보강 1

자바 : 데이터

목차

자바란?

- 컴퓨터 프로그래밍이란 무엇인가?

- 컴퓨터 : computer => (무엇인가를)계산하기 위한 장치

- 프로그램 : 수행 순서(절차)
  음악 프로그램 = 음악 + 프로그램
  라디오 프로그램 = 라디오 + 프로그램...

자바란?

- 컴퓨터 프로그래밍이란 무엇인가?

- 컴퓨터를 이용해서 (무엇인가를)계산하기위한 절차를 기술한 것

=> 컴퓨터에게 지시하기 위해서는 컴퓨터의 언어를 알아야 한다.

자바란?

- 컴퓨터가 사용하는 언어 : 기계어
- 사람이 사용하는 언어 : 사람어(영어,한국어등...)

=> 서로 다른 언어 체계를 가지기 때문에 통역이 필요함

자바란?

- 통역의 방법 : (실시간)동시 통역, 통번역
- 동시통역 => 인터프리터 : Javascript,python...
- 통번역 => 컴파일러 : C,C++,Java...

자바란?

- 컴퓨터는 하드웨어와 그것을 제어하는 소프트웨어로 구분되며 하드웨어를 직접 제어하는 소프트웨어를 OS(Operation Software)라고 부른다.

- OS(운영체제)마다 하드웨어를 제어하는 방식(명령어 : 언어 번역체계)가 다르기 때문에 운영체제마다 별도의 프로그램을 제작해야함

자바란?

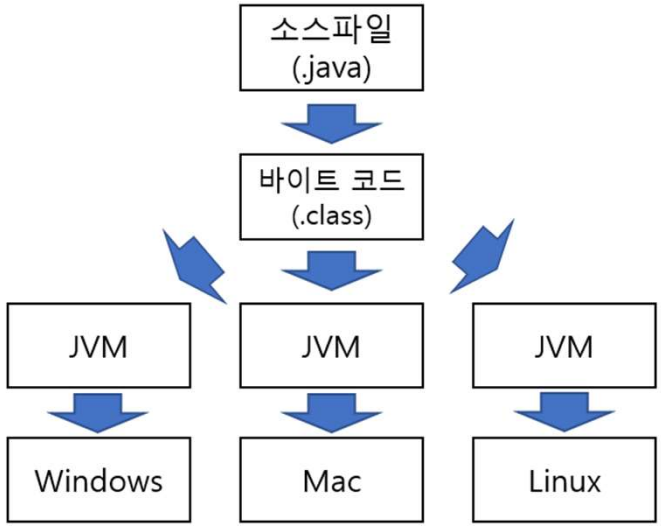
- 컴퓨터는 하드웨어와 그것을 제어하는 소프트웨어로 구분되며 하드웨어를 직접 제어하는 소프트웨어를 OS(Operation Software)라고 부른다.

- OS(운영체제)마다 하드웨어를 제어하는 방식(명령어 : 언어 번역체계)가 다르기 때문에 운영체제마다 별도의 프로그램을 제작해야함

- 자바는 이런 불편한 점을 해소하기 위해 개발됨

자바란?

- 자바는 자바가상머신(JVM)위에 동작하기 때문에 운영체제에 독립적이다.
  - 단 JVM은 운영체제에 맞는 것을 사용해야 함

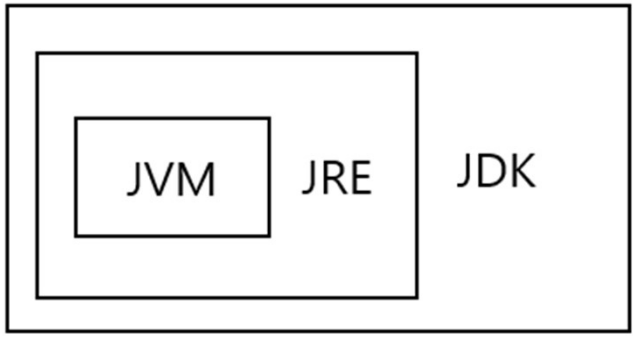


자바란?

- 자바로 제작된 프로그램(바이트 코드)을 실행하기 위해서는 운영체제에 Java어를 해석할 수 있는 번역기를 탑재해야 하는데 이것이 JRE(자바 실행 환경, java runtime environment)이다.
  - JRE는 당연히 JVM를 포함한다.

자바란?

- 다만 사람의 언어를 자바의 언어로 변환하기 위해서는 컴파일러가 있어야 하는데 이 도구를 내장한 것이 JDK(자바 개발도구, java development Kit)이다.
  - JDK는 JRE를 내장하고 있다.



자바란?

- JDK를 설치 후 최초의 프로그램을 개발해 보자

=>메모장으로 켜고 다음 내용을 작성해본다.

```
public class HelloWorld {
	public static void main(String[] args) {
		System.out.println("Hello Java!!");
	 }
}
```

- 저장시 파일 이름과 형식은 'HelloWorld .java'로 저장한다.
  - 단 파일명과 클래스명은 같아야 하며 대소문자를 구분한다.

자바란?

- 이제 명령프롬프트에서 컴파일후 실행 해보자

자바란?

- 다만 코드가 길어지고 파일이 많아지면 메모장으로 작업하기 대단 불편해진다.
- 파일들을 관리하고, 코드의 오류를 수정해주며, 콘솔로 컴파일과 실행까지 통합적으로 관리 해주는 도구를 통합개발환경(Integrated Development Environment(**IDE**)라고 한다.
- 자바IDE의 대표 주자는 무료의 **이클립스**가 있고 유료의 **인텔리제이**가 있다.
- 그외에도 다수 존재하지만 우리는 이클립스(Eclipse)를 사용해서 개발을 해보도록 한다.

자료형과 변수

- 변수란?
  - 데이터를 저장하기 위한 그릇!!
  - 어떤 종류의 데이터든 담을 수 있다

- 변수를 사용하는 이유 :
  - 프로그램의 자료가 되는 데이터의 저장,참조,변경,삭제를 위함

자료형과 변수

- 변수의 사용방법
  1. 변수의 선언
  2. 변수의 초기화
     - 선언과 동시에 초기화 가능

```
int a;
char b;
String c;
double d,f;

a = 10;
b = 'A';
c = "안녕하세요";
d = 3.14;

int g = 10;
String h = "반갑습니다.";

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

자료형과 변수

- 변수의 이름을 짓는 방법
  - 첫글자는 문자 ,$,_만 가능 숫자로는 안된다.
    단 두번째 글자부터는 숫자 가능
  - 대소문자 구분
  - 길이의 제한은 없다
  - 예약어는 사용 불가

- 여러 단어를 붙여서 이름을 정할 때 : 카멀케이스(낙타체:CamelCase)

  ex : greenComputerAcademy

자료형과 변수

- 자료형이란?
  - 변수에 담긴 데이터의 정체를 알려주는 태그

- 변수에 실제로 담기는 데이터는 운영체제에 기록된 표준화된 형식을 따르고 이것을 리터럴이라고 부른다.

자료형과 변수

- 자료형의 종류
  - 기본 자료형
    - 정수 – byte, short, int, long
    - 실수 – float, double
    - 문자 – char
    - 논리 - boolean
  - 참조 자료형
    - 기본 자료형을 제외한 모든 자료형

자료형과 변수

- 정수형
  - byte (1바이트) : -128~127
  - short(2바이트) : -32768~32767
  - int(4바이트) : -2,147,483,648~2,147,483,647
  - long(8바이트): 엄청 크다

  - byte < short < int < long

자료형과 변수

- 실수형
  - 얼마만큼 정밀한가?(소수점 아래로 얼마만큼 표현이 가능한가?)

  - float(4바이트)
  - double(8바이트)
  - E표기법
    - 12345.12 => 1.234512*10^4 =>1.234512e+4
    - 0.001234 => 1.234*10^-3 => 1.234e-3

  - byte < short < int < long < float < double

자료형과 변수

- 문자형
  - 문자 코드표 => ASCII -> unicode(www.unicode.org)
  - '한문자'만을 표기
    => 여러 문자가 담긴 문장을 담기 위해서는 String이라는 참조 자료형이 필요

  - 이스케이프 문자 : 키보드에 있는 문자인데 한문자로 표기하기 힘든 것 (예시: enter키, tap키등)

이스케이프 문자	의미
₩n	개행
₩t	수평 탭
₩'	'
₩"	"
₩₩	₩

자료형과 변수

- 명제의 결과를 담는 자료형
  true , false

자료형과 변수

- 형변환
  - 변수에 지정된 타입을 다른 타입으로 바꾸는 것을 말한다
  - 방법
    - 묵시적 형변환(자동 형변환)
    - 명시적 형변환(강제 형변환)

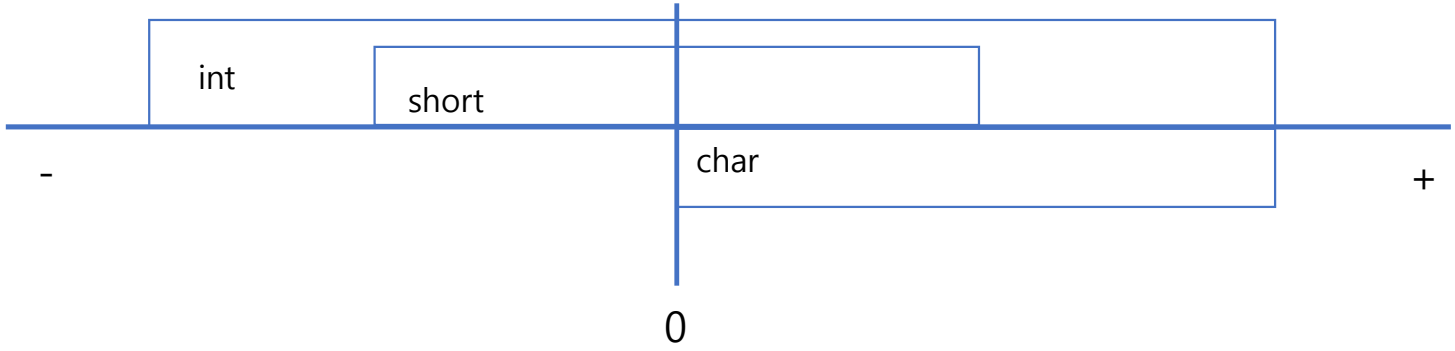
자료형과 변수

- 묵시적 형변환(자동 형변환)
  - 알아서 변환되는 방법
  - 작은 그릇에 있는 데이터를 큰 그릇으로 옮겨 담을 수 있다.
    - 실제 데이터 크기는 중요하지 않다.

- byte < short < int < long < float < double

자료형과 변수

- 묵시적 형변환(자동 형변환)
  - 문자형도 실제로는 숫자로 저장되기 때문에 숫자로 형변환 가능
  - 다만 char는 2바이트이지만 short(2바이트) 자동 변환 불가능하다



자료형과 변수

- 명시적 형변환(강제 형변환)
  - 강제로 변환해줘야하는 변환
    - 오류가 날 가능성을 감수하고 변환 해줌
  - 큰 그릇의 데이터를 작은 그릇에 옮겨 담는다.
    - 실제 데이터 크기는 중요하지 않다.
    - 실제 데이터의 크기는 개발자가 알고 있기 때문에 강제로 옮겨 담음
  - 참조 타입에서의 형변환

- byte < short < int < long < float < double

배열

- 배열의 개념
  - 변수란 기본적으로 하나의 데이터를 담을 수 있다.
  - 그러나 데이터의 양이 많아 질수록 변수의 개수가 많아지게 되고 많아진 변수를 관리하기 불편해진다.
  - 그래서 다수의 데이터를 하나의 변수에 관리하기 위한 기술 이것이 배열이다.

배열

- 배열의 개념
  - 자바 성적을 관리하는 프로그램을 만들어 본다
  - 학생이 5명이라고 할 때 자바 성적은 5개가 존재한다.
  - 100명의 학생이라면 성적 100개 존재

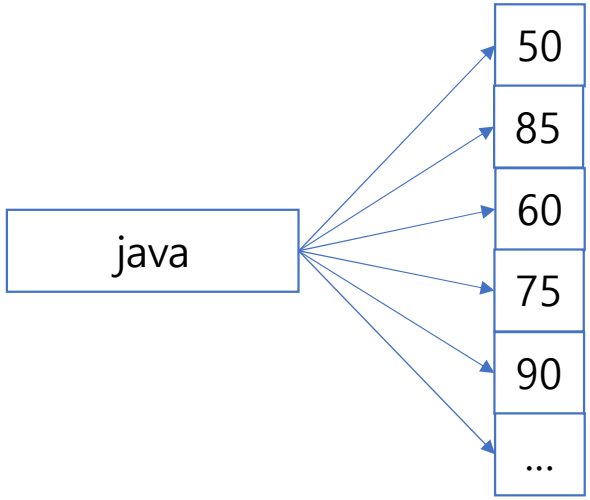
  - 만약 학생들의 총점을 계산하고자 한다면???

```
//배열을 사용하지 않을때 다섯 학생의 자바 점수
int java1=50;
int java2=85;
int java3=60;
int java4=75;
int java5=90;

//총점 계산
int total = java1 + java2 + java3 + java4 + java5;
```

java1	50
java2	85
java3	60
java4	75
java5	90
java.....	...

배열

- 배열의 개념
  - 관리의 편의성과 연산의 효율성을 위해서 같은 타입의 여러 데이터를 하나의 변수로 통제할 필요가 있다.



배열

- 배열의 사용법
  - 선언 =>타입[] 변수명

  ex) int[] 변수명; => int타입의 데이터를 저장할 배열변수

  - 선언후 초기화

  변수명 = new int[배열크기]; => 크기 만큼의 빈 배열이 생성
  변수명 = new int[]{데,이,터} => 배열의 내용물 지정

배열

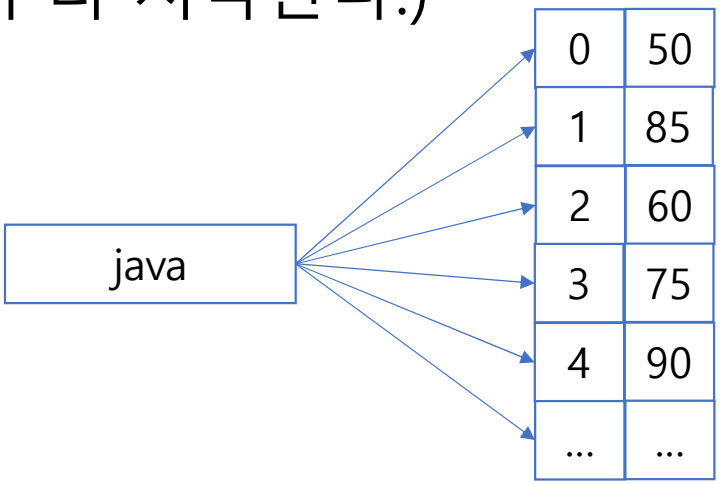
- 배열은 참조 타입이므로 출력시 원하는 정보가 출력되지 않는다.

- 배열도 실제 데이터를 사용할 수 있어야 하는데 이때 필요한 것이 인덱스 이다.(인덱스는 0번 부터 시작한다.)

- 인덱스를 사용한 배열은

일반 변수처럼 사용하면 된다.

```
java[3] = 75;
System.out.println("java[3] : "+java[3]);
```



배열

- 배열의 강점은 정수형 데이터를 인덱스로 사용가능하다는 점이다.

```
int num = 4;
java[num] = 90;
System.out.printf("java[%d] : %d",num,java[num]);
```

```
java[4] : 90
```

- 인덱스를 활용해서 반복을 이용할 수 있다.

```
for(int i=0;i<java.length;i++) {
    System.out.printf("java[%d] : %d \n",i,java[i]);
}
```

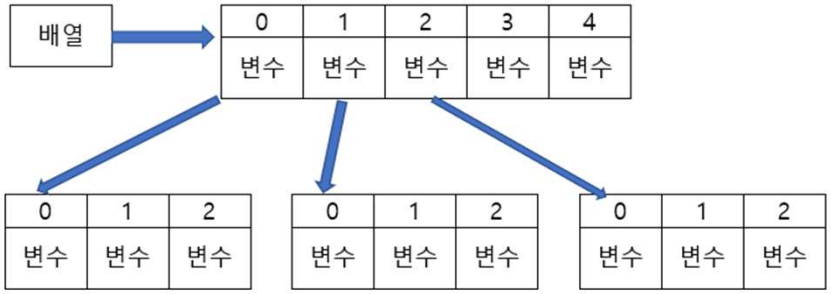
- 배열 자체도 하나의 데이터 덩어리이므로 크기가 바뀐다.
  - 배열의 크기를 알 수 있는 방법 : length 필드를 사용

```
System.out.println("배열의 크기 : "+java.length);
```

배열

- 다 차원 배열
  - 데이터의 양이 많이지면 보다 다각적으로 관리 할 필요가 생긴다.
  - 배열을 확장해서 관리해야 하는데 확장하는 법은 그룹을 지어서 관리 하는 것이다.

예) 5명의 학생이 있다. 학생 성적을 담기 위한 배열이 필요한데 과목이 3개라고 가정한다면 담아야 할 데이터는 총 15개의 데이터가 필요하게 된다.



배열

- 다 차원 배열
  - 다차원 배열의 기본 구조는 배열 안의 배열을 의미한다.

  - 배열의 기본 타입
    - int[] 의미하는 것 => int타입의 데이터를 담는 배열
    - Int[][] 의미하는 것 => int[]타입의 데이터를 담는 배열
    - Int[][][] 의미하는 것 =>int[][]타입의 데이터를 담는 배열

배열

- 다 차원 배열

```java
//다차원 배열 선언
int[] num01 = new int[5];

int[][] num02 = new int[5][];
num02[0] = new int[3];
num02[1] = new int[3];
num02[2] = new int[3];
num02[3] = new int[3];
num02[4] = new int[3];

//=>
int[][] num03 = new int[5][3];

System.out.println("num03 배열의 길이 : "+num03.length);
System.out.println("num03[0] 배열의 길이 : "+num03[0].length);
```

```
num03 배열의 길이 : 5
num03[0] 배열의 길이 : 3
```

배열

- 다 차원 배열

```
//초기화
int[][] num04 = new int[][] {{1,2,3,4,5},{6,7,8,9,10},{11,12,13,14,15}};

//가독성을 위해서
int[][] num05 = new int[][]{
                            {1,2,3,4,5},
                            {6,7,8,9,10},
                            {11,12,13,14,15}
                        };
```

배열

- 향상된 for문(for-each문)
  - 배열에서 사용하는 for문을 보면 항상 똑같은 형태를 가진다

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

  - 똑같은 구조의 for문을 보다 단순화 시킨것이 향상된 for문이다

```
for(배열타입 변수명:배열){...}
```

  - 향상된 for문은 구조가 단순하여 사용하기 쉽지만 인덱스 정보가 필요한 경우 사용이 불가능하다.

배열

- 배열의 단점 :
  - 배열의 단점은 크기가 미리 지정되어 있다는 점이다.
  - 그래서 배열의 크기보다 더 많은 데이터를 저장할 수 없게 된다.

  - 더 많은 데이터를 저장해야 하는 경우 배열 자체를 늘릴 수 없으므로 약간의 절차가 필요하다.

  1) 더 큰 배열을 준비한다.
  2) 이전 배열로 부터 데이터를 복사한다.

배열

- 배열의 복제를 위한 기능

```
System.arraycopy()
```

- 사용법

```
System.arraycopy(원본 배열, 원본 시작 인덱스,대상 배열, 대상 시작 인덱스, 복사 길이)
```

Enum(열거형)

- 일반적인 데이터를 분류하면 숫자와 문자로 구분된다.

- 다만 그룹을 지을 수 있는 데이터의 모음이 있다.

예) 일주일 : {월,화,수,목,금,토,일}
성별 : {남,여}
학원에 소속된 사람의 그룹 : {원장, 행정팀, 강사팀, 학생...}

Enum(열거형)

- 기존에는 데이터 그룹을 분류하기 위해서 숫자코드형태를 사용했다.

예) 일주일 : 일=0, 월=1, 화=2, 수=3, 목=4, 금=5, 토=6

이 경우 데이터 그룹이 많아지면 에러 발생 가능성이 높아진다.

Enum(열거형)

- 에러 발생을 줄이고 데이터 이름 자체를 데이터처럼 사용하고자 만든 자료형이 열거형이다.

- 사용 방법

```
public enum Week {
    SUN, MON, TUE, WED, THU, FRI ,SAT
}
```

열거형의 값은 고정된 값이므로 상수처럼 대문자로 작성한다.

Enum(열거형)

- 에러 발생을 줄이고 데이터 이름 자체를 데이터처럼 사용하고자 만든 자료형이 열거형이다.

- 사용 방법

```
public enum Week {
    SUN, MON, TUE, WED, THU, FRI ,SAT
}
```

열거형의 값은 고정된 값이므로 상수처럼 대문자로 작성한다.

Enum(열거형)

- 사용 방법

```
public static void main(String[] args) {
    Week week = Week.SUN;

    if(week==Week.SUN) {
        System.out.println("와! 일요일다.");
    }
}
```

클래스

- 데이터 저장 관점에서 바라본 클래스

- 데이터를 저장하는 목적중에 하나는 데이터를 잘 포장해서 전달하기 위함이다.

클래스

- 다수의 데이터를 전달해야 하는데 사용가능한 변수가 하나뿐인 경우

예

```
public signUp() {
    Scanner scan = new Scanner(System.in);
    System.out.println("회원 가입을 진행합니다.");
    System.out.println("아이디를 입력하세요.");
    String userId = scan.nextLine();
    System.out.println("암호를 입력하세요.");
    String userPassword = scan.nextLine();
    System.out.println("이름를 입력하세요.");
    String name = scan.nextLine();

    // 생성된 3개의 데이터를 메서드 밖으로 내보내는 방법??

}
```

클래스

- 다수의 데이터를 전달해야 하는데 사용가능한 변수가 하나뿐인 경우

예

```
public scoreInput() {
    Scanner scan = new Scanner(System.in);
    System.out.println("성적을 입력합니다.");
    System.out.println("학생 이름을 입력하세요.");
    String name = scan.nextLine();
    System.out.println("자바점수를 입력하세요.");
    int java = scan.nextInt();
    System.out.println("오라클를 입력하세요.");
    int oracle = scan.nextInt();

    // 생성된 3개의 데이터를 메서드 밖으로 내보내는 방법??
}
```

클래스

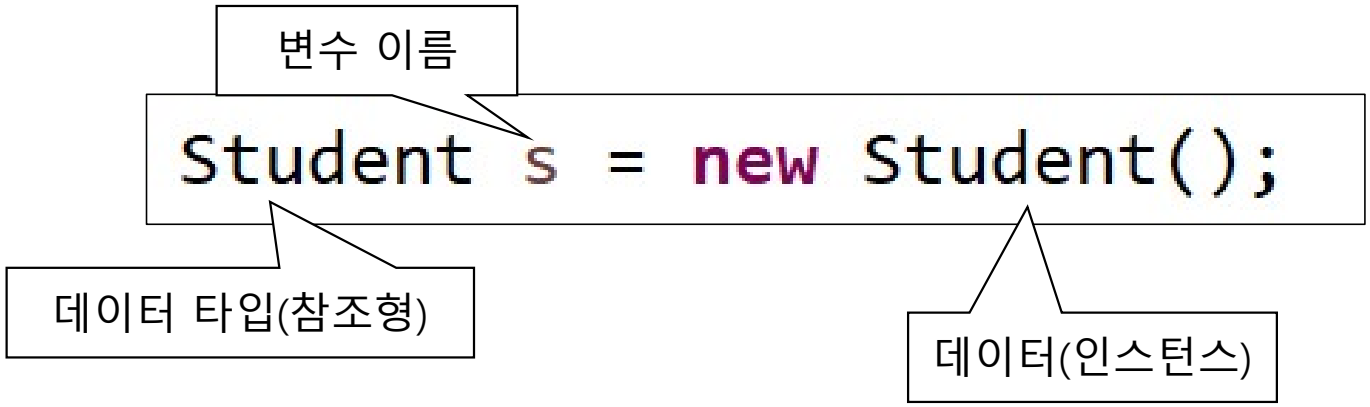
- 서로 다른 타입의 다수의 데이터를 저장할 새로운 데이터 형태가 필요하다.

- 이런 데이터의 형태(Type)를 클래스라고 부른다.
- 그리고 클래스내부에 데이터를 저장 할 수 있는 변수를 필드라고 부른다.

```
public class Student {

    public String userName;
    public int javaScore;
    public int oracleScore;

}
```

클래스

- 클래스 형태의 데이터 타입이 존재하면 그런 타입의 변수에 담을 데이터도 존재하는데 그런 데이터를 인스턴스라고 부른다.



클래스

- 앞서 본 예시에서 데이터를 인스턴스에 담아서 전달해 보자.

```
public Student scoreInput() {
    Scanner scan = new Scanner(System.in);
    System.out.println("성적을 입력합니다.");
    System.out.println("학생 이름을 입력하세요.");
    String name = scan.nextLine();
    System.out.println("자바점수를 입력하세요.");
    int java = scan.nextInt();
    System.out.println("오라클를 입력하세요.");
    int oracle = scan.nextInt();

    Student s = new Student();
    s.userName = name;
    s.javaScore = java;
    s.oracleScore = oracle;

    return s;
}
```

클래스

- 클래스도 변수나 메서드처럼 이름을 규칙에 맞게 지어주어야 한다
  - 하나 이상의 문자로 구성
  - 첫 글자는 숫자가 올수 없다 -> 두번째 글자 이후로는 가능
  - $,_이외의 특수문자 사용불가
  - 예약어 사용불가
  - 첫글자는 대문자 필수

클래스

- 인스턴스는 독립된 데이터이므로 인스턴스 내부에 들어있는 데이터 필드는 독립된 존재가 된다.

```
public static void main(String[] args) {
    Student s1 = new Student();
    s1.userName = "고길동";
    Student s2 = new Student();
    s2.userName = "홍길동";

    System.out.println("s1 인스턴스의 이름 필드 : "+s1.userName);
    System.out.println("s2 인스턴스의 이름 필드 : "+s2.userName);
}
```

클래스

- 인스턴스의 필드에 데이터를 직접 저장하는 것은 보안 그리고 기능이라는 관점에서 문제가 있다.

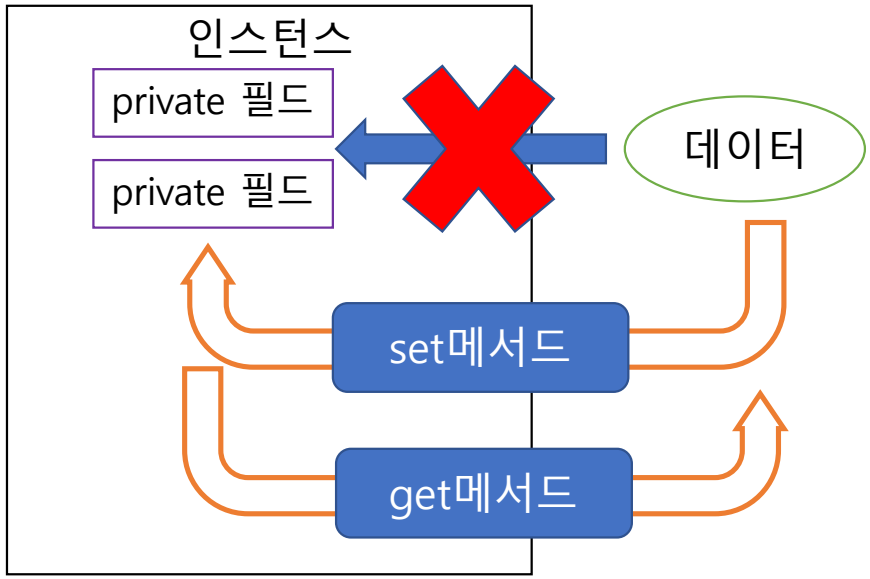
```
public static void main(String[] args) {
    Student s1 = new Student();
    s1.userName = "고길동";
    s1.javaScore = 1000;
    s1.oracleScore = -2000;

    System.out.println(s1.userName+"의 합계 점수 : "+(s1.javaScore+s1.oracleScore));
}
```

클래스

- 따라서 인스턴스의 필드에 데이터를 직접 저장하는 것을 막고 메서드등 기능을 통해서 데이터를 필터링 해서 저장하는 프로그래밍 기법을 캡슐화라고 하며 캡슐화는 클래스간에 결합으로 프로그래밍하는 객체 지향 프로그래밍에서 대단히 중요한 특성중 하나이다.

클래스

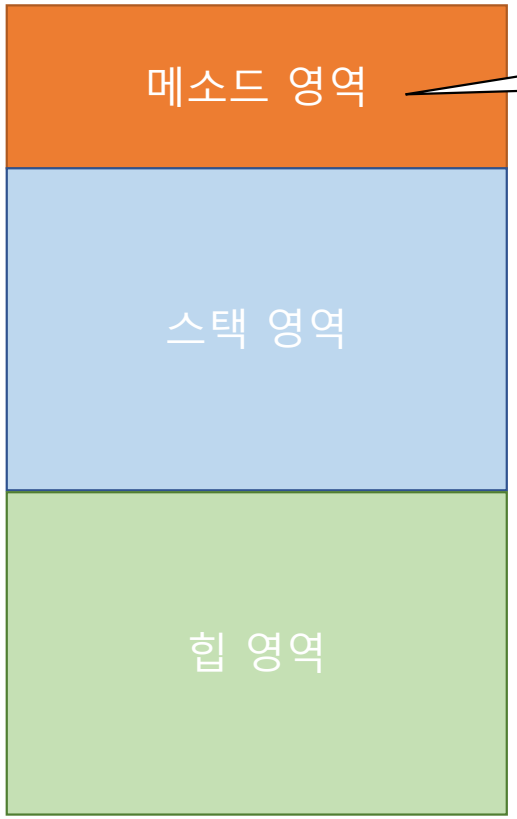
- 필드의 직접적인 접근을 막기 위해 접근제어자 private를 사용한다.



- 데이터를 필드에 저장할 목적으로 사용되는 메서드를 Seter메서드라고 한다.
- 필드에 저장된 데이터을 읽어드릴 목적으로 사용되는 메서드를 Geter메서드라고 한다.

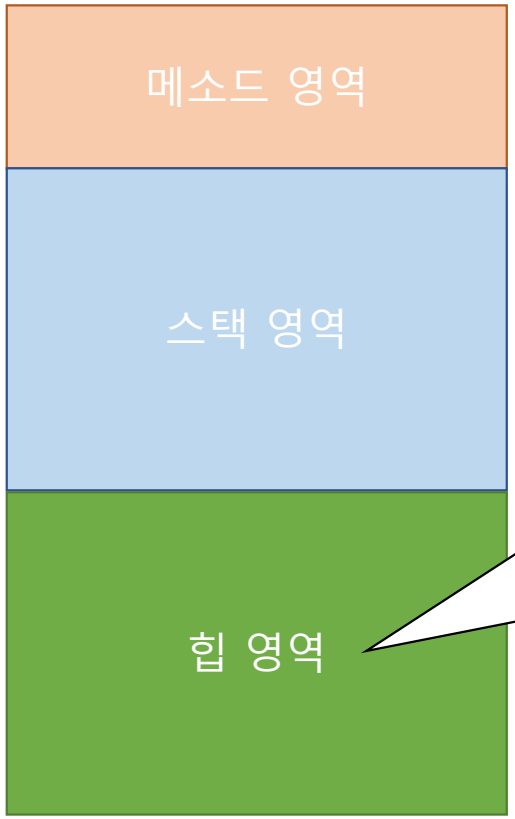
메모리 구조

메모리 구조



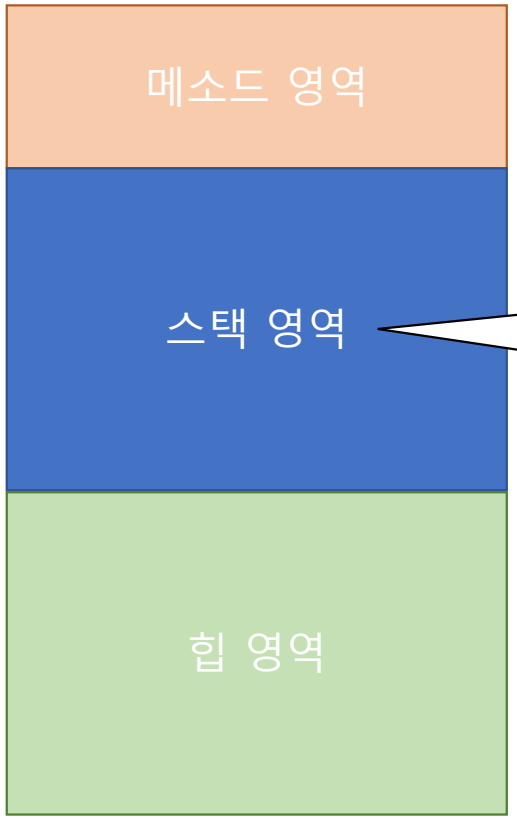
메소드 영역
클래스의 바이트 코드를 읽어서 저장하는 영역
정적 멤버등이 여기에 저장된다

메모리 구조



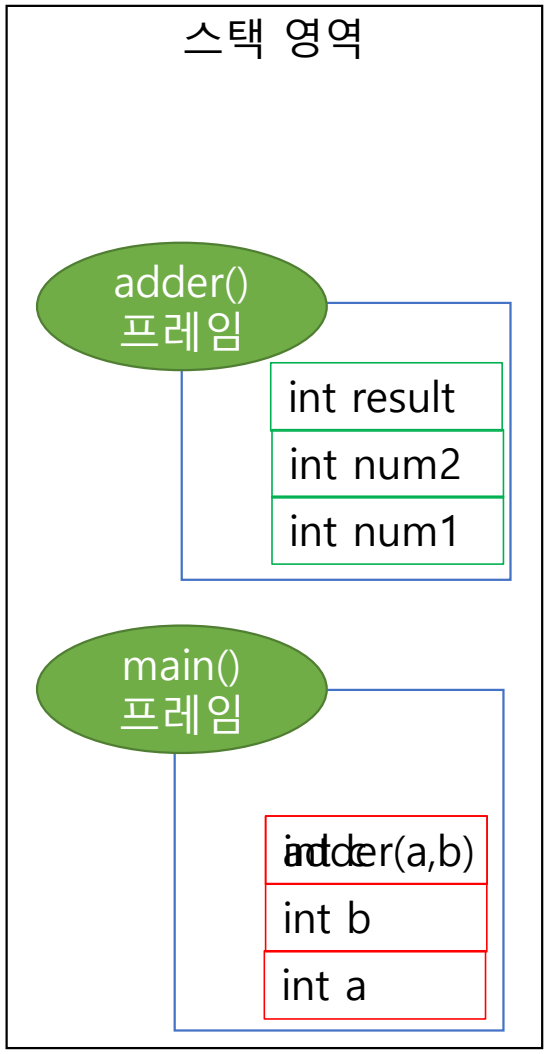
힙 영역:
-메소드영역에 있던 클래스로 부터 생성된 인스턴스가 머무는 영역
-해당 인스턴스를 참조하는 변수가 없다면 메모리만 차지하는 의미 없는 객체(쓰레기)가 된다.
-JVM은 이런 쓰레기 객체를 처리하기 위해 가비지 컬렉터를 실행시킨다.
-단 자바는 가비지(쓰레기)를 직접 처리하기 위한 코드를 제공하지 않는다.

메모리 구조



스택 영역:
- 변수가 선언되어 생성되는 영역
- 메소드가 추가될 때 마다 프레임이 추가되고 메소드가 종료되면 프레임채 삭제된다.
- 여기서 선언된 참조변수는 힙 영역에 생성된 인스턴스를 가르킨다.

메모리 구조



- 스택이란?
  - 자료구조에서 이야기 하는 저장방식
  - FILO방식으로 한쪽면이 막힌 상자같은 구조

```
public static void main(String[] args) {
    int a = 10;
    int b = 20;
    int c = adder(a,b);

}

private static int adder(int num1, int num2) {
    int result = num1+num2;
    return result;
}
```

배열 문제

배열을 사용해서 학생 성적을 관리해봅시다
- 학생 5명의 이름은 고길동,김길동,이길동,박길동,홍길동
- 각각 학생의 점수는 국어,영어,수학 3과목의 점수를 입력받는다.
- 학생 이름으로 개개인의 성적을 조회한다.
- 전체 학생 성적을 과목별로 조회한다.
- (추가)각 학생의 평균 성적으로 석차를 만들어서 조회한다.

종료를 선택하지 않으면 메뉴가 반복적으로 출력되도록 하세요.
메뉴는 성적입력, 성적조회, 석차조회, 종료로 만들고

성적 조회시 개인별 성적조회, 과목별 성적조회를 선택할수 있도록 제작하세요.

예시 정보

	국어	영어	수학
고길동	78	64	82
김길동	85	71	64
이길동	74	69	57
박길동	74	77	95
홍길동	68	95	84

석차 알고리즘

	평균
고길동	60
김길동	80
이길동	40
박길동	100
홍길동	20

	점수	석차					석차
고길동	60	1	+1	+1			3등
김길동	80	1	+1				2등
이길동	40	1	+1	+1	+1		4등
박길동	100	1					1등
홍길동	20	1	+1	+1	+1	+1	5등

클래스 문제

클래스을 사용해서 학생 성적을 관리해봅시다
학생 5명의 이름은 고길동,김길동,이길동,박길동,홍길동 입력받는다.
각각 학생의 점수는 국어,영어,수학 3과목의 점수를 입력받는다.
학생 이름으로 개개인의 성적을 조회한다.
전체 학생 성적을 과목별로 조회한다.
(추가)각 학생의 평균 성적으로 석차를 만들어서 조회한다.

더미 데이터

	국어	영어	수학
고길동	78	64	82
김길동	85	71	64
이길동	74	69	57
박길동	74	77	95
홍길동	68	95	84

1단계 : Student 클래스 생성
- 필드는 이름, 점수배열(3), 총점, 평균, 석차로 구성한다.
- 생성자는 이름과 점수배열을 받아서 생성하고 총점과 평균은 내부 메소드를 통해서 생성한다.석차는 기본 1로 저장한다.
- 메소드는 총점구하는 메소드, 평균구하는 메소드와 함께 정보를 출력하는 메소드로 구성한다.
- 정보는 이름, 국어,영어,수학점수, 총점, 합계 점수를 출력한다.

2단계 : 메뉴를 구성한다.
- 성적 입력, 성적 조회, 석차 조회로 구성한다.

3단계 각 기능을 완성한다.
- 성적 입력기능을 구현한다.
- 성적 조회 기능을 구현하되
  - 학생 별 이름조회시 정보를 출력하고
  - 과목별 이름 입력시 과목의 총점과 평균을 출력한다.
- 석차는 모든 학생의 정보가 입력된 후 실행 하도록 한다.

문제

회원 가입, 로그인, 종료
로그인 성공시 이름과 전화번호 출력하는 코드를 작성해보자

1. 아이디, 비밀번호, 이름, 전화번호 4가지 정보를 담을 수 있는 Member클래스를 만들어 봅시다.
2. prt()메소드를 통해서 이름과 전화번호를 출력하는 메소드를 만들어 봅니다.
3. 멤버 정보는 총 100명분의 정보만 저장이 가능하도록 합니다.

.

1단계: [Member클래스]를 만들어 보자
필드 : 아이디,비밀번호, 이름, 전화번호
메소드 : 이름과 전화번호를 출력하는 메소드
생성자 : 아이디,비밀번호,이름,전화번호를 입력받아서 인스턴스 생성하는 생성자

2단계 : [Main클래스] main메소드를 만들어 보자
메뉴로 회원가입기능, 로그인 기능, 종료기능 작동시키는 while~ switch문을 완성시켜보자

3단계 [Main클래스]Member타입의 static변수를 생성
3-1단계 : 회원 가입 : 3개의 정보를 입력받고 인스턴스를 생성한후 위 변수에 담는다.
3-2단계 : 로그인 : 아이디와 비밀번호를 입력받고 위 변수에 담긴 인스턴스의 아이디와 비밀번호를 비교해서 로그인 처리를 진행하자

문제

회원 가입, 로그인, 종료 --- 추가 문제
로그인 성공시 이름과 전화번호 출력하는 코드를 작성해보자

1. 아이디, 비밀번호, 이름, 전화번호 4가지 정보를 담을 수 있는 Member클래스를 만들어 봅시다.
2. prt()메소드를 통해서 이름과 전화번호를 출력하는 메소드를 만들어 봅니다.
3. 멤버 정보는 총 100명분의 정보만 저장이 가능하도록 합니다.

.

[추가]: 로그인 성공후 다시 로그인 시도할 때는 "이미 로그인 되어있습니다"메시지를 띄우세요.

단, 로그아웃 메뉴를 추가하여 로그아웃이 이루어지면 "로그 아웃이 되었습니다."메시지를 띄우도록 처리하세요.

문제

과제 게시판

게시판 작성 프로그램을 만들어 보자
게시판은 제목, 작성자, 내용, 삭제시 비밀번호로 구성된다.
메뉴는 게시글 보기, 글작성, 글 삭제로 구성된다.
글 삭제시 삭제 여부를 묻고 작성자, 비밀번호를 입력하여 일치여부를 확인한후 삭제한다.
게시글보기는 우선 제목으로 목록을 보여주고 해당 제목에 번호를 입력하면 게시글의 제목, 작성자, 글 내용을 출력하도록한다.
출력 후 목록으로 돌아간다.

[추가] 글 수정 기능도 추가해봅니다.
비밀번호를 입력해야지 수정이 가능하도록 만들어 봅니다.

1단계 : 게시판(Board) 클래스를 작성합니다.
- 필드는 제목, 작성자, 내용, 비밀번호로 구성합니다.
- 생성자는 위 4개의 정보를 입력받아서 생성자를 구성합니다.
- 메소드는 기본 출력문 제목,작성자,내용으로 구성합니다.

2단계 : 기본 메뉴를 구성합니다.
- 게시글보기, 글작성, 글 삭제로 구성합니다.

3단계 : 각 기능을 완성합니다.
- 게시글 보기는 우선 제목을 번호 매겨서 보여주고 해당 번호 입력을 받아서 게시글의 내용을 출력하도록 합니다.
- 글 작성을 제목 작성자 비밀번호 내용 작성해서 저장합니다.
- 글 삭제는 글 목록을 보여주고 해당 번호와 작성자와 비밀번호를 받아서 일치하면 해당 글을 삭제 합니다.

문제

게시판을 만들어봅니다.

1) 회원 가입과 로그인 기능을 구현해 봅니다.
2) 회원이 아닌 사람도 게시글의 목록을 볼 수 있습니다.
3) 로그인은 아이디와 비밀번호를 사용해서 로그인 처리를 합니다.
4) 로그아웃 기능을 구현해 봅니다.
5) 로그인 성공 후 글 목록보기, 글쓰기, 자기 글 보기를 할 수 있다.
6) 자기 글 보기에서 자기 글을 수정하거나 삭제 할 수 있다.
7) 회원 가입시 같은 아이디가 존재해서는 안된다.

단 로그인관련 로직과 게시판 관련 로직을 구분해야 한다.

1단계: