

Day11 (H1)

Array of Objects

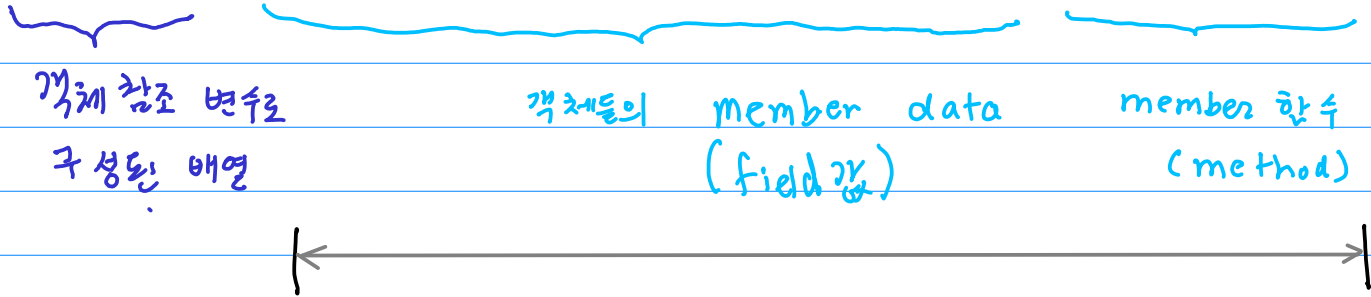
20150824

Copyright (c) 2015 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

st1	s1 →	Kor = 99	Eng = 45	Math = 90	→ Avg()
st2	s2 →	Kor = 88	Eng = 55	Math = 80	→ Avg()
st3	s3 →	Kor = 77	Eng = 65	Math = 90	→ Avg()
st4	s4 →	Kor = 66	Eng = 75	Math = 80	→ Avg()
st5	s5 →	Kor = 55	Eng = 85	Math = 90	→ Avg()

st1	S[0]	Kor = 99	Eng = 45	Math = 90	→ Avg()
st2	S[1]	Kor = 88	Eng = 55	Math = 80	→ Avg()
st3	S[2]	Kor = 77	Eng = 65	Math = 90	→ Avg()
st4	S[3]	Kor = 66	Eng = 75	Math = 80	→ Avg()
st5	S[4]	Kor = 55	Eng = 85	Math = 90	→ Avg()



class Student 선언

- S[0].Avg()
- S[1].Avg()
- S[2].Avg()
- S[3].Avg()
- S[4].Avg()

학생들 객체 의 참조변수

정수형 byte, short, int, long
실수형 float, double
논리형 boolean
문자형 char

} 기본형

class

interface

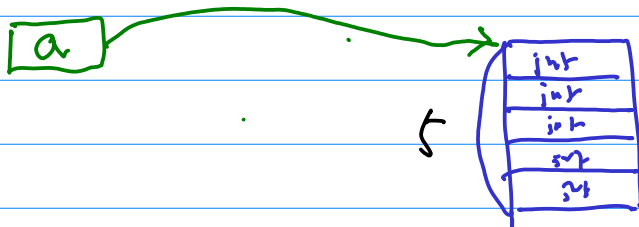
} 객체형

Primitive Type	Size	Minimum Value	Maximum Value	Wrapper Type
char	16-bit	Unicode 0	Unicode 216-1	Character
byte	8-bit	-128	+127	Byte
short	16-bit	-215 (-32,768)	+215-1 (32,767)	Short
int	32-bit	-231 (-2,147,483,648)	+231-1 (2,147,483,647)	Integer
long	64-bit	-263 (-9,223,372,036,854,775,808)	+263-1 (9,223,372,036,854,775,807)	Long
float	32-bit	Approx range 1.4e-045 to 3.4e+038		Float
double	64-bit	Approx range 4.9e-324 to 1.8e+308		Double
boolean	1-bit	true or false		Boolean

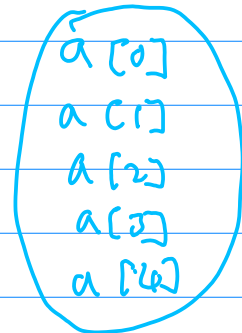
Primitive Data Type	Default Value 3.6
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
boolean	false

Array of References to Objects

`int[] a = new int[5]`



`a[i]`: int 변수



`double [] a = new double [5]`

`a[i]`: double 변수

String [] a = new String [5]
class type

`a[i]`: String 객체 참조 변수

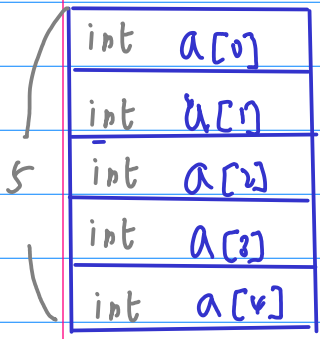
Student [] a = new Student [5]
class type

`a[i]` Student 객체 참조 변수

↓
references

⑨

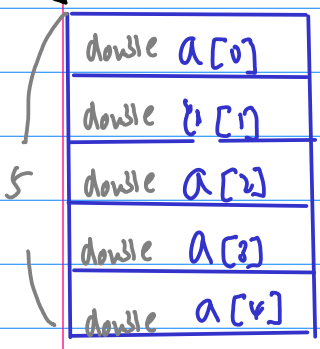
`int [] a = new int [5]`



a[0] : int 변수 a[0] = 100
a[1] : int 변수
a[2] : int 변수
a[3] : int 변수
a[4] : int 변수

⑩

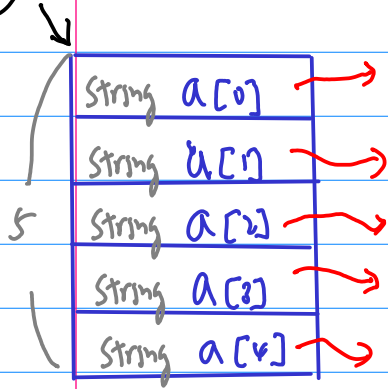
`double [] a = new double [5]`



a[0] : double 변수 a[1] = 3.14
a[1] : double 변수
a[2] : double 변수
a[3] : double 변수
a[4] : double 변수

String [] a = new String [5]

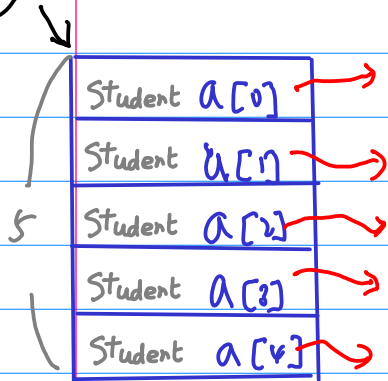
①



a[0] : String 클래스 type의 객체 참조 변수
a[1] : String 클래스 type의 객체 참조 변수
a[2] : String 클래스 type의 객체 참조 변수
a[3] : String 클래스 type의 객체 참조 변수
a[4] : String 클래스 type의 객체 참조 변수

Student [] a = new Student [5]

①



a[0] : Student 클래스 type의 객체 참조 변수
a[1] : Student 클래스 type의 객체 참조 변수
a[2] : Student 클래스 type의 객체 참조 변수
a[3] : Student 클래스 type의 객체 참조 변수
a[4] : Student 클래스 type의 객체 참조 변수

String : 클래스 이름, 생성자 함수

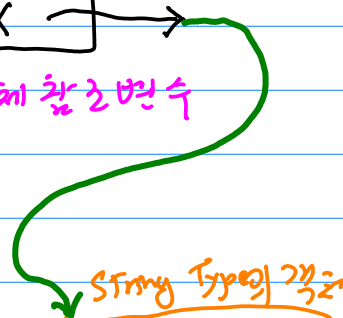
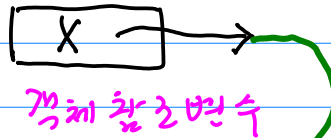
String X ;
class type 변수



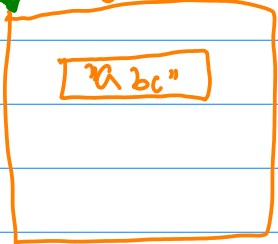
new String ("abc");

new
클래스
이름
생성
자
함수

생성자
함수



String Type의 객체



String

X ;

class
type



참조변수만 선언

객체 생성

String

X = new String ("hello");

String

X = "hello";

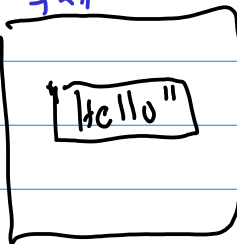
class
type

생성과
할 수

참조변수

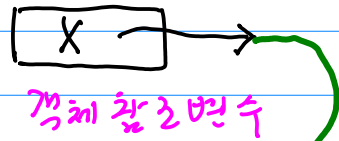


객체

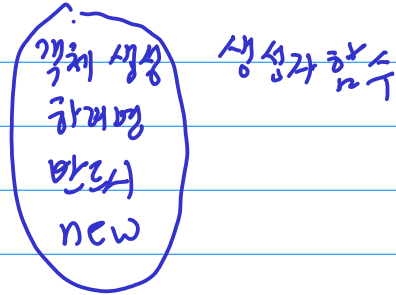


Student : 클래스 이름, 생성자 함수

Student X ;
class type 변수



new Student (90, 80, 95);



class type

변수

개체 참조 변수

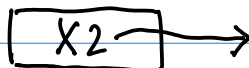
Student

X1;



Student

X2;



Student

X3;



Student

X4;

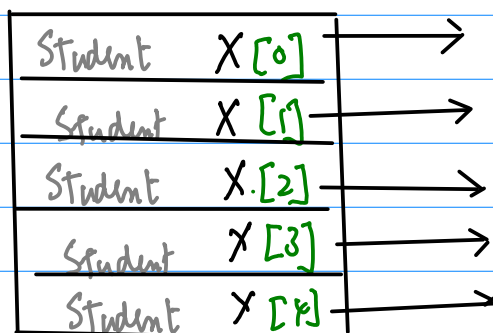


Student

X5;



Student[] X = new Student[5]

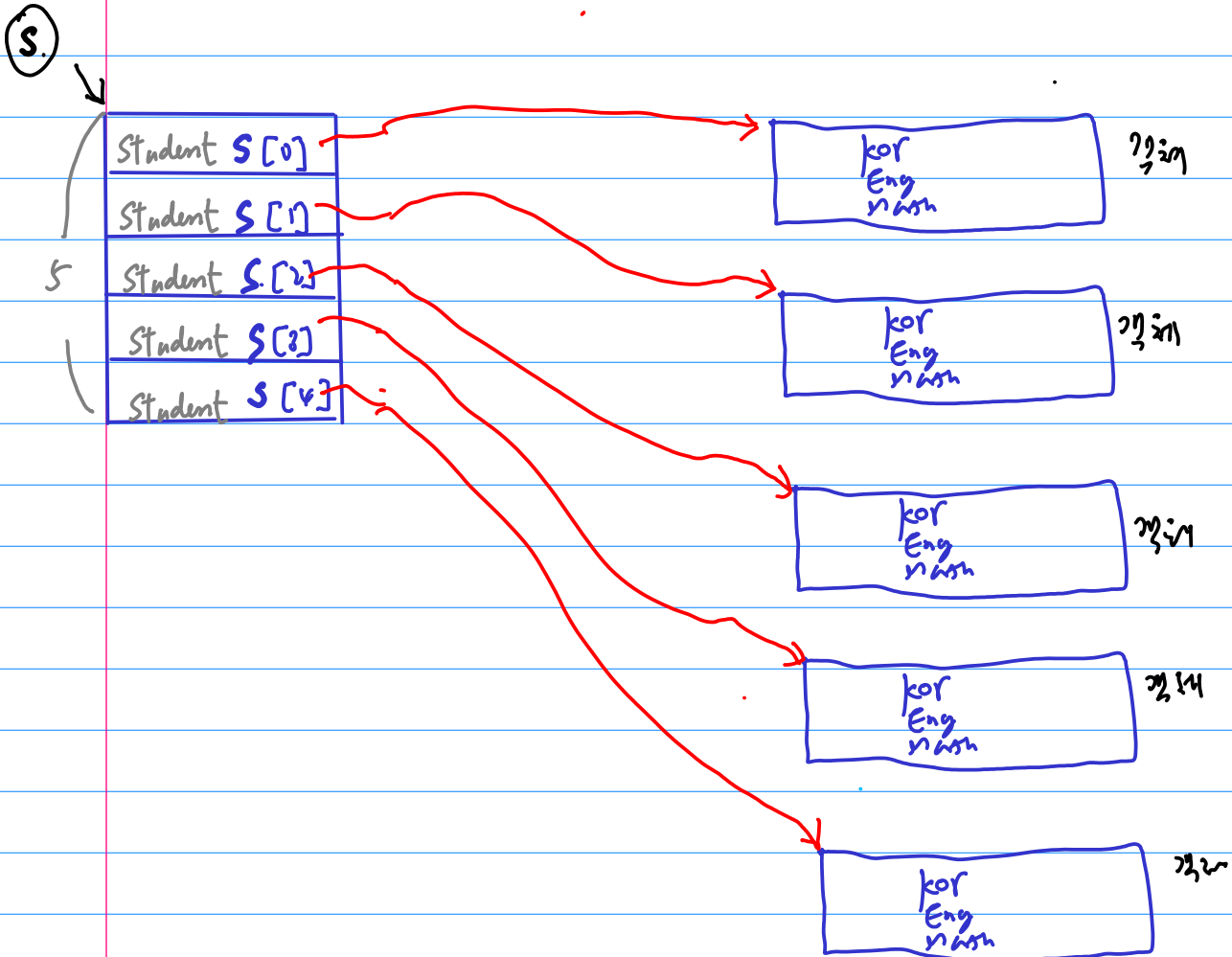


```
Student[] S = new Student[5];
```

객체
참조
배열

```
S[0] = new Student(99, 45, 90);  
S[1] = new Student(88, 55, 80);  
S[2] = new Student(77, 65, 90);  
S[3] = new Student(66, 75, 80);  
S[4] = new Student(55, 85, 90);
```

```
S[0].disp();  
S[1].disp();  
S[2].disp();  
S[3].disp();  
S[4].disp();
```



S

st1	S[0]	Kor = 99	Eng = 45	Math = 90	→ Avg()
st2	S[1]	Kor = 88	Eng = 55	Math = 80	→ Avg()
st3	S[2]	Kor = 77	Eng = 65	Math = 90	→ Avg()
st4	S[3]	Kor = 66	Eng = 75	Math = 80	→ Avg()
st5	S[4]	Kor = 55	Eng = 85	Math = 90	→ Avg()

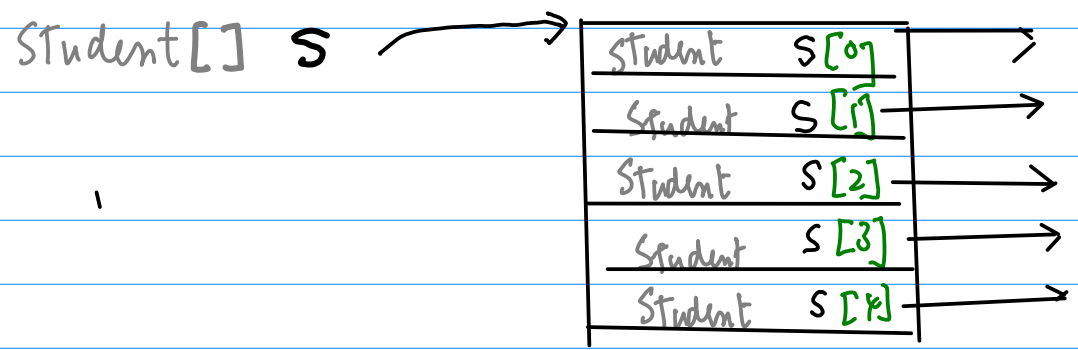
avg-kor() avg-eng() avg-math()

~~S[0].avg-kor()
 S[1].avg-kor()
 S[2].avg-kor()
 S[3].avg-kor()
 S[4].avg-kor()~~

Student.avg_kor()
 Student.avg_eng()
 Student.avg_math()

Static method

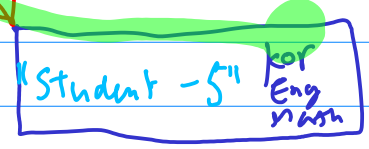
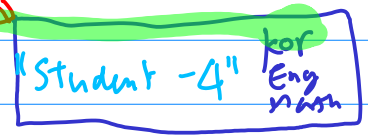
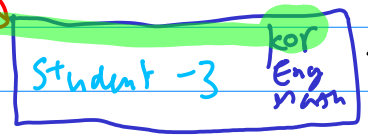
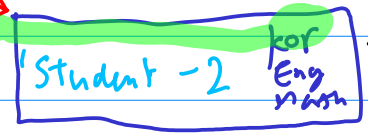
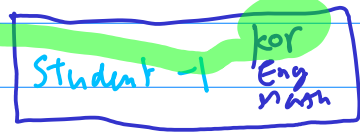
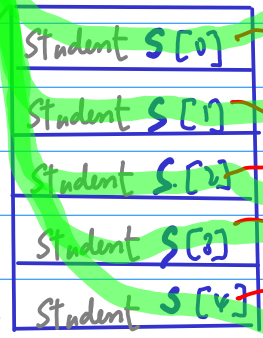
Student[] S = new Student[5]



S

Student

i=0
i=1
i=2
i=3
i=4



29.27

29.27

29.27

29.14

29.20

call main

Student.avg kor(S);

class static method

Class Student

```

static void avg_kor(Student[] S) {
    int i;
    double avg = 0;
    for (i=0; i<S.length; ++i) {
        avg += S[i].Kor;
    }
    avg /= S.length;

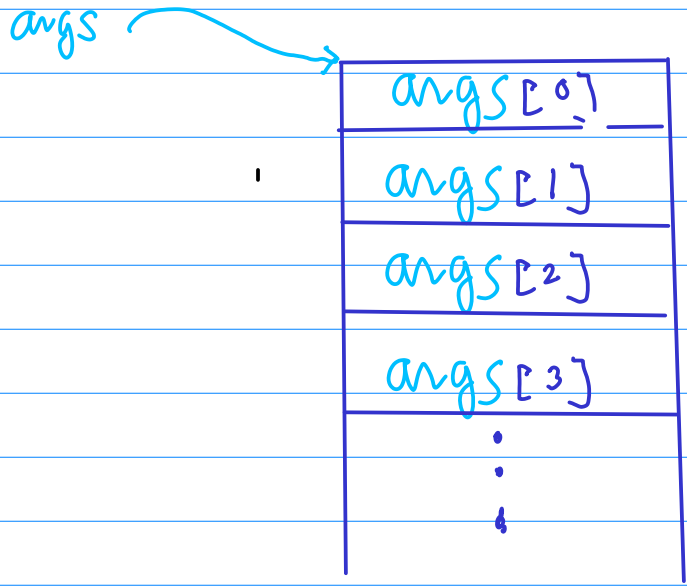
    System.out.println("Kor Avg = " + avg);
}

```

S[0].kor
S[1].kor
S[2].kor
S[3].kor
S[4].kor

parameter

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




```
class Student {
    int Kor;
    int Eng;
    int Math;

    Student(
        ) { Kor= 0; Eng=0; Math=0; }
    Student(int x, int y, int z) { Kor= x; Eng= y; Math= z; }

    double Avg() { return (Kor+Eng+Math) / 3.0; }

    void disp() {}

    static void avg_mode( Student[] X, int mode) {}

    static void avg_kor( Student[] X ) {}
    static void avg_eng( Student[] X ) {}
    static void avg_math( Student[] X ) {}
}

```

```
public class StudentTest2 {

    /**
     * @param args
     */
    public static void main(String[] args) {}

}

```

```
class Student {
    int Kor;
    int Eng;
    int Math;

    Student(
        ) { Kor= 0; Eng=0; Math=0; }
    Student(int x, int y, int z) { Kor= x; Eng= y; Math= z; }

    double Avg() { return (Kor+Eng+Math) / 3.0; }

    void disp() {
        System.out.println( "-----");
        System.out.println( "Kor= " + Kor    );
        System.out.println( "Eng= " + Eng    );
        System.out.println( "Math=" + Math  );
        System.out.println( "GPA= " + Avg()  );
    }
}
```

```
static void avg_mode( Student[] X, int mode) {  
    double avg= 0.0; int i;  
  
    for (i=0; i<X.length; ++i) {  
        switch (mode) {  
            case 0: avg += X[i].Kor; break; // mode=0  
            case 1: avg += X[i].Eng; break; // mode=1  
            case 2: avg += X[i].Math; break; // mode=2  
            default: avg = 0; break;  
        }  
    }  
    avg /= X.length;  
  
    String str;  
    switch (mode) {  
        case 0: str = "*** Kor ="; break; // mode=0  
        case 1: str = "*** Eng ="; break; // mode=1  
        case 2: str = "*** Math="; break; // mode=2  
        default: str = "*** Wrong mode!!!"; break;  
    }  
  
    System.out.println(str + avg);  
}
```

```
static void avg_kor( Student[] X ) {  
    double avg= 0.0; int i;  
    for (i=0; i<X.length; ++i) avg += X[i].Kor;  
    avg /= X.length;  
    System.out.println("** Kor Avg = " + avg);  
}  
static void avg_eng( Student[] X ) {  
    double avg= 0.0; int i;  
    for (i=0; i<X.length; ++i) avg += X[i].Eng;  
    avg /= X.length;  
    System.out.println("** Eng Avg = " + avg);  
}  
static void avg_math( Student[] X ) {  
    double avg= 0.0; int i;  
    for (i=0; i<X.length; ++i) avg += X[i].Math;  
    avg /= X.length;  
    System.out.println("** Math Avg = " + avg);  
}
```

```
public class StudentTest2 {
```

```
    /**
```

```
     * @param args
```

```
     */
```

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

```
    }
```

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
  
    Student[] S = new Student[5]; // S[i] : refer  
  
    S[0] = new Student(99, 45, 50);  
    S[1] = new Student(88, 55, 80);  
    S[2] = new Student(77, 65, 90);  
    S[3] = new Student(66, 75, 80);  
    S[4] = new Student(55, 85, 90);  
  
    S[0].disp();  
    S[1].disp();  
    S[2].disp();  
    S[3].disp();  
    S[4].disp();  
  
    Student.avg_kor( S );  
    Student.avg_eng( S );  
    Student.avg_math( S );  
  
    Student.avg_mode( S, 0);  
    Student.avg_mode( S, 1);  
    Student.avg_mode( S, 2);  
}
```