

Data Types

국민대학교 컴퓨터공학부
강 승 식

Topics



- Primitive Data Types
 - int, char, float, boolean
- Character String
- User-Defined Ordinal Types
- Array, Associative Arrays
- Record, Union
- Pointer, Reference
- Type Equivalence, Strong Typing

더이상 쪼갤수 없음

Portran 4 쓸만함 1950년대 부터 성행(정수, 실수)
Portran 77 (77년) - 문자타입이 들어감 총 3가지

string type은 primitive data type인가?
파이썬은 O
C언어는 X
반반이다.

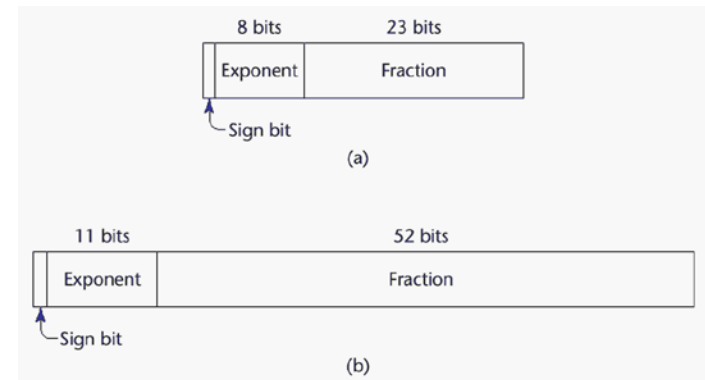
long, short는 같은타입인가 다른타입인가?
기준을 타이트하게하면 다른타입
기준을 느슨하게하면 같은타입

Definition of a Type

- A set of data
 - $-32768 \sim 32767$
5bit
- A set of operations on those data
 - $+, -, *, /, ==$
- Descriptor
 - Collection of attributes of a variable

Primitive Data Types

- Reflections of a hardware?
- Integer : byte, short, int, long
- Real : float, double
 - c언어 float로 하면 정확하지않음.. 계산용이아님
 - 정확한계산할라면 portran 쓰셈
- Character : UCS-2, UCS-4
- String
- Boolean, Complex, Decimal, Date



String type

strcpy 함수 쓰려면 <string.h> 포함해야함

- A special kind of array, or a primitive type?
- Operations
 - Assignment, copy, comparison, concatenation
 - Substring reference, pattern matching

- C : **character array**

- strcpy(s, "This is a string");
 - strcat(s, " and it needs <string.h>");

String copy
virus 때문에 쓰진 말자
요즘 safe coding이 유행함
string copy할때 긴걸 copy하게 해서 virus를 뽑는다

- Java : **string class**
- SNOBOL4, Python, Perl : **string type**
 - s = "This is a string"
 - s = "this" + "is" + "a" + "string"
 - t = 2 * "string"

string type이 primitive면 이렇게 치환이 된다.

String Length

- **Static length**: Java's String class
 - Compile-time descriptor

Static string
Length
Address

- **Limited dynamic length**: C/C++

글자 크기를 끝에 null 넣어서 크기 확인

Limited dynamic string
Maximum length
Current length
Address

- **Dynamic length**: Python, JavaScript
 - Run-time descriptor

자바스크립트는 글자 크기를 지정해서 하고

- Ada supports all three options

Ada -> C, C++, Java 기능 둘다 가지고 있음

User-defined Ordinal Types

- Primitive ordinal type in Java
 - Integer, char, Boolean
- Enumeration type
 - `enum days { mon, tue, wed, thu, fri, sat, sun};`
 - `enum colors { white, yellow, red, green, blue, black };`
- What if enum values are coerced to integers?

Subrange Types

- 13..19 is a subrange of integer
- Ada's design

```
type Days is (mon, tue, wed, thu, fri, sat, sun);  
subtype Weekdays is Days range mon..fri;  
subtype Index is Integer range 1..100;
```

```
Day1: Days;
```

```
Day2: Weekday;
```

```
Day2 := Day1;
```

Day1은 Day라는 타입

Day2는 Weekday라는 타입

그 범주 안에 들어가면 에러 안남

월~금은 가능..? 토요일 하면 에러남?

Array Types

- Index syntax
 - () – Fortran, PL/I, Ada
 - [] – C, Java, ...

알콜?
int a[100]
1~100
만약 1~100 아니면 바로 out
반면 c언어는
0~99
자유분방하다.

- Index type
 - Integer only – Fortran, C
 - Integer, enumeration, Boolean, char -- Ada

c언어에서의 enum 타입은 정수임

무늬만 월화수목금토일이고(무늬만 문자) 실제로는 01234567임 – c언어

Array Initialization

- C-based languages

- `int list [] = {1, 3, 5, 7}`
- `char *names [] = {"Mike", "Fred", "Mary Lou"};`

- Ada

Ada 스타일 : 정수 배열 인덱스는 1~5까지 크기가 5인

- `List : array (1..5) of Integer :=
(1 => 17, 3 => 34, others => 0);`

- Python

- List comprehensions

파이썬은 선언이 필요없음

x의 2승 0~12까지 각각들의 제곱 0 1 4 9 16 ---
3나머지 0인거만 list에 넣어라

```
list = [x ** 2 for x in range(12) if x % 3 == 0]  
puts [0, 9, 36, 81] in list
```

Array Operations

- APL
 - powerful array processing for vectors and matrixes
- Assignment, catenation, slice in Python
 - $a = [1, 2, 3]$
 - $b = a + [4, 5, 6]$
 - $b[2:4]$
- + operator in Fortran
 - $a = b + c$

Multi-dim. Arrays

- Compile-time descriptors

c언어는 1차원밖에 없음

일반적으로는 1차원 다차원 구별되게 씀 언어마다 다르긴함
다차원 배열의 descriptor

Array
Element type
Index type
Index lower bound
Index upper bound
Address

정수배열이나 문자배열이나

정수나 불린이나 문자열이나

Single-dimensioned array

Multidimensioned array
Element type
Index type
Number of dimensions
Index range 1
⋮
Index range n
Address

차수가 몇차원?

Multi-dimensional array

Associative Arrays

연관배열은 지원하는 언어가 있고
없는 배열이 있음

- Index is not restricted to integers
- Key-value pairs
 - `dic = {'name':'kim', 'phone':'010-1234-5678', 'major':'CS'}`
 - `dic['name']`

Record Types

구조체

COBL은 기본적으로 레벨 넘버가 있음 01 레벨 02레벨 05레벨 등 레벨이 꼭
순서대로 1 2 3 일 필요는 없음

- COBOL: level numbers

```
01 EMP-REC.
```

```
02 EMP-NAME.
```

```
05 FIRST PIC X(20). 여기서 () 배열임 - 문제 20개 까지 표현가능
```

```
05 MID PIC X(10).
```

```
05 LAST PIC X(20).
```

```
02 HOURLY-RATE PIC 99V99. hourly-rate는 시급을 나타낸 것임
```

v자 표시된게 소수점자리를 나타냄

- Ada

00.00 ~ 99.99

decimal 타입과 정수형 타입의 차이점?

```
type Emp_Rec_Type is record
```

100넘어가면 예러남 integer i=23

23이 2진수로 바뀜.

10111 이게 인터저임

decimal은 이게 아님

이건 4비트 4비트

2 3

10 11

이렇게 표현함

first는 필드명 First: String (1..20);

```
Mid: String (1..10);
```

```
Last: String (1..20);
```

```
Hourly_Rate: Float;
```

```
end record;
```

```
Emp_Rec: Emp_Rec_Type;
```

- Record Field References

- COBOL

- field_name OF record_name_1 OF ... OF record_name_n

- Others

- record_name_1.record_name_2. ... record_name_n.field_name

- Operations

- Assignment, comparison, initialization

- 기본적으로 지원이 됨

- 구현 어려움

- 초기화 지원함

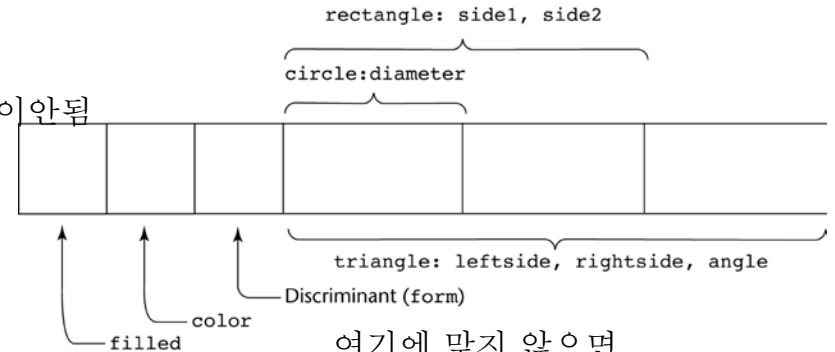
Union Types

c언어스타일, 안c언어스타일로 나뉘

- Free unions 공유하는 타입이 다른 체크가 안됨
 - No type checking

- Discriminated unions: Ada
 - Type checking

```
type Shape is (Circle, Triangle, Rectangle);
type Colors is (Red, Green, Blue);
type Figure (Form: Shape) is record
  Filled: Boolean; 변수이름 Form type은 Shape
  Color: Colors;
  case Form is
    when Circle => Diameter: Float;
    when Triangle =>
      Leftside, Rightside: Integer;
      Angle: Float;
    when Rectangle => Sidel, Side2: Integer;
  end case;
end record;
```



여기에 맞지 않으면
타입에러를 일으킴(?)
아무튼 큰 차이점은
discriminated가 있느냐 없느냐의
차이임

Pointer Types

- Pointers are used for

- Dynamic memory management
- Indirect addressing
- Linked list

```
int a[100]
int *p = a;
int *q;
q=a
```

- Pointers in C/C++

- Extremely flexible
- Point at any variable
- Used for dynamic storage management and addressing
- Pointer arithmetic is possible: $p++$, $*(p+i)$, $p-q$, $p < q$
- `void *` can point to any type

Reference Type

링크드 리스트 기능은 뺄 수 없음

C++은 c와 자바의 중간임

자바는 포인터개념 없음

C++은 포인터 쓸수는 있지만 대신 레퍼런스 쓰는게 좋음

- Special kind of pointer type
- Dangling pointers and dangling objects(garbage)
 - Reference counter
 - Garbage collection
 - Mark-sweep

가비지가 발생할 수 있다

Type Checking

- Compatible type
 - Name type equivalence
 - Structure type equivalence
- Type error
- Strong typing

- Consider the problem of two structured types:
 - Are two record types equivalent if they are structurally the same but use different field names?
 - Are two array types equivalent if they are the same except that the subscripts are different?
(e.g. [1..10] and [0..9])
 - Are two enumeration types equivalent if their components are spelled differently?
 - With structural type equivalence, you cannot differentiate between types of the same structure