

# 성적처리 실습

경북대학교  
소프트웨어융합과  
배희호 교수

# 학생별 성적처리

- 학생 10명의 학번(hakbun), 이름(name), 국어(kor), 영어(eng), 수학(math) 성적을 입력 받아 학생 각각의 총점(sum), 평균(avg)과 등수(rank)를 구하는 Program을 작성 하여라
- 목적
  - Data에 대한 반복적인 처리를 배열을 이용하여 간단하게 처리하는 방법을 실습

2. 교과학습 발달 상황 ( 2 학기 )

교과	도	국	사	산	자	체	음	미	신	합	담	보
구분	덕	어	회	수	연	육	악	술	과	계	임	호
월											인	자
9 월 말	85	95	85	100	85	85	85	95	90			
10 월 말	100	95	85	85	85	85	80	85	85			
11 월 말	90	85	100	60	65	80	65	90	85			
총 점	285	275	270	245	235	250	220	270	260			
평 균	95	92	90	82	78	83	73	90	87	770		
평 가	수	수	수	우	미	우	미	수	우			
종합의견	성리하고 학습태도 바르나 좀더 노력요합니다.											

# 학생별 성적처리

## ■ 문제 분석

- 배열을 이용하여 학생의 성적을 관리하는 프로그램을 작성 (편의상 학생은 10명으로 제한)

## ■ Data와 정보 확인

- 입력내용 – 학번, 이름, 국어, 영어, 수학 (Data)
- 출력결과 – 학번, 이름, 국어, 영어, 수학,  
합계, 평균, 등수 (정보)

# 학생별 성적처리

학번	이름	국어	영어	수학	총점	평균	등수
0801211	홍길동	90	90	95	275	91.66	3
0801234	이혜인	87	88	90	265	88.33	5
0801345	김명성	75	100	95	270	90.00	4
0801567	경복대	100	100	99	299	99.66	1
0801678	이대학	100	75	89	264	88.00	6
0801251	여성학	79	70	65	214	71.33	10
0801987	정대인	83	69	89	241	80.33	9
0801684	학성기	92	91	98	281	93.66	2
0801754	조동기	90	65	89	244	81.33	8
0801840	박대박	77	71	100	248	82.66	7



Futuristic Innovator

京福大學校

KYUNGBOK UNIVERSITY

# 학생별 성적처리

## ■ 국어 성적 평가

국어 점수	등급(grade)
90~100	수(秀)
80~89	우(優)
70~79	미(美)
60~69	양(良)
0~59	가(可)

# 학생별 성적처리

## ■ 수학 성적 평가

수학 점수	등급(grade)
90~100	A
80~89	B
70~79	C
60~69	D
0~59	F

# 학생별 성적처리

## ■ 영어 성적 평가

점수(score)	등급(grade)
95~100	A+
90~94	A
85~89	B+
80~84	B
75~79	C+
70~74	C
65~69	D+
60~64	D
0~59	F

# 학생별 성적처리

## ■ 배열을 초기화 하는 방법

```
String[ ] hakbun = {"0801211", "0801234", "0801345",  
                    "0801567", "0801678", "0801251", "0801987",  
                    "0801684", "0801754", "0801840"};  
String[ ] name = new String[] {"홍길동", "이혜인",  
                                "김명성", "경복대", "이대학", "여성학",  
                                "정대인", "학성기", "조동기", "박대박"};  
  
int[ ] kor = new int[SIZE];  
int[ ] eng = new int[SIZE];  
int [ ] math = new int[SIZE];  
int[] sum = new int[SIZE];  
float[] avg = new int [SIZE];  
int rank[] = new int [SIZE];
```



# 학생별 성적처리

- 성적 Data는 Keyboard를 통해 입력

C:\WINDOWS\system32\cmd.exe					
1 번째 학생	홍길동님의	국어 성적	:	78	
1 번째 학생	홍길동님의	영어 성적	:	90	
1 번째 학생	홍길동님의	수학 성적	:	67	
2 번째 학생	이대한님의	국어 성적	:	56	
2 번째 학생	이대한님의	영어 성적	:	90	
2 번째 학생	이대한님의	수학 성적	:	100	
3 번째 학생	한나라님의	국어 성적	:	65	
3 번째 학생	한나라님의	영어 성적	:	89	
3 번째 학생	한나라님의	수학 성적	:	90	
4 번째 학생	이순신님의	국어 성적	:		

# 학생별 성적처리

C:\WINDOWS\system32\cmd.exe

```
*****
학번      이름      국어  영어  수학  총점  평균
*****
1601003   홍길동      80   100   95   275   91.67
1601007   이대한      90    88   89   267   89.00
1601013   한나라      56    45  100   201   67.00
1601024   이수신      78    68   67   213   71.00
1601026   김동근      34    67   45   146   48.67
1601058   김현호      69    88   78   235   78.33
1601077   이동국      78    99   83   260   86.67
1601085   박예림      88   100   92   280   93.33
1601096   김혜숙     100    67   77   244   81.33
1601110   나희영      75    88   67   230   76.67
*****
계속하려면 아무 키나 누르십시오 . . .
```



# 학생별 성적처리

- 총점을 기준으로 석차를 계산해보자
- 국어 성적을 평점 점수(수, 우, 미, 양, 가)로 평가하여라.  
(switch ~ case문 사용)
- 영어 성적을 평점 점수(A, B, C, D, F)로 평가하여라 (if ~ else문 사용)
- 수학 성적을 평점 점수(A0, A+, B0, B+, C0, C+, D0, D+, F)로 평가하여라.  
(switch ~ case문 사용)
- 최종 출력은 학번 순으로 정렬하여 출력하여라.

# 학생별 성적처리

학번	이름	국어	영어	수학	총점	평균	석차
1601003	홍길동	78(미)	90(A)	100(A+)	268	89.33	3
1601007	이대한	78(미)	90(A)	56(F)	224	74.67	9
1601013	한나라	89(우)	99(A)	89(B+)	277	92.33	2
1601024	이순신	78(미)	90(A)	90(A0)	258	86.00	4
1601026	김동근	67(양)	89(B)	78(C+)	234	78.00	8
1601058	김현호	98(수)	87(B)	65(D+)	250	83.33	5
1601077	이동국	70(미)	90(A)	89(B+)	249	83.00	6
1601085	박예림	78(미)	90(A)	78(C+)	246	82.00	7
1601096	김혜숙	56(가)	78(C)	89(B+)	223	74.33	10
1601110	나희영	90(수)	98(A)	90(A0)	278	92.67	1

계속하려면 아무 키나 누르십시오 . . .

# 학생별 성적처리(1차원 배열)

```
public static void main(String[] args) throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    String[] hakbun = {"1601003", "1601007", "1601013",  
        "1601024", "1601026", "1601058", "1601077",  
        "1601085", "1601096", "1601110"};  
    String[] name = {"홍길동", "이대한", "한나라",  
        "이순신", "김동근", "김현호", "이동국",  
        "박예림", "김혜숙", "나희영"};  
    String s_temp;  
    int[] kor = new int[name.length];  
    int[] eng = new int[name.length];  
    int[] math = new int[name.length];  
    int[] sum = new int[name.length];  
    int test;  
    float[] avg = new float[name.length];  
    int[] rank = new int[name.length];  
    float f_temp;
```

# 학생별 성적처리(1차원 배열)

```
int i, j;  
String[] hakjum = new String[name.length];  
char[] score = new char[name.length];  
char[] grade = new char[name.length];  
char[] plus = new char[name.length];
```

# 학생별 성적처리(1차원 배열)

```
i = 0;
while (i < name.length) {
    System.out.printf("%d 번째 %s님의 국어 성적 : ", i+1, name[i]);
    kor[i] = keyboard.nextInt( );
    System.out.printf("%d 번째 %s님의 영어 성적 : ", i+1, name[i]);
    eng[i] = keyboard.nextInt( );
    System.out.printf("%d 번째 %s님의 수학 성적 : ", i+1, name[i]);
    math[i] = keyboard.nextInt( );
    if ((kor[i] >= 0 && kor[i] <= 100) &&
        (eng[i] >= 0 && eng[i] <= 100) &&
        (math[i] >= 0 && math[i] <= 100)) {
        System.out.println();
        i++;
    } else {
        System.out.print(" ERROR 성적은 0점에서 100점사이의 값");
        System.in.read( );
    }
}
```

# 학생별 성적처리(1차원 배열)

```
for (i = 0; i < name.length; i++) {  
    sum[i] = kor[i] + eng[i] + math[i];  
    avg[i] = sum[i] / 3.0f;  
}
```



# 학생별 성적처리(1차원 배열)

```
for (i = 0; i < name.length - 1; i++)          /* 총점순으로 정렬 */
    for (j = i + 1; j < name.length; j++) {
        if (sum[i] < sum[j]) {
            s_temp = hakbun[j];
            hakbun[j] = hakbun[i];
            hakbun[i] = s_temp;

            s_temp = name[j];
            name[j] = name[i];
            name[i] = s_temp;

            test = kor[j];
            kor[j] = kor[i];
            kor[i] = test;

            test = eng[j];
            eng[j] = eng[i];
            eng[i] = test;
        }
    }
```

# 학생별 성적처리(1차원 배열)

```
test = math[j];  
math[j] = math[i];  
math[i] = test;
```

```
test = sum[j];  
sum[j] = sum[i];  
sum[i] = test;
```

```
f_temp = avg[j];  
avg[j] = avg[i];  
avg[i] = f_temp;
```

```
}  
}
```

# 학생별 성적처리(1차원 배열)

```
for (i = 0; i < name.length; i++) {           /* 석차 부여 */  
    rank[i] = i + 1;  
}
```

```
for (i = 0; i < name.length - 1; i++) {       /* 동점자 처리 */  
    if (sum[i] == sum[i + 1])  
        rank[i + 1] = rank[i];  
}
```

# 학생별 성적처리(1차원 배열)

```
/*  
*          학번 기준으로 정렬          *  
*/  
for (i = 0; i < name.length - 1; i++)  
    for (j = i + 1; j < name.length; j++) {  
        if (hakbun[i].compareTo(hakbun[j]) > 0) {  
            s_temp = hakbun[j];  
            hakbun[j] = hakbun[i];  
            hakbun[i] = s_temp;  
  
            s_temp = name[j];  
            name[j] = name[i];  
            name[i] = s_temp;  
  
            test = kor[j];  
            kor[j] = kor[i];  
            kor[i] = test;  
        }  
    }
```

# 학생별 성적처리(1차원 배열)

```
test = eng[j];  
eng[j] = eng[i];  
eng[i] = test;  
test = math[j];  
math[j] = math[i];  
math[i] = test;  
test = sum[j];  
sum[j] = sum[i];  
sum[i] = test;  
f_temp = avg[j];  
avg[j] = avg[i];  
avg[i] = f_temp;
```

```
test = rank[j];  
rank[j] = rank[i];  
rank[i] = test;
```

```
}  
}
```

# 학생별 성적처리(1차원 배열)

```
/*  
*          국어 과목 학점 부여          *  
***/
```

```
for (i = 0; i < name.length; i++) {  
    if (kor[i] >= 90)  
        hakjum[i] = "수";  
    else if (kor[i] >= 80)  
        hakjum[i] = "우";  
    else if (kor[i] >= 70)  
        hakjum[i] = "미";  
    else if (kor[i] >= 60)  
        hakjum[i] = "양";  
    else  
        hakjum[i] = "가";  
}
```

# 학생별 성적처리(1차원 배열)

```
/*
 *          영어 과목 학점 부여          *
 */
for (i = 0; i < name.length; i++) {
    if (eng[i] >= 90)
        score[i] = 'A';
    else if (eng[i] >= 80)
        score[i] = 'B';
    else if (eng[i] >= 70)
        score[i] = 'C';
    else if (eng[i] >= 60)
        score[i] = 'D';
    else
        score[i] = 'F';
}
```

# 학생별 성적처리(1차원 배열)

```
/*
 *          수학 과목 학점 부여          *
 */
for (i = 0; i < name.length; i++) {
    switch (math[i] / 10) {
        case 10 : grade[i] = 'A'; plus[i] = '+';
            break;
        case 9  : grade[i] = 'A';
            break;
        case 8  : grade[i] = 'B';
            break;
        case 7  : grade[i] = 'C';
            break;
        case 6  : grade[i] = 'D';
            break;
        default : grade[i] = 'F';
            plus[i] = ' ';
    }
}
```



# 학생별 성적처리(1차원 배열)

```
switch (math[i] / 10) {  
    case 9 :  
    case 8 :  
    case 7 :  
    case 6 : switch ((math[i] % 10) - 5 >= 0 ? 1 : 0) {  
        case 0 : plus[i] = '0';  
            break;  
        case 1 : plus[i] = '+';  
    }  
}  
}
```

# 학생별 성적처리(1차원 배열)

```
System.out.println("*****");
System.out.println(" 학번    이름  국어  영어  수학  총점  평균  석차");
System.out.println("*****");
for (i = 0; i < name.length; i++) {
    if ((i != 0) && (i % 5 == 0))
        System.out.println();
    System.out.printf("%8s %5s %3d(%s) %3d(%c) %3d(%c%c)
                      %3d %.2f %3dWn", hakbun[i], name[i],
                      kor[i], hakjum[i], eng[i], score[i], math[i], grade[i],
                      plus[i], sum[i], avg[i], rank[i]);
}
System.out.println("*****");
}
```

# 학생별 성적처리(1차원 배열)

## ■ 석차 생성 (Sort 사용 없이)

```
for (i = 0; i < rank.length; i++) {  
    int count = 1;  
    for (j = 0; j < rank.length; j++) {  
        if (sum[j] > sum[i]) {  
            count++;  
        }  
    }  
    rank[i] = count;  
}
```

# 학생별 성적처리(1차원 배열)

## ■ 석차 생성 (Sort 사용 없이)

```
for (i = 0; i < sum.length; i++) {  
    rank[i] = 1;  
    for (j = 0; j < sum.length; j++) {  
        if (sum[i] < sum[j])  
            ++rank[i];  
    }  
}
```

# 학생별 성적처리(1차원 배열)

- 등수별로 출력하면서 학급 평균 이하를 위한 구분선을 추가해보자

```
for (i = 0; i < sum.length; i++){  
    total += avg[i];  
}
```

```
average = total / sum.length;
```

```
System.out.println("*****");  
System.out.println(" 학번      이름   국어 영어 수학 총점 평균 석차");  
System.out.println("*****");  
for (i = 0; i < hakbun.length; i++) {  
    if (avg[i] < average) {  
        System.out.println("-----평균 이하-----");  
        break;  
    }  
}
```

# 학생별 성적처리(1차원 배열)

```
System.out.printf("%8s %5s %3d %3d %3d %3d %.2f %3d\n",  
    hakbun[i], name[i],  
    kor[i], eng[i], math[i], sum[i], avg[i], rank[i]);  
}  
for ( ; i < hakbun.length; i++) {  
    System.out.printf("%8s %5s %3d %3d %3d %3d %.2f %3d\n",  
        hakbun[i], name[i],  
        kor[i], eng[i], math[i], sum[i], avg[i], rank[i]);  
}  
System.out.println("*****");  
System.out.printf("학급 평균 : %5.1f", average);  
}
```

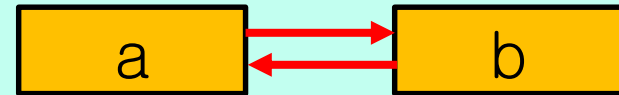
# 석차 구하는 방법

- 정렬 방법
  - 정렬한 후
  - 석차를 부여
  - 동점자 처리
- 정렬하지 않는 방법
  - 모든 등수를 1등으로 초기화
  - 반복하면서 보다 크면 등수를 올려주는 것

# 두 값을 바꾸는 방법

- 프로그램에서 가끔 두 변수의 값을 서로 바꿔야 할 때
- 아래와 같이 변수 A와 B의 값이 서로 다르게 존재한다고 가정해보자

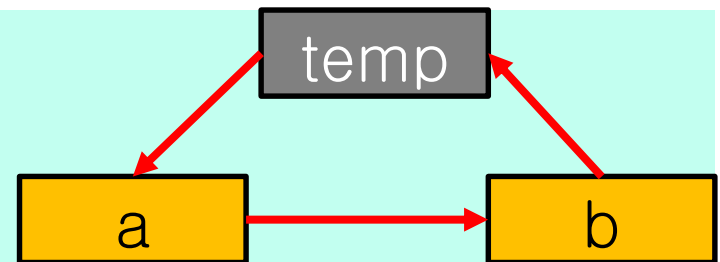
```
int a = 10;  
int b = 20;
```



- 임시 변수 사용 방법

- 가장 일반적인 방법으로 같은 데이터형의 새로운 변수(temp)를 하나 더 만들어 다음과 같이 서로의 값을 바꾼다

```
int temp = b;  
b = a;  
a = temp;
```





# 두 값을 바꾸는 방법

## ■ 사칙 연산 이용 방법

- 숫자형 데이터에 적용할 수 있음

- 새로운 임시 변수를 사용하지 않아도 됨

```
a += b;
```

```
b = a - b;
```

```
a -= b;
```

## ■ XOR(^) 연산 이용 방법

```
a = a ^ b;
```

```
b = a ^ b;
```

```
a = a ^ b;
```