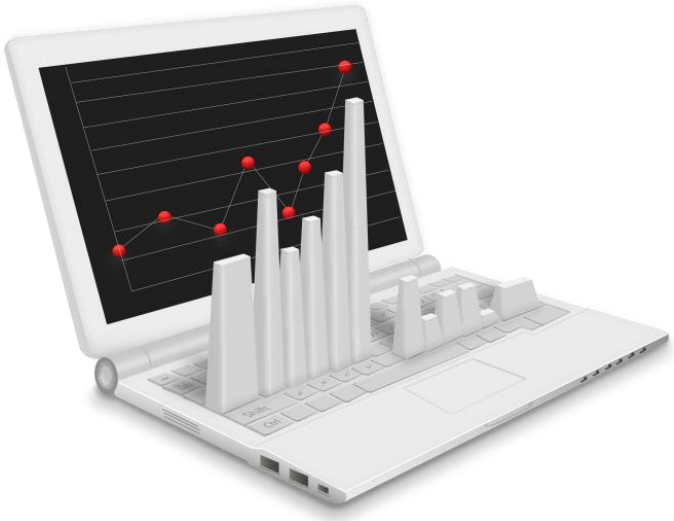
프로그래밍 언어론

문자열과 Ordinal(순서) 타입

컴퓨터공학과

조은선



충남대학교
CHUNGNAM NATIONAL UNIVERSITY

문자열 타입과 Ordinal 타입

학습목표

- 문자열 타입과 Ordinal(순서) 타입에 대해 이해한다.

학습내용

- 문자열 타입의 개념
- 문자열의 길이와 연산
- Ordinal타입의 개념
- Enumeration 타입과 Subrange타입 개념

목 차

- 들어 가기
- 학습하기
  - 문자열 타입
  - 문자열 길이
  - 문자열 연산
  - User-Defined Ordinal 타입
  - Enumeration 타입
  - Subrange 타입
- 평가하기
- 정리하기

알고가기

다음 C 프로그램에 대한 설명으로 틀린 것은?

```
char * h = "Hello";
printf("%s", h);
```

① 화면에 Hello 라고 출력한다.

② 변수 h 선언부는

```
char h[5] = {'H', 'e', 'l', 'l', 'o'};
```

라고 해도 결과는 동일하다.

③ %s는 출력하는 데이터의 타입이 문자열임을 의미한다.

④ printf("%s", h)는 printf(h)와 결과가 동일하다.

Ⅰ 문자열(String) 타입

- 문자들의 열을 나타냄 – 필수
- 문자열 타입은 언어마다 나타내는 방법이 다름
  - 문자열 타입이 기본 타입인 경우

    Ada, FORTRAN
  - 문자열 타입이 일련의 문자타입으로 구성됨

    C: **`char*`**
  - 또 다른 방법으로 문자열 타입이 구성됨

    Java: **`String/StringBuffer`**

| 문자열 길이

종류

고정 길이 (FORTRAN, ADA, COBOL ...)

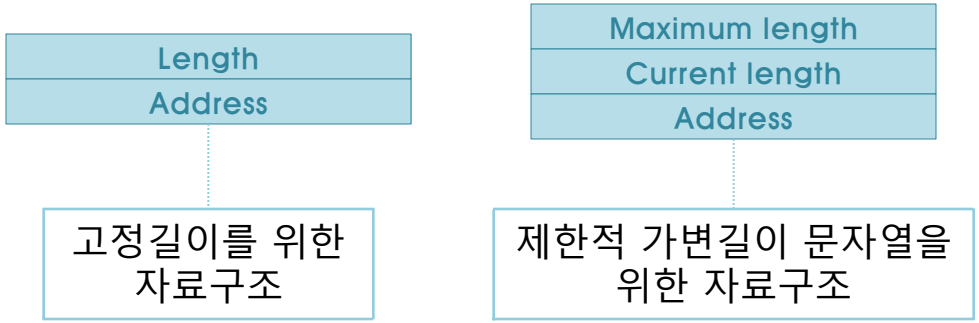
```
CHARACTER (LEN = 15) NAME;
```

제한적 가변 길이 (C, C++)

```
char c[10]; c[0] = 'a'; c[1]='\0';
```

가변길이 (SNOBOL4, Perl, JavaScript)

- 실제 사용되는 양을 알고 있으므로 메모리 사용이 효율적



문자열 연산

가능한 연산

Assignment, 비교 (=, >, etc.), 붙이기, 부문자열추출, 패턴매칭 ...

예

문자열 붙이기, 부문자열추출, (Ada)

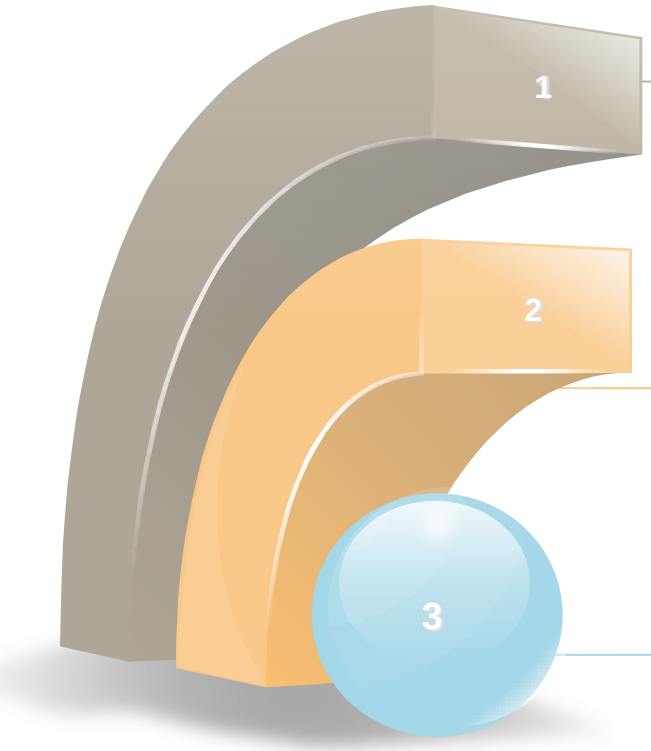
- `N := N1 & N2` (붙이기)
- `N(2..4)` (추출)

예

패턴매칭 (Perl, JavaScript, PHP)

- `/[A-Za-z][A-Za-z\d]+/`

User-Defined Ordinal 타입(사용자 정의 순서 타입)



User-Defined Ordinal 타입

- 그 값들이 자연수에 대응시킬 수 있을 때 사용
- 기본 순서 타입? (사용자 정의가 아닌)
  : 정수, 문자, 참거짓 …

User-Defined Ordinal 타입의 종류

- Enumeration 타입(열거 타입)
- Subrange 타입(부분 범위 타입)

장점

- 가독성 향상
- 타입 검사로 런타임 오류 방지

Enumeration 타입(열거 타입)

Enumeration 타입

- 모든 값들을 열거하여 타입을 정의함
  > 그 값들은 주로 상수들

예

```
enum workdays {MON=1, TUE=2, WED=3, THU=4, FRI=5};
// 초기값을 주지 않으면 0부터 시작
enum workdays d TUE;
d++; printf("%d\n",d); // 가능
```

논 점

-같은 상수가 두 개 이상의 enumeration 타입에 들어갈 수 있을까?
-지정되는 값이 상수 범위를 넘어가는 경우 (위 예의 d=10; 등)
오류처리?
- 수치값과 enumeration 타입간의 자유로운 변환 허용?

예

- **PASCAL – input/output에는 못사용됨, 상수 재사용 안됨**
- **Ada – 상수 재사용가능 (상황으로 타입 판단함)**
- **C/C++ - PASCAL하고 동일. input/output에는 정수로 변환되어 사용**
- **Java, C# - C/C++과 동일 (Java 는 1.5부터) 하며,**
  Enumeration 타입 정의에 사용 가능한 연산들이 추가될 수 있음

Subrange 타입(부범위타입)

Subrange 타입

- 수의 연속적인 범위를 타입으로 지정할 수 있도록 함

예

```
type pos = 0 .. MAXINT;
```

- for 문, 배열 인덱스 등 범위 검사가 중요한 곳에서 사용 (PASCAL, Ada)
- 장 점: 가독성, 신뢰성 등 향상
- 구 현: 정수타입 등 상위 타입과 동일하게 처리하면서 동시에 범위 검사를 위한 코드를 삽입
- 논 점: 꼭 데이터타입 형태로 지원해야 하는가?
  - 대부분의 언어가 요즘은 제공하지 않음

평가하기

마지막으로 내가 얼마나 이해했는지를 한번 확인해 볼까요?

총 3문제가 있습니다.

START

평가하기 1

1. 다음 문자열 표현 방식 중 컴파일 전에 문자열의 길이를 정하고, 실제 수행 할 때 사용되는 메모리 영역에 대해서는 전혀 고려하지 않는 방식은?

① 고정 길이 문자열

② 제한적 가변 길이 문자열

③ 가변 길이 문자열

④ C의 문자열

확인

평가하기 2

2. User-defined ordinal 타입의 특징이 아닌 것은?

① 가독성 향상

② 타입 검사로 코드의 신뢰성 향상

③ 처리 속도 향상이 목적

④ 그 값들이 자연수에 대응 시킬 수 있을 때 사용

확인

평가하기 3

3. 나타날 수 있는 모든 데이터를 열거함으로써 데이터 타입을 정의할 수 있도록 하는 타입은?

① Subrange 타입
② 제한적 가변길이 문자열
③ 가변길이 문자열
④ Enumeration 타입

확인

정리하기

- **문자열 타입**
  Assignment, 비교 (=, >, etc.), 붙이기, 부문자열추출, 패턴매칭 등의 문자열을 위한 연산들이 있으며, 문자열 길이에 따른 메모리 사용 방법은, 고정길이, 제한된 가변길이, 가변길이 등이 있다.

- **Ordinal 타입**
  그 값들이 자연수에 대응시킬 수 있을 때 사용하여, 가독성과 신뢰성을 향상시킨다.
  데이터를 나열하는 Enumeration 타입, 수의 범위를 지정하는 Subrange 타입이 있다.

만일 사용된 프로그래밍 언어에 ordinal 타입이 없다면,
가독성과 신뢰성을 최대한 지키면서
구현할 수 있는 효과적인 방안을 찾아보자..