

1 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

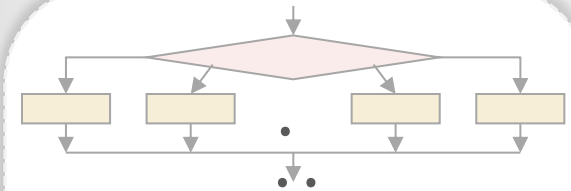
문장 번호의 계산에 의한 분기 (FORTRAN)

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)

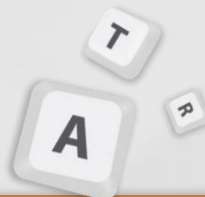
10 번 문번호로 실행을 옮긴다.
I 값이 10, 20, 30 중 하나가
아니면 오류 남.



```

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

```





◉ iterator

Iterator(반복연산자)

개 념

- ➔ 순서가 있는 데이터 구조의 모든 노드를 방문해서
- ➔ 각 노드에 사용자가 제공하는 연산을 수행

•수동 iterator

- ➔ 사용자가 노드를 처리하는 메소드를 해당 객체에 제공. 객체는 모든 노드에 순서대로 메소드를 적용

➡ 예제) Smalltalk

```
class List {
    Object[] listElements = new Object[size];
    ...
    public void do( Function userOperation ) {
        for (int k=0; k < listElements.length(); k++)
            userOperation( listElements[k] );
    }
}

List grades = new List();
aFunction = (item){ print(item) }; // 수행할 메소드
...
grades.do( aFunction );           // 메소드 전달
```



능동 iterator

•능동 iterator

➔ 사용자가 객체의 다음 원소로 이동하고, 처리하는 것을 책임짐.

➔ 예제) Java, C++

```
List grades = new List();  
  
Iterator gradeList = grades.iterator();  
  
while ( gradeList.hasNext() ) {  
  
    listItem = gradeList.next();    // 다음 원소로 이동  
  
    print ( listItem );            // 원소의 처리  
  
}
```

