추가 2. 출력 3. 검색 4. 종료
        Manager m = new Manager();
        Scanner sc = new Scanner(System.in);

        while(true) {
            System.out.println("기능을 선택해 주세요.");
            System.out.println("1. 추가 2. 출력 3. 검색 4. 종료");
            System.out.println("번호: ");
            String num = sc.nextLine();
            // switch문 사용
            switch (num) {
                case "1":
                    m.addPhoneInfo();
                    break;
                case "2":
                    m.listPhoneInfo();

```

```

        break;
    case "3":
        m.findPhoneInfo();
        break;
    case "4":
        System.out.println("기능을 종료합니다.");
        return;
    }
}

}

}

```

## 문제 3

- 책 객체 만들기
- 멤버 변수: 책 이름, 정가, 할인가
- 기능: 할인 가격 산출, 책 정보 출력
  - 할인: 30000 이상 ⇒ 85%, 20000 이상 ⇒ 80%, 15000 이상 ⇒ 75%
  - 책 정보: JAVA(30000원), JPA(20000원), DB(15000원)

```

package kosa.book;

public class Book {
    // 무작정 만들지 말고 뭐가 필요한지 생각하기
    // 상태: 책 이름, 정가, 할인된 가격
    // 기능: 할인된 가격 산출, 책 정보 출력
    // 같은 클래스 내에서는 private 접근 가능
    private String name;
    private int price;
    // 이건 계산해 줄 값이기 때문에 초기화하지 않는다
    private int salePrice;

    Book() {
    }

    // 초기화
    public Book(String name, int price) {
        super();
        this.name = name;
        this.price = price;
    }

    // 할인 가격 찾기
    public void discount() {
        // 25% 할인
        if(price >= 30000) {
            // "괄호 친 값"을 int로 casting
            salePrice = (int)(price * 0.75);
            // 20% 할인
        } else if (20000 <= price && price < 30000) {
            salePrice = (int)(price * 0.8);
        } else if (15000 <= price && price < 20000) {
            salePrice = (int)(price * 0.85);
        }
    }
}

```

```

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public int getPrice() {
    return price;
}

public void setPrice(int price) {
    this.price = price;
}

public int getSalePrice() {
    return salePrice;
}

public void setSalePrice(int salePrice) {
    this.salePrice = salePrice;
}

// 책 정보 찾기
public void printBook() {
    System.out.println(name + " 교재 정가는 " + price + "원, " + "할인된 가격은 " + salePrice + "원입니다.");
}
}

```

```

package kosa.book;

public class BookMain {
    public static void main(String[] args) {
        // 지역 변수는 초기화 필수
        int totalPrice = 0;

        // 생성자를 이용해 넣어 준다
        Book[] bookList = {
            new Book("JAVA", 30000),
            new Book("JSP", 20000),
            new Book("Oracle", 15000)
        };

        // for~each문 사용
        for(Book b: bookList) {
            b.discount();
            b.printBook();
            totalPrice += b.getSalePrice();
        }
        System.out.println("총 가격은 " + totalPrice + "원입니다.");
    }
}

```

## 복습

### 1. 오늘 했던 거 다시 풀기

2. 교재 다시 정리
3. 숙제 ⇒ 게시판 구현
  - 속성: 글 번호, 제목, 작성자, 글 내용
  - 기능: 글쓰기 , 글 목록 보기, 글 세부 보기
4. PhoneInfo 예제, 전화번호 수정, 삭제