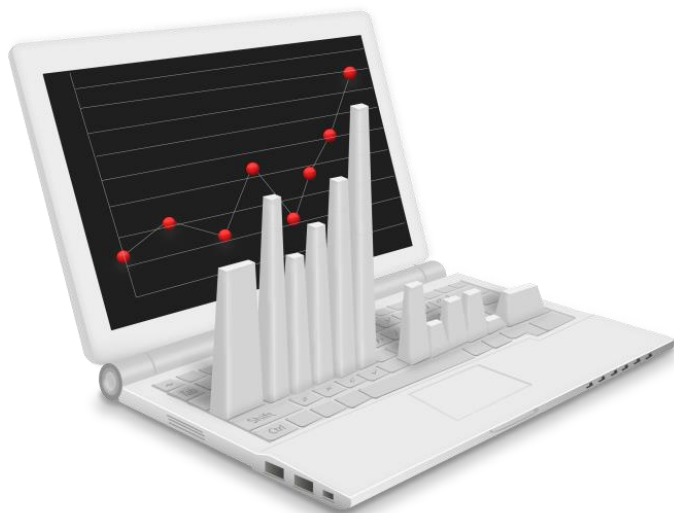


프로그래밍 언어론

기타 여러 가지 추상화

컴퓨터공학과

조은선



기타 여러 가지 추성화

학 습 목 표

- 보편적인 추상 데이터타입외의 다양한 데이터 추상화 기법에 대한 개념과 쓰임을 이해한다.

학 습 내 용

- 인자를 가지는 추상 데이터 타입
- 명칭 캡슐화
- 기타 캡슐화



목 차

- 들어가기
- 학습하기
 - 인자를 가지는 추상 데이터 타입
 - Name Encapsulation
 - 기타 Encapsulation
- 평가하기
- 정리하기



알고가기



추상 데이터 타입에 대한 설명과 관련이 가장 적은 것은?

- ① 함수와 데이터를 묶어 encapsulation한다.
- ② 외부로 구현과 관련된 부분은 노출하고 인터페이스는 감추는 기능이 있다.
- ③ Ada, C++, Java, C# 등에서 지원한다.
- ④ 내장된 기본 타입들은 넓게 보면 추상 데이터 타입이다.

확인



| 인자를 가지는 추상 데이터 타입

인자를 가지는 추상 데이터 타입 (Parameterized Abstract Data Type)

추상데이터 타입 자체가 고정되어 있는 것이 아니라 인자를 받아서 생성됨

→ Ada 예

```
package INT_STACK is new GENERIC_STACK(100, INTEGER);  
package FLOAT_STACK is new GENERIC_STACK(500, FLOAT);
```



I 인자를 가지는 추상 데이터 타입의 예

→ C++ 예

- > 템플릿 (template) : 타입 인자를 받아 컴파일 동안 보통 클래스로 변환 시킴

```
template <class Type>
class Stack {
    Type * stackPtr;
    ...
    Stack() {
        stackPtr = new Type[100];
        ...
    }
    ...
    Type top(){ return stackPtr[topPtr]; }
}
...
Stack stk = new Stack<int>;
```

→ Java 예

> Generic type(제너릭타입)

- > C++ 템플릿과 비슷하나
- > 단순 타입 검사용도의 힌트, 최근 지원 시작

| Name Encapsulation(명칭 캡슐화)

Name space, Name scope(명칭인식 범위)

- 큰 프로그램에서는 전역변수가 많아 namespace를 나누어서 구조화 시킬 필요가 있음
- 주로 계층적으로 구조화됨

Name Encapsulation(명칭 캡슐화)

- 새로운 name space를 만드는 것

현재 name space 밖의 명칭을 접근하기

- 해당 identifier를 포함하고 있는 모든 namespace를 나타내는 경로명(qualifier) 을 붙여 참조함
예) A.B.C.D.f



| Name Encapsulation

C++

라이브러리가 하나의 인식 범위, 경로명으로 인식

Java

- ➔ 클래스 외에, 클래스의 모임인 패키지 (package) 단위의 명칭 인식범위도 존재
 - > 패키지는 중첩 가능
- ➔ 경로명 전체를 적거나, `import`를 사용해서 단순화

Ada

- ➔ 패키지는 파일 계층구조와 유사한 계층구조를 가지며 이에 따라 명칭 인식 범위도 구조화 됨
- ➔ 외부 명칭은 `with` 로 정의하여 사용을 가능하게 함.

| 추상데이터 타입외의 Encapsulation

➔ 부프로그램으로 쪼개는 것 이외의 구조화 도구 제공

특히, 부분 컴파일이 가능한 컴파일 단위를 제공

C : 부프로그램 명세와 구현을 헤더와 구현 파일로 분리하고, 부분 컴파일을 가능하게 함.

문제 : 링커가 명세와 구현간의 타입 검사를 하지 않음

Ada **FORTRAN95** : 부프로그램 간의 nesting 허용

C# Assembly

- 한 링크 라이브러리나 실행파일에 나타나는 파일들의 묶음
- NET 환경 전체에서 사용되는, class 보다 큰 단위



평가하기

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

START



평가하기 1

1. 다음 중 추상데이터타입이나 name encapsulation, 인자가 있는 추상데이터타입 등에 대한 기능과 지원하는 프로그래밍 언어가 바르게 짝지워 지지 않은 것은?

- ① Java - 패키지 (package)
- ② C# - 어셈블리 (assembly)
- ③ Ada - 제너릭 (generic)
- ④ C++ - 템플릿 (template)

확인



평가하기 2

2. Name encapsulation에 대한 설명과 가장 거리가 먼 것은?

- ① name encapsulation은 변수의 name의 인식 scope을 나누어서 구조화 시키는 것으로 주로 크기가 작은 프로그램에 사용된다.
- ② name encapsulation에서는 name들이 주로 계층적으로 구조화된다.
- ③ name encapsulation을 하면 해당 name을 포함하고 있는 모든 인식범위를 나타내는 경로명(qualifier)을 붙여 참조할 수 있다.
- ④ name의 전체 name scope path를 모두 적어주기 불편하므로 간략하게 표현하는 방법을 제공하기도 한다.

확인



정리하기

- 인자를 가지는 추상 데이터타입은 추상데이터 타입 자체가 고정되어 있는 것이 아니라 인자를 받아서 생성되는 것으로 C++의 template, Ada의 package, Java의 generic 등이 여기 해당된다.
- name encapsulation은 전역변수가 많이 사용되는 크기가 큰 프로그램에서 변수명을 scope 별로 나누어 구조화 시키는 것을 의미한다.



“ 강의를 마치겠습니다. 수고하셨습니다. ”

