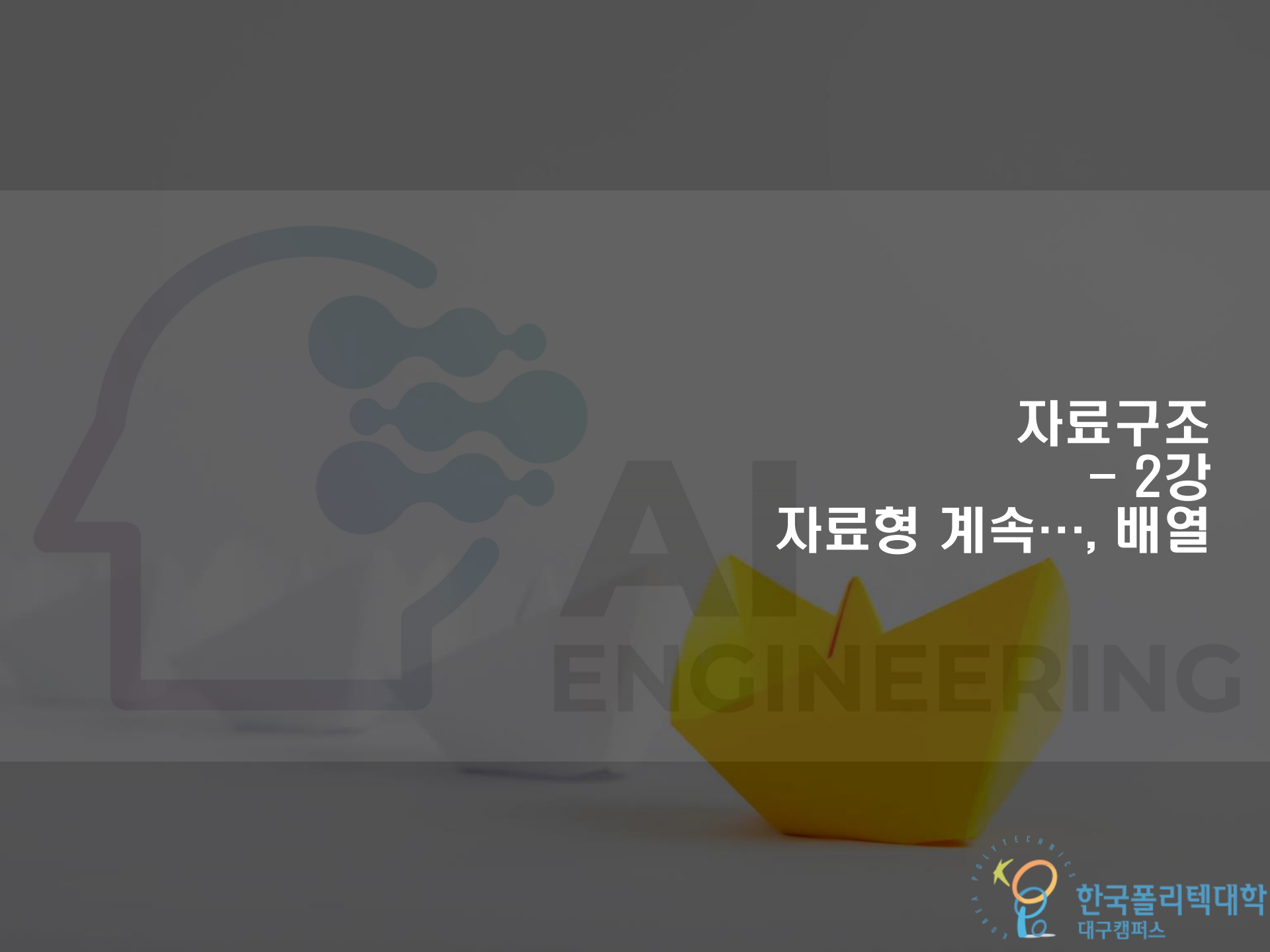


# 자료구조

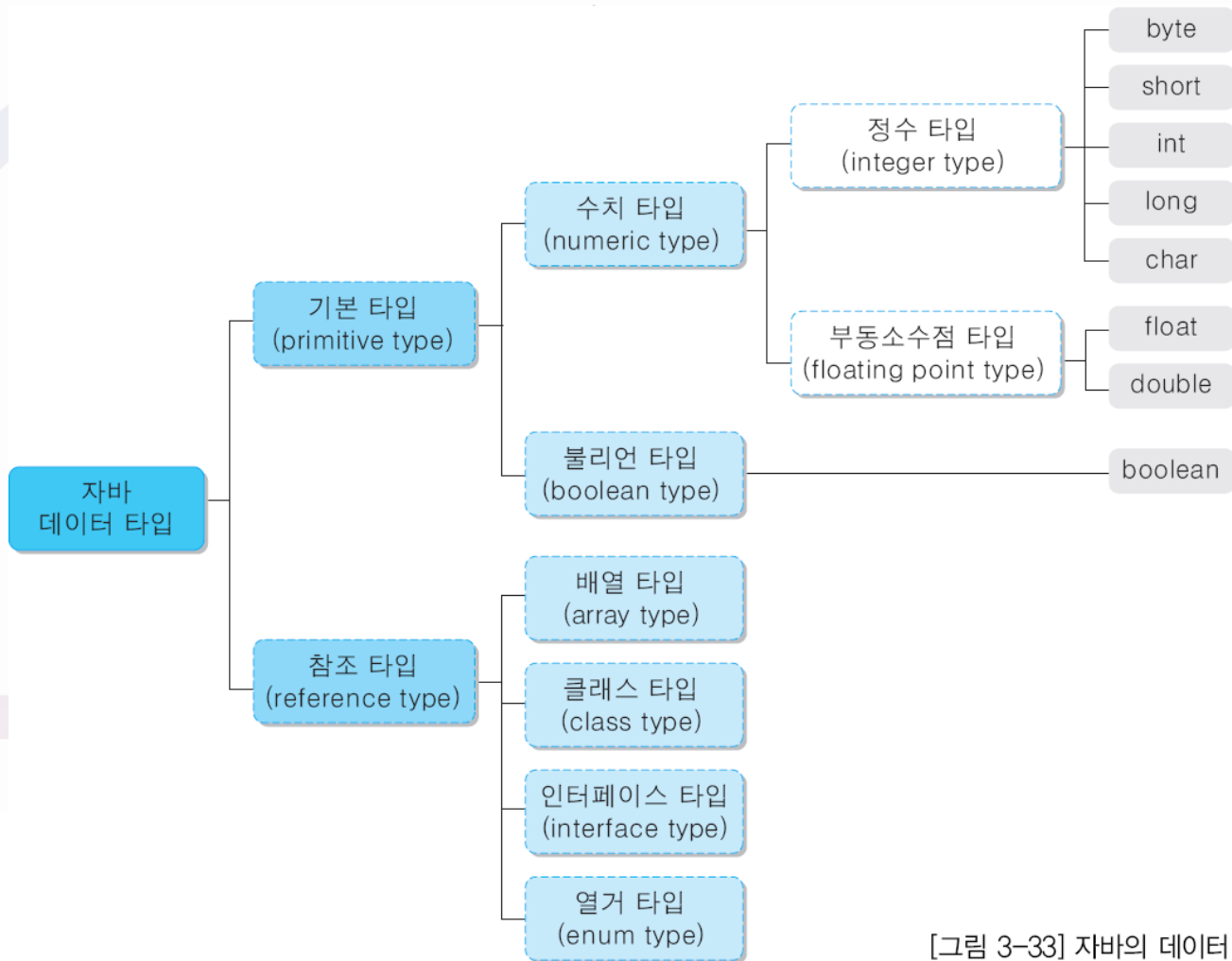
한국폴리텍대학 대구캠퍼스  
SI엔지니어링학과 강현우



# 자료구조 - 2강 자료형 계속..., 배열



# 지난 시간 복습



[그림 3-33] 자바의 데이터 타입

# 지난 시간 복습

## ◆ 배열 (Array) 타입

➤ 같은 데이터 타입의 변수들을 메모리에 연속으로 저장한 데이터들의 그룹

➤ 인덱스

✓ 배열의 요소를 구별하기 위해 사용하는 번호

✓ 0부터 시작

✓ 1차원 배열의 선언과 생성 및 사용

- 기본형 변수와 마찬가지로 배열 변수를 선언한 후에 사용

①	데이터 타입	배열이름	[ ] ;
	①	②	③
②	데이터 타입	[ ]	배열이름 ;
	①	③	②

① 데이터 타입 : 배열을 구성하는 각 요소에 대한 공통된 데이터 타입

② 배열 이름 : 배열 변수의 이름으로 사용자가 정한 식별자

③ 대괄호 : 보통의 변수가 아니라 배열 변수임을 나타내는 기호

# 구구단을 배열에 저장하여 보았다.

```
for (int i = 0; i < 9; ++i)
{
    gugudan_array[i] = dan * (i + 1);
}

for (int i = 0; i < 9; ++i)
{
    System.out.println(dan + " * " + (i + 1) + " = " + gugudan_array[i]);
    System.out.printf("%d * %d = %d\n", dan, i+1, gugudan_array[i]);
}
```

위 2개의 print문의 출력은 같다.  
편한 것을 사용하면 된다.

# 다차원 배열

✓ 다차원 배열의 선언과 생성 및 사용

- 다차원 배열 - 1차원 배열의 집합
- 다차원 배열의 크기 = 배열크기1 x 배열크기2

① 데이터 타입 배열이름 [    ] ;

①                      ②                      ③

② 데이터 타입 [    ] 배열이름 ;

①                      ③                      ②

new 데이터타입 [배열크기1] [배열크기2] ;

①                      ②                      ③

✓ 3차원 부터는 특별한 일이 아닌 이상 잘 안 쓴다.

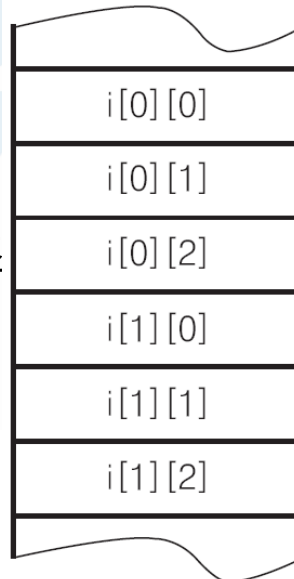
# 다차원 배열

✓ 다차원 배열의 선언과 생성 예

```
int i_arr2[][] = new int [2][3];
```

	열 번호 0	열 번호 1	열 번호 2
행 번호 0	i[0][0]	i[0][1]	i[0][2]
행 번호 1	i[1][0]	i[1][1]	i[1][2]

• 물리적 구조



# 다차원 배열 사용해보기

```
int row = 2, col = 3;
```

```
int [][] arr_2 = new int[row][col];
```

배열의 선언 및 크기 할당  
행과 열로 생각하면 된다.

```
arr_2[0][0] = 11;
```

```
arr_2[0][1] = 12;
```

```
arr_2[0][2] = 13;
```

```
arr_2[1][0] = 21;
```

```
arr_2[1][1] = 22;
```

```
arr_2[1][2] = 23;
```

인덱스로 접근한다.  
일반 변수처럼 사용이 가능하다

```
for (int i = 0; i < row; ++i)
```

```
{
```

```
    for (int j = 0; j < col; ++j)
```

```
    {
```

```
        System.out.printf("[%d]", arr_2[i][j]);
```

```
    }
```

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

```
}
```

배열의 값을 출력





# 구구단 계산 결과 저장하기

## ◆ 구구단 계산 결과를 2차원 배열에 저장하여 사용하고자 한다.

- 2차원 배열을 생성할 것
- 인덱스로 구구단의 결과를 접근할 수 있도록 할 것
- 메모리 낭비가 있어도 괜찮다.

## ◆ 인덱스를 사용하여 구구단의 계산결과를 바로 접근하도록 하여야 한다.

- ex) `gugudan[2][1] = 2`
- ex) `gugudan[9][9] = 81`

# 정답

10x10으로 잡는다.  
0단 1단은 버린다.

```
int row = 10, col = 10;
```

```
int [[[ gugudan = new int[row][col];
```

```
for (int i = 2; i < row; ++i)
```

```
{
```

```
    for (int j = 1; j < col; ++j)
```

```
    {
```

```
        gugudan[i][j] = i * j;
```

```
    }
```

```
}
```

```
for (int i = 2; i < row; ++i)
```

```
{
```

```
    System.out.printf("===== %d 단 =====\n", i);
```

```
    for (int j = 1; j < col; ++j)
```

```
    {
```

```
        System.out.printf("%d * %d = %d\n", i, j, gugudan[i][j]);
```

```
    }
```

```
}
```

계산 결과를 저장

새로 계산하지 않고  
저장된 값 사용

# Enum(열거) 타입

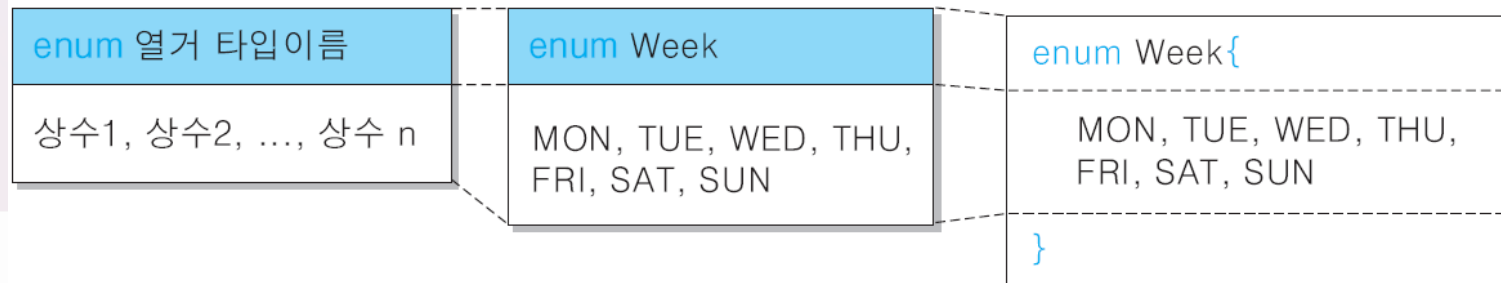
## ➤ 열거 타입

✓ 표현할 데이터가 정해진 수의 값만 가지고 있는 경우

- 예) 요일을 표현하는 값 = 월, 화, 수, 목, 금, 토, 일 7개

✓ 열거 타입 정의

- 키워드 `enum` 뒤에 열거타입 이름을 쓰고, 중괄호 사이에 선언할 상수를 콤마로 구분



# Enum(열거) 타입

## ➤ 열거타입 Week 정의

```
01 enum Week{  
02     MON, TUE, WED, THU, FRI, SAT, SUN  
03 }
```

## ➤ 열거 타입 변수의 선언과 사용

Week    myWeek;  
↑            ↑  
열거 타입    변수 이름

myWeek = Week.FRI;



# Enum의 사용

```
enum Week { MON, TUE, WED, THU, FRI, SAT, SUN }  
  
Week today = Week.FRI;  
  
if (today == Week.FRI)  
{  
    System.out.printf("%s 에는 청소를 합니다.", today);  
}
```

## ◆ 코드가 읽기 쉬워 진다.

➤ 숫자 0~6으로 요일을 표현했다고 상상해보라.

## ◆ 문자처럼 출력도 가능하다

# Enum의 사용

## ◆ 문자열로 enum 타입 변수 생성이 가능하다.

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

```
Week today = Week.valueOf("MON");  
System.out.println(today);
```

Enum 타입에 정의한 것과 같은  
문자열 이어야 한다.

```
enum FishType { Salmon, Bass }
```

```
FishType fish = FishType.valueOf("Salmon");  
System.out.println(fish);
```

# Switch / Case 문

## ◆ if-else 가 여러 개 이어질 때 사용

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

Week today = Week.FRI;

switch (today) // 변수
{
    case MON:
        System.out.println("월요일은 피곤하네요");
        break;
    case THU:
        System.out.println("화요일은 이론 수업이 많아요.");
        break;
    default:
        System.out.println("이런 값이 나오면 버그입니다.");
        break;
}
```

브레이크가 없으면  
다음 케이스를 또 비교한다.

NG

# Switch 문 응용

```
switch (today) // 변수
{
    case MON:
    case TUE:
    case WED:
    case THU:
        System.out.println("평일이 싫다...");
        break;
    case FRI:
        System.out.println("불금불금");
        break;
    default:
        System.out.println("이런 값이 나오면 버그입니다.");
        break;
}
```







# 기계학습 - 2강 가장 간단한 분류기 구현 Part 1.5



# 물고기 타입을 Enum형으로

```
enum FishType { Salmon, Bass }

int num_data = 318;
int index = 0;
int [] length_arr = new int[num_data];
FishType [] fish_data = new FishType[num_data];
String line = br.readLine();

while(true)
{
    line = br.readLine();
    if (line == null) break;

    String [] parse = line.split(",");

    fish_data[index] = FishType.valueOf(parse[0]);
    length_arr[index] = Integer.parseInt(parse[1]);
    ++index;
}
```

Parse[0]이 뭐였죠?



# 길이 모델 만들기

◆ 길이  $x$  보다 크면  
→ bass

◆  $x$  이하이면 →  
salmon

```
int model_length = 10;
int n_hit = 0, n_miss = 0;

for (int i = 0; i < num_data; ++i)
{
    FishType ans;

    if (length_arr[i] > model_length) {
        ans = FishType.Bass;
    }

    else {
        ans = FishType.Salmon;
    }

    if (ans == fish_data[i]) {
        ++n_hit;
    }
    else {
        ++n_miss;
    }
}

System.out.printf("hit = %d, miss = %d\n",
n_hit, n_miss);
```

# 실습

- ◆ 앞의 코드를 사용하여 분류율이 가장 높은  $x$ 를 찾아보라!

