

1주차(3)

클래스와 객체

클래스와 객체에 앞서 먼저 배열을 알아야 하므로 배열부터 시작해볼까?

배열은 '같은 자료형 끼리 모아두는 하나의 묶음' 이다.

자료 관리를 효율적으로 하기 위해서 꼭 필요하다.

배열은 기본 자료형 뿐만 아니라 객체형으로도 만들 수 있다. 차차 배워보자

먼저 1차원 배열에 대해 알아봅시다.

//ArrayEx1클래스 생성

```
public class ArrayEx1 {  
    public static void main(String[] args) {  
        // 정수변수 선언  
        int su1 = 100;  
        int su2 = 200;  
        int su3 = 300;  
        int su4 = 400;  
  
        // 편하게 자원들을 관리하고 제어하기 위해서는  
        // 다음과 같은 배열을 생성한다.  
        // 1) 배열선언  
        int[] ar;  
  
        // 2) 배열생성  
        ar = new int[4];  
  
        //생성 후에는 값을 넣어 초기화가 필요하다.  
        //아무런 값도 넣지않으면 기본자료형은 각 자료형의 초기값,  
        //스트링형은 null이 들어간다.  
        //int[] ar = {100, 200, 300, 400};//초기화 방법1  
  
        // 3) 초기화 방법2  
        ar[0] = 100;  
        ar[1] = 200;  
        ar[2] = 300;  
        ar[3] = 400;  
  
        for(int I = 0; I < ar.length; I++){
```

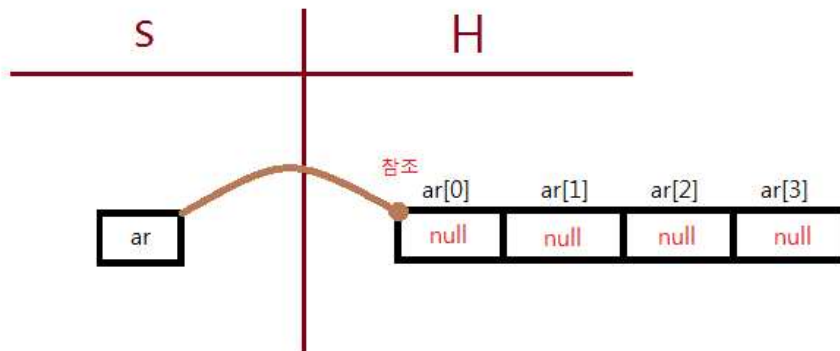
```

        System.out.println(ar[i]);
    }
}
}

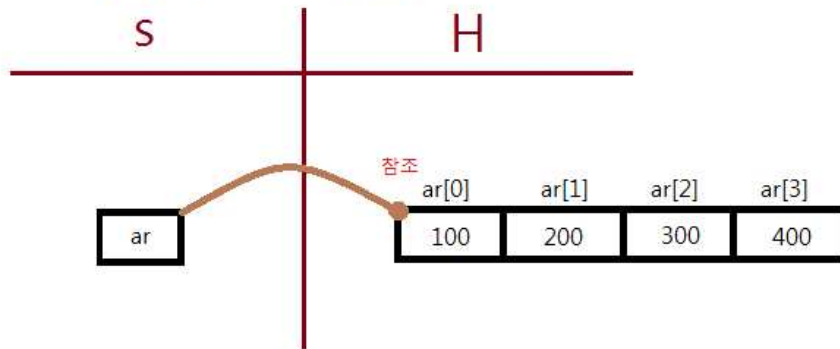
```



ar이 4개짜리 공간을 갖는 메모리 영역의 주소를 참조한다.



초기화를 통해 배열의 각 방에 값을 할당한다.



----- 예제 1

```

//ArrayEx2클래스 생성
public class ArrayEx2 {
    public static void main(String[] args) {
        char[] ch;
        ch = new char[4];
        //char[] ch = new char[4];
    }
}

```

```

//배열 초기화
ch[0] = 'J';
ch[1] = 'A';
ch[2] = 'V';
ch[3] = 'A';

//배열 내용 출력
for(int i = 0; i < ch.length; i++){
    System.out.println("ch[" + i + "] : " + ch[i]);
}

//개선된 루프.
//편리하지만, 배열의 각 요소에 대한 값의 수정과 삭제가 불가
//for(char ch2 : ch)
    //System.out.println(ch2);
}
}

```

----- 예제2

자바 강의 1주차(3) 문제(배열에 담긴 값 더하기) !!!!

자바 강의 1주차(3) 문제(배열) !!!!

문제.

1 ~ 45의 난수를 발생시켜 로또번호를 생성하는 프로그램 만들기.

```

public class MyLotto {
    public static void main(String[] args) {

        //로또번호 6개를 담을 배열 준비
        int[] lotto = new int[6];

        outer : for(int i = 0; i < lotto.length;){//나중을 위해 i++을 생략

            lotto[i] = new Random().nextInt(45) + 1;

            //중복값을 비교하는 반복문
            for(int j = 0; j < i; j++){

```

```

        if(lotto[i] == lotto[j]){
            continue outer;
        }

    } //inner For

    System.out.print(lotto[i] + " ");
    i++;
} //outer For
}
}

```

다차원 배열시작 고고

1차원 배열이 2개 모이면 2차원 배열.

1차원 배열이 3개 모이면 3차원 배열이다.

2차원 이상은 많이 쓰진 않음. 4차원 배열은 한번도 못봄

```
int test[][] = new int[2][3];
```

```
test[0][0] = 100;
```

```
test[0][1] = 200;
```

```
test[0][2] = 300;
```

```
test[1][0] = 400;
```

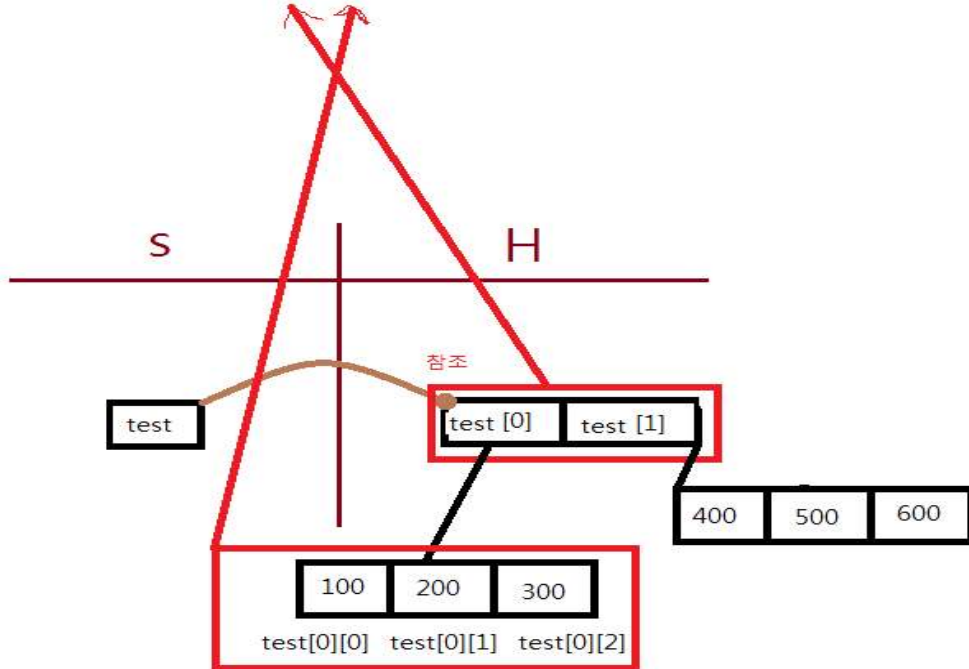
```
test[1][1] = 500;
```

```
test[1][2] = 600;
```

```
System.out.println(test[0][1]); //숫자 바뀌가며 확인
```

아래 페이지에 그림설명 있다. 설명부터 해주자

```
int test[ ][ ] = new int[ 2 ][ 3 ];
```



-----예제 1

```
String[][] java = {"영희", "java : 100", "android : 90"},
                  {"철수", "java : 80", "android : 75", "jsp : 98"}; //이런식으로 정의 가능
```

```
for(int i = 0; i < java.length; i++){

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

    System.out.println();
}
```

↓↓↓이렇게 각 방 사이즈 지정해줘도 됨

```
int num[][] = new int[2][];
num[0] = new int[3];
num[1] = new int[2];
int n = 0;
```

```
for(int i = 0; i < num.length; i++){
```

```

        for(int j = 0; j < num[i].length; j++){

            System.out.print((num[i][j] = n += 100) + " ");

        }
        System.out.println();
    }
}

```

-----예제2

자바 강의 1주차(3) 문제(다차원 배열2) 문제 출제!!!!

자바 강의 1주차(3) 문제(다차원 배열) 문제 출제!!!!

자바 강의 1주차(3) 문제(마방진) 문제 출제!!!!

배열 끝!! 드디어 클래스 시작!

클래스중에 가장 대표적이라고 할 수 있는 String클래스를 알아봅니다.

String클래스

api보기 - <http://www.oracle.com> 접속 - Downloads마우스 오버 - 왼쪽에 Java for Developers클릭 - 우측 탭에 Java Resources 아래 APIs - Java SE7클릭

왼쪽 상단 java.lang - String클래스

(아래 그림 보면서 설명)예제로 바로 확인해 봅시다

```
public static void main(String[] args)
```

```
// String은 두가지 특징이 있다.
// 1) 객체생성법이 2가지다.(암시적, 명시적)
// 2) 한번 생성된 문자열의 내용은 변하지 않는다.(불변적 특징)
```

```
String s1 = "abc"; // 암시적 객체 생성
String s2 = new String("abc"); // 명시적 객체 생성
```

```
// == 연산자는 객체를 비교할 때는 주소값 비교가 된다.
```

```
if(s1 == s2)
    System.out.println("s1과 s2의 주소가 같습니다.");
else
    System.out.println("s1과 s2의 주소가 다릅니다.");
```

```
if(s1.equals(s2)) //equals는 값 비교
    System.out.println("s1과 s2의 내용이 같습니다.");
else
    System.out.println("s1과 s2의 내용이 다릅니다.");
```

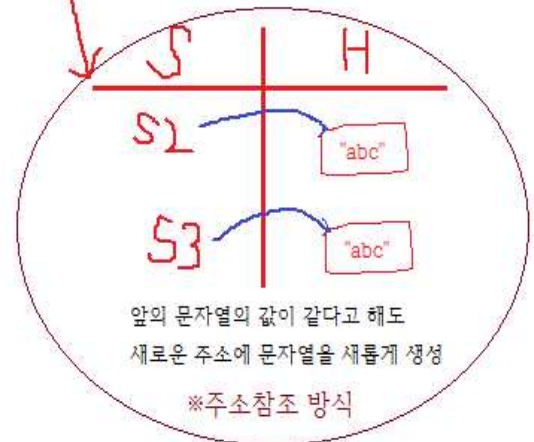
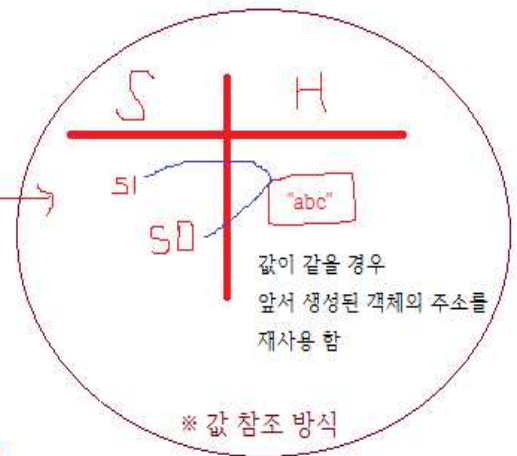
```
String s3 = "abc"; // 암시적 객체 생성
// 이미 앞서 같은 문자열 값으로 생성된 암시적 객체가 있다면
// 앞서 생성된 객체의 주소를 재 사용한다.
```

```
if(s1 == s3)
    System.out.println("s1과 s3의 주소가 같습니다.");
else
    System.out.println("s1과 s3의 주소가 다릅니다.");
```

```
// String객체의 불변적 특징을 증명하는 예제를 구현해 보자
String s4 = s1.replaceAll("b", "?");
```

```
// s1이 가리키는 곳의 문자열 값들 중에 "b"를 "?"로 변경하여
// 새로운 문자열로 생성한 후 그것을 반환한다. 그 반환 값을
// 변수 s4가 기억한다.
```

```
System.out.println("s1:"+s1);
System.out.println("s4:"+s4);
```



String클래스의 메서드들

메서드란! 어떤 작업을 수행하기 위한 명령문의 집합이다!!!
메서드를 사용하는 작성하는 가장 큰 이유는 반복적으로 사용되는 코드를 줄이기 위해서이다.
자주 사용되는 내용의 코드를 메서드로 작성해 두고 필요할때마다 호출만 하면 된다.

indexOf, lastIndexOf, charAt, substring, split정도만 설명. api보면서 하면 더 좋겠지만
융통성있게 하자

```

String str = "Kim Mal Ddong";
System.out.println("문자열 str의 길이 : " + str.length());

int index = str.indexOf('k');
System.out.println("맨 처음 문자 k의 위치 : " + index); //대소문자 구별 함

index = str.indexOf("Mal");
System.out.println("문자열 Mal의 위치 : " + index); //띄어쓰기 포함

index = str.lastIndexOf('n');
System.out.println("마지막 문자 n의 위치 : " + index);

char c = str.charAt(4);
System.out.println("추출한 문자 : " + c);

String str2 = str.substring(0, str.indexOf('M'));
System.out.println("0번째부터 M앞자리까지 글자 잘라내기 : " + str2);
System.out.println("잘라낸 str2의 길이 : " + str2.length()); //길이는 띄어쓰기 포함 1부터
증가

스트링은 아니지만 스트링으로 작성된 숫자형태의 문자열을 실제 정수로 바꿔주는 코드
String number = "1";
System.out.println(Integer.parseInt(number) + 10);

String arr[] = str.split(" "); //띄어쓰기를 기준으로 분할

for(int i = 0; i < arr.length; i++)
    System.out.println("arr[" + i + "] : " + arr[i]);

```

-----스트링클래스 끝

자바 강의 1주차(3) 문제(String) 문제 [4문제 들어있다ㄷ]출제!!!!
 자바문제 - 2차원 배열과 스트링을 이용한 배우검색 문제 출제!!!

스트링 클래스가 끝났으니 직접 클래스를 만들고 정의해봅시다.

S전자에서 컴퓨터를 생산한다고 가정하자.

예를 들어 c-1이라는 모델의 컴퓨터가 양산되어야 한다면, 공장에서 컴퓨터를 대량으로 찍어내기 위해 필요한 컴퓨터의 부품이나, 기능 등을 명시한 설계도가 있을 것.

이 설계도가 바로 클래스다.

그리고 이 설계도를 바탕으로 만들어진 컴퓨터가 바로 객체.

거의 비슷한 개념이기에 혼동하기 쉽지만 조금은 다른 내용이라는 것 정도만^^

클래스는 한번 사용되고 버리는 것이 아니라, 재사용에 용이하게 작성함으로써 프로그램 상에서 낭비되는 자원을 줄일 수 있도록 구성하는 것이 좋다.

이것은 자바의 가장 큰 특징인 '객체지향개념'과 결부되는데, 객체가 수행하는 능력이 뛰어나면 다른 곳에서도 그 객체를 사용하려는 횟수가 많아지게 되고 이런 재사용성을 목적으로 클래스에는 '속성(변수, 자료)'과 '기능(메서드, 함수)'을 정의하여 사용하게 된다.

말이 너무 어렵습니다.

예제로 확인하시죠.

//1. 컴퓨터클래스 작성

```
public class Test_ComputerClass {
```

```
    //설계도를 만드는 작업.
```

```
    //컴퓨터의 속성에는 뭐가 있는지 생각해보자.
```

```
    int hdd;//하드디스크
```

```
    int ram;//램
```

```
    float cpu;//cpu
```

```
    String color;//색상
```

```
    //컴퓨터의 정보를 반환할 메서드를 만들자
```

메서드란! 어떤 작업을 수행하기 위한 명령문의 집합이다!!!
메서드를 사용하는 작성하는 가장 큰 이유는 반복적으로 사용되는 코드를 줄이기 위해서이다.
자주 사용되는 내용의 코드를 메서드로 작성해 두고 필요할때마다 호출만 하면 된다.

```
    public void getInfo(){//메서드는 잠시후에 배운다. 일단 따라써
```

```
        System.out.println("하드 디스크 : " + hdd + "GB");
```

```
        System.out.println("램 : " + ram + "MB");
```

```
        System.out.println("cpu : " + cpu + "GHz");
```

```
        System.out.println("색상 : " + color);
```

```
    }
```

```
}
```

//잠깐!! 메서드의 구성

접근제한 반환형 메서드명

public void getInfo(파라미터(인자)){ 메서드의 영역 }

접근제한자 에는 public, protected, default, private의 네종류가 있다.

1. public : 모든 접근을 허용. 같은 프로젝트 내의 모든 객체들이 사용할 수 있도록 허용.
2. private : 현재 클래스 내에서만 사용을 허가.
3. protected : 상속관계의 객체들에만 사용을 허가.
4. default : 같은 패키지(폴더)내의 객체에만 사용을 허가(아무것도 쓰지 않으면 default)

1, 2번은 자주 쓰인다. 일단 개념만 알고 자세한 것은 코드를 작성하면서 이해하도록 하자.
3번은 거의 쓰이지 않음. 아무것도 명시하지 않으면 4번

반환형은 메서드가 처음부터 끝까지 수행을 마친 후에 반환해야 할 값이 있을 경우에 기입.

int, String, boolean등 기본자료형을 포함하여 사용자가 만든 객체로도 반환이 가능.

아무것도 반환하지 않을때는 void

역시 너무 어렵죠?? 예제를 한 개씩 해보면서 이해하도록 합시다.

메서드명은 말그대로 메서드의 이름(첫글자는 소문자로 시작한다.)

파라미터는 외부에서 해당 메서드를 통해 특정 값을 전달하고자 할 때, 그 특정 값을 받아서 처리할 수 있도록 하는 역할.

쉬운게 한 개도 없습니다. 일단 이렇게 있다 정도만 알고가자.

진행하면서 자연스럽게 알게될 수 있도록 노력!

//1. 컴퓨터클래스Main 작성

```
public class Test_ComputerMain {
    public static void main(String[] args) {

        Test_ComputerClass cp1;
        //int num; //설계도(자료형)와 제품명(변수) 이렇게 생각하면 된다.

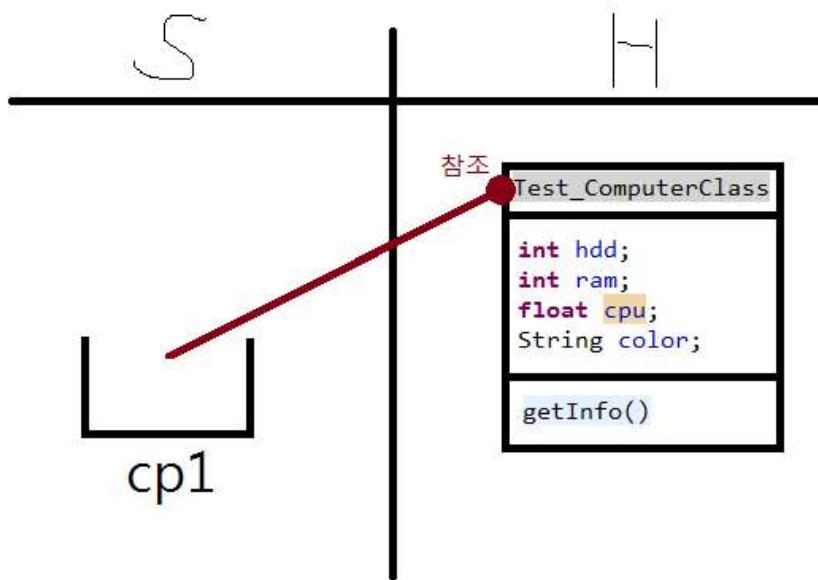
        cp1 = new Test_ComputerClass();//객체 생성
        //아래쪽에 그림으로 설명 먼저!!

        cp1.hdd = 200;
        cp1.color = "화이트";
        cp1.cpu = 2.5f;
        cp1.ram = 256;
        cp1.getInfo();

        System.out.println("-----");

        Test_ComputerClass cp2 = new Test_ComputerClass();
        cp2.hdd = 300;
        cp2.color = "블랙";
        cp2.cpu = 3.0f;
        cp2.ram = 512;
        cp2.model = "cp2";
        cp2.getInfo();

    }
}
```



이제 Test_ComputerClass의 변수중 하나의 접근제한자를 private로 바꿔보자
오류나는 이유 설명.(외부접근이 불가하기 때문)

외부에서 직접 값을 바꾸는 것이 불가능하기 때문에 보안성과 은닉성이 UP

다시 public으로 바꿔주자.

위에서 대충 설명 했지만, 본격적으로

메서드의 구성과 정의!!

일단 클래스 생성

```

public class ValueTest {
    public void test(int n){
        n++;
        System.out.println("n : " + n);
    } // 메서드의 기능이 모두 끝났으면 호출한 곳으로 돌아간다.
    // 이때 반환형이 있으면 반환값을 가지고 돌아가지만
    // 반환형이 없는 void라면 빈손으로 돌아간다. 그리고
    // 지역변수 n은 소멸된다.
}
  
```

메인클래스 생성

```

public class ValueTestMain {
    public static void main(String[] args) {
        // 변수 선언과 값대입
        int su = 100;

        // test라는 메서드를 호출하기 위해 test메서드를 가지고 있는
        // 클래스를 생성한다.
        ValueTest vt = new ValueTest(); // 명시적 객체 생성
        vt.test(su); // su에 있는 값이 복사되어 전달된다.

        System.out.println("su: " + su);
    }
}
  
```

```
}
```

----- 예제 1

Person 클래스 생성

```
public class Person {
    private String name; // 멤버변수 선언
    private String email;
    private String phone;

    public String getName() {
        return name;
    }

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

    // 전화번호를 변경하는 동작(메서드)
    public void setPhone(String n){
        // 현재 메서드는 문자열을 하나 받아서
        // 멤버변수인 phone에 저장하는 동작이다.
        this.phone = n;
    }

    //멤버변수 phone의 값을 반환하는 동작
    public String getPhone(){
        return this.phone;
    }
}
```

PersonMain 클래스 생성

```
public class PersonMain {
    public static void main(String[] args) {
        // 원하는 클래스를 객체화 시킨다.
        Person p1 = new Person();// 명시적 객체 생성

        //객체 p1으로 부터 getPhone, getName을 호출하여 반환받는 값을
        // 바로 출력하는 문장
        System.out.println(p1.getPhone());// null
        System.out.println(p1.getName());// null

        //객체 p1을 통해 정보를 저장한다.
        p1.setPhone("010-123-4567");
        p1.setName("쥬똥이");

        System.out.println(p1.getPhone());
        System.out.println(p1.getName());
    }
}
```

----- 예제 2

객체형 배열 연습

클래스 생성

```
public class ArrayTest {
    private String name;
    private int age;

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

```

        public int getAge() {
            return age;
        }
        public void setAge(int age) {
            this.age = age;
        }
    }
}

```

메인클래스 생성

```

public class ArrayMain {
    public static void main(String[] args) {
        ArrayTest at[] = new ArrayTest[2];

        /*
        for(int I = 0; I < at.length; I++){

            at[i] = new ArrayTest();

        }
        */

        at[0] = new ArrayTest(); // 이 부분이 없으면 at[]의 공간만 있고 각각
                                // ArrayTest()를 참조하지 못해 예러남
        at[1] = new ArrayTest();

        at[0].setAge(19);
        at[0].setName("말똥이");

        at[1].setAge(20);
        at[1].setName("양똥이");

        for(int i = 0; i < at.length; i++){
            System.out.println(at[i].getAge());
            System.out.println(at[i].getName());
            System.out.println("-----");
        }
    }
}

```

-----예제 3

//클래스로 계산기 만들어보기 문제.

```

/*
* 첫번째 숫자 입력 : 5
* 두번째 숫자 입력 : 10
* 연산기호 입력 : +
* 결과 : 15
*
* Scanner를 사용해
* 숫자 두 개와 연산기호를 받은 뒤 계산해주는 클래스를 만들고 실행하기
*
* 참고: String의 비교는 ==아닌 String변수.equals("비교값")으로 한다
* if(str.equals("+"))
* else if(str.equals("-"))
* else if(str.equals("*"))
* else if(str.equals("/"))
*/

```

//풀이

```

public class CalTest {

    public int getResult(int n1, int n2, String str){

```

```

        if(str.equals("+"))
            return n1 + n2;
        else if(str.equals("-"))
            return n1 - n2;
        else if(str.equals("*"))
            return n1 * n2;
        else if(str.equals("/"))
            return n1 / n2;
        else{
            System.out.println("연산기호가 올바르지 않습니다.");
            return -1;
        }
    }
}

//CalMain클래스 구현
public class CalMain {
    public static void main(String[] args) {
        int n1, n2;
        String str;
        CalTest cal = new CalTest();

        Scanner sc = new Scanner(System.in);
        System.out.print("첫번째 숫자 입력 : ");
        n1 = sc.nextInt();

        System.out.print("두번째 숫자 입력 : ");
        n2 = sc.nextInt();

        System.out.print("연산기호 입력 : ");
        Scanner sc2 = new Scanner(System.in);
        str = sc2.next();//next와 nextLine차이점 설명

        System.out.print("결과 : ");
        System.out.println(cal.getResult(n1, n2, str));
    }
}

```

자바문제 -객체(구구단)

자바문제 -객체(업다운)

자바문제 -객체(입출금)

자바 강의 1주차(3) 문제(배열과 클래스)

자바 강의 1주차(3) 문제(배열을 이용한 그래프)

자바문제 -객체형배열(자판기)

위의 5문제 출제!!!!!!