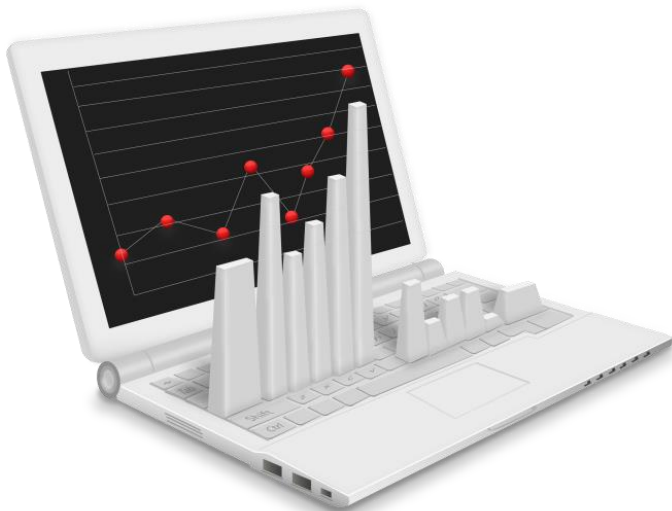


# 프로그래밍 언어론

## 선택문(Selection Statement)

컴퓨터공학과

조은선



## 선택문 (Selection Statement)

### 학습 목표

프로그래밍 언어의 구문요소 중에서  
문장 실행 흐름을 제어하는 Control structure (제어구조)  
에 대해 알아보고 선택문 (selection statement)의 의미를  
공부한다.

### 학습 내용

- Control structure 의 설계
- 선택문(selection statement)



# 목 차

- 들어가기
- 학습하기
  - Control Structure 설계
  - 선택문(Selection Statement)
    - 단일선택, 양자택일, 다중선택
- 평가하기
- 정리하기



# 알고가기



**x=1, y=1 일때 다음 C 프로그램이 출력하는 값은?**

보기

```
if (x < 0)
    if (y < 0)
        printf("l");
else
    printf("r");
```

- ① l
- ② r
- ③ lr
- ④ 아무것도 출력하지 않는다.

확인



# | 제어구조 (Control Structure)

## Control flow (제어 흐름)의 종류

- 수식 내 연산 순서
- 프로그램 문장들의 수행 순서 ✓ (이것을 "control structure (제어 구조)"라 한다.)
- 단위 프로그램 호출 순서

## Control structure(제어 구조)의 구성

제어문(Control statement) : 선택문(Selection statement), 반복문(Iteration)

복합문(Compound statement) : 제어를 수행하는 문장의 모음

여러 문장을 하나의 문장으로 추상화  
ALGOL의 begin.. end 에서 시작함  
C에서는 {...} 형태이고 선언을 포함가능

## 발전

Fortran I : IBM704 하드웨어를 직접 흉내

→ 1960년대 결과 : 모든 순서도는 if-then-else와 while-do로 작성될 수 있고, goto는 불필요

## 설계 쟁점

control structure는 단일 입구만을 가져야 하나? if-then-else와 while-do 외에 어떤 제어문이 더 필요한가? ..



# | 선택문(Selection Statement)

## 단일 선택(Single-way Selector)

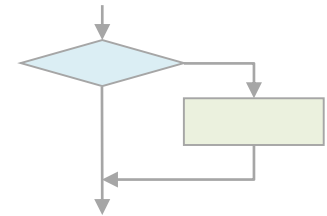
→ 단일 statement만 허용(FORTRAN)

```
IF (Boolean_expr) single_statement
```

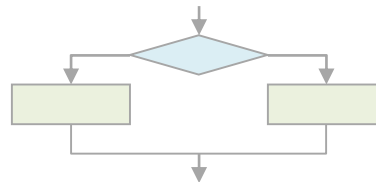
→ Compound statement 허용 (대부분의 언어)

```
Algol : if (Boolean_expr) then begin ... end
```

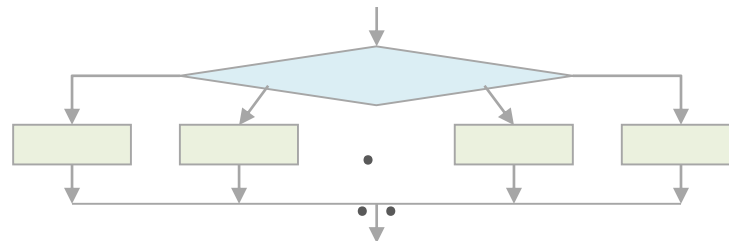
```
C : if (Boolean_expr) { ... }
```



## 양자 택일(Two-way Selector)



## 다중 선택(Multiple Selector)



# | 양자 택일(Two-way Selector)

## 설계 쟁점

>> 조건식의 형태와 타입은? then 과 else 절은 어떻게 명세하는가? 중첩 선택문의 의미는 어떻게 명세하는가?

### Algol 60

```
if (Boolean_expr)
  then statement
  else statement
```

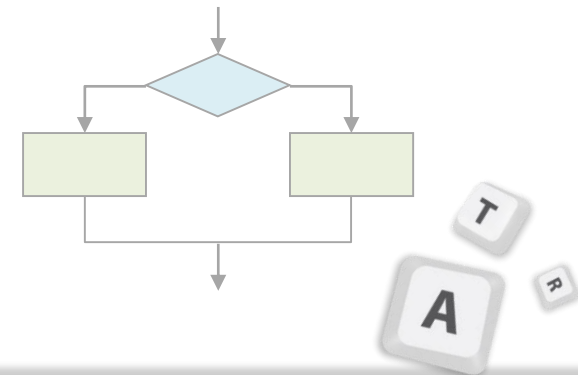
> **statement** 는 단일(혹은 복합) 문장

### C

```
if (Boolean_expr)
  statement;
else
  statement;
```

### Ada

```
if (Boolean_expr)
  statement; ... ; statement의 나열
else
  statement; ... ; statement의 나열
end if;
```



# | Nested-If (중첩 선택문)

```
if (expr-1)
  if (expr-2)
    stmt-a;
  else
    stmt-b;
```

“어떤 if가  
else를 취하는가?”

```
if (expr-1)
  if (expr-2)
    stmt-a;
  else
    stmt-b;
```

**해결법1 (Algol 60)**  
직접 중첩은 불허

```
if (expr-1) then
  begin
    if (expr-2)
      then stmt-a
      else stmt-b
  end
```

```
if (expr-1) then
  begin
    if (expr-2) then stmt-a
  end
else stmt-b
```

**해결법2 (Ada, Fortran 90)**  
닫기 단어(closing word) **end if** 사용

```
if (expr-1) then
  if (expr-2) then
    stmt-a
  else
    stmt-b
  end if
end if
```

```
if (expr-1) then
  if (expr-2) then
    stmt-a
  end if
else
  stmt-b
end if
```

**해결법3 (C, Java)**  
가장 가까운 if

```
if (expr-1)
  if (expr-2)
    stmt-a;
  else
    stmt-b; // 위 예
```





# | 다중 선택(Multiple Selector)

## 설계 쟁점

- > 조건식의 형태와 타입은?
- > 선택가능한 요소를 어떻게 명세하는가?
- > 제시되지 않은 값이면 어떤 결과가 되는가?

## → 초기 다중 선택문 : 3항 선택문 (FORTRAN)

IF (산술식) N1, N2, N3

- 0보다 작으면 N1로, 같으면 N2로, 크면 N3으로 분기 이동

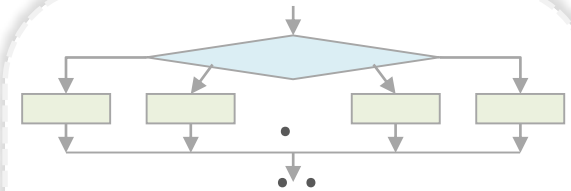
문장 번호의 계산에 의한 분기 (FORTRAN)

GOTO (10, 20, 30, 40), K

K = 1이면 10번 명령문을 실행,  
2이면 20번, 3이면 30 번,  
4면 40번 명령문 실행

ASSIGN 10 TO I  
GOTO I (10, 20, 30)

10 번 문번호로 실행을 옮긴다.  
I 값이 10, 20, 30 중 하나가  
아니면 오류 남.



```

IF (...) 10, 20, 30
10 ...
   ...
   goto 90
20 ...
   ...
   goto 90
30 ...
   ...
90 ...
   ...

```



# | 다중 선택(Multiple Selector)

## 설계 쟁점

- > 조건식의 형태와 타입은?
- > 선택가능한 요소를 어떻게 명세하는가?
- > 제시되지 않은 값이면 어떤 결과가 되는가?

## → 초기 다중 선택문 : 3항 선택문 (FORTRAN)

IF (산술식) N1, N2, N3

- 0보다 작으면 N1로, 같으면 N2로, 크면 N3으로 분기 이동

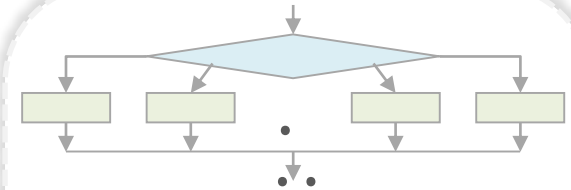
문장 번호의 계산에 의한 분기 (FORTRAN)

GOTO (10, 20, 30, 40), K

K = 1이면 10번 명령문을 실행,  
2이면 20번, 3이면 30 번,  
4면 40번 명령문 실행

ASSIGN 10 TO I  
GOTO I (10, 20, 30)

10 번 문번호로 실행을 옮긴다.  
I 값이 10, 20, 30 중 하나가  
아니면 오류 남.



```

IF (...) 10, 20, 30
10 ...
   ...
   goto 90
20 ...
   ...
   goto 90
30 ...
   ...
90 ...
   ...

```



# | C, C++, Java의 Switch

## C, C++, Java의 Switch

```
switch (expression) {  
    constant_expression_1 : statement_1;  
    ...  
    constant_expression_n : statement_n;  
    [default: statement_n+1;]  
}
```

### ⇒ 설계 선택

- statement\_k 는 문장 나열, 블록, 복합문이 될 수 있음
- expression은 모든 ordinal 타입이 가능
- 여러 개의 선택가능 요소가 실행될 가능성이 있음
  - 프로그래머는 각 선택가능 요소마다 break 문을 제공해야 함
- 제시되지 않은 값을 위해 default문을 제공

신뢰성과 유연성 사이의 trade-off



# | Pascal의 case

## Pascal의 case

**Hoare의 Algol W에서 현대의 다중 선택문 형태를 제시**

```
case expression of
    constant_list_1 : statement_1;
    ...
    constant_list_n : statement_n
end
```

### ⇒ 설계 선택

- ⇒ expression은 어떤 타입이든 됨 (int, Boolean, char, enum)
- ⇒ 요소(statement\_k)는 단일 혹은 복합이 될 수 있음
- ⇒ 한 구문의 실행에서는 단 하나의 선택가능 요소만 실행됨
- ⇒ Wirth의 Pascal에서, 제시되지 않은 값의 결과는 정의되지 않음
  - ➡ 지금의 많은 파생언어는 otherwise 혹은 else 절을 가짐



# | if를 이용한 다중 선택

## if를 이용한 다중 선택

>> else-if를 사용한 양자 선택문의 직접 확장을 통해 다중 선택문을 표현할 수 있다.

- Algol 68, Fortran 90, Ada

### Ada

```
if (expr-1)
  then ...
  else if (expr-2)
    then ...
    else if (expr-3)
      then ...
      else ...
    end if
  end if
end if
```

```
if (expr-1)
  then ...
  elsif (expr-2)
    then ...
  elsif (expr-3)
    then ...
  else ...
end if
```

더 가독성 있음



# 평가하기

마지막으로 내가 얼마나 이해했는지를 한번 확인해 볼까요?  
총 2문제가 있습니다.

START



# 평가하기 1

## 1. 다음 중 선택문(selection statement)에 대한 설명으로 맞는 것은?

- ① 선택문은 제어문(control statement)이 아니다.
- ② 선택문은 하나의 실행 경로만을 갖는다.
- ③ C의 switch 문은 다중 선택 (multi-way selector) 이다.
- ④ 각 언어의 if문은 else의 모호성 때문에 nesting(중첩)을 허용하지 않는다.

확인



## 평가하기 2

### 2. 다음 중 FORTRAN, PASCAL, Ada 의 제어구조에 대한 설명으로 틀린 것은?

- ① FORTRAN은 중첩 if 문에서 짝이 모호한 else가 발생하면 가장 가까운 if 문과 연결한다.
- ② FORTRAN의 IF (산술식) N1, N2, N3 는 산술식이 0보다 작을 때 N1으로 분기한다.
- ③ PASCAL의 다중 선택문은 case expression of 문으로, 산술식이 expression으로 올 수 있다.
- ④ Ada에서는 else if의 줄임으로 elsif 를 제공하는데 else if를 띄어쓰는 경우보다 가독성이 좋다.

확인





## 정리하기

- ➡ 제어문(control statement)은 선택문(selection statement)과 반복문(iterative statement)로 이루어진다.
- ➡ nested if의 모호한 else 문제는 closing word를 사용하거나 가장 가까운 if에 else를 연결하여 해결하는 경우가 많다.
- ➡ 선택문(selection statement)에는 단일선택, 양자택일, 다중 선택이 있다.



“ 강의를 마치겠습니다. 수고하셨습니다. ”

