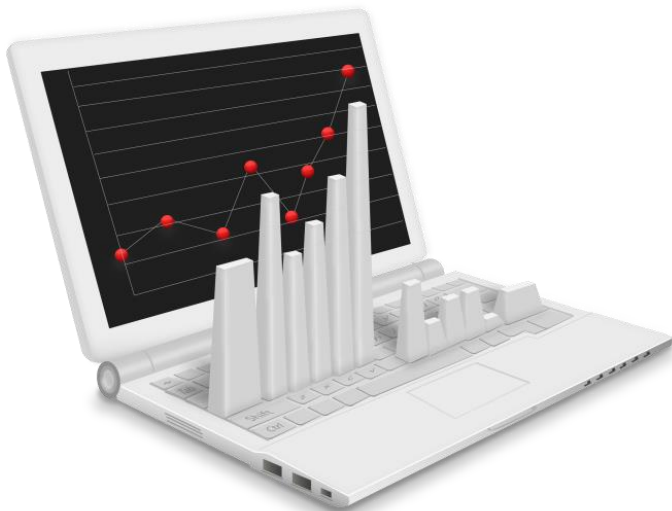


# 프로그래밍 언어론

데이터 추상화

컴퓨터공학과

조은선



## 데이터 추상화

### 학 습 목 표

- 데이터 추상화에 대한 기본적인 개념과 쓰임을 이해한다.

### 학 습 내 용

- 추상화 개념
- 데이터 추상화와 추상데이터 타입



# 목 차

- 들어가기
- 학습하기
  - 추상화 개념
  - Data 추상화와 추상 데이터 타입
- 평가하기
- 정리하기



# 알고가기

! 다음 중 많이 사용하면 할 수록 프로그램이 복잡해지고 가독성이 떨어져서 런타임 오류가 많이 나기 때문에 사용하지 않는 것이 좋다고 알려진 프로그램 요소는?

- ① 변수 선언
- ② `for`
- ③ `goto`
- ④ 함수

확인



# | 추상화 개념

## 추상화 (abstraction)

- 구체적인 세부 사항을 제거한 채 단순화된 모습을 다룰 수 있도록 해주는 기능
- 크고 복잡한 프로그램 작성을 용이하게 해줌

### → 종류

Control flow(제어 흐름) 에  
대한 추상화

Data 추상화



# | 추상화 개념

## control flow에 대한 추상화

분기, 반복 등 제어구조

goto에 대한 일종의 추상화

### 부프로그램

- ➔ 내부는 몰라도 인자와 결과 타입만 알면 호출해서 사용 가능
- ➔ 거의 모든 프로그램 지원
- ➔ 부분 컴파일을 가능하게 함

```
typedef struct {
    float re,im;
} COMPLEX;
COMPLEX add( COMPLEX x, y)
{ . . . }
COMPLEX subtract( COMPLEX x, y) { . . . }
COMPLEX multiply( COMPLEX x, y) { . . . }
COMPLEX divide( COMPLEX x, y)
{ . . . }
COMPLEX a, b, c;
c = add(a,b);
```



# | Control flow 추상화에 대한 한계

## 한계 (앞의 예)

COMPLEX를 사용하는 다른 연산이 더 있는지 불투명

COMPLEX의 변수에 연산을 통하지 않고 직접 접근 가능

한꺼번에 정의

## 해결 방법

아예 다른 연산을 제한하는 것이 낫겠다.

못하게 하는 것이 좋겠다

상세한 구현과 추상화된 인터페이스를 처음부터 따로 떼어 정의하는 것이 낫겠다.

"부프로그램 보다 좀 더 조직적으로 구성되고  
부프로그램처럼 독립적인 컴파일 단위가 되어주는 것이 없을까?"

**Data 추상화**



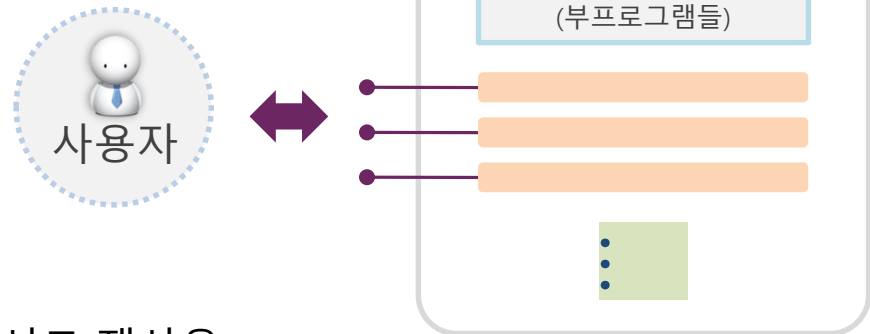
# I 추상 데이터 타입

## 추상 데이터 타입 (Abstract Data Type)

데이터추상화의 기본단위

### 1. 캡슐화 (encapsulation) 지원 단위


- ➔ 1 data type +  $n$  operations를 한 단위 (capsule) 로 보게 해줌
- ➔ 한 개의 데이터 타입과 관련 연산들 (부 프로그램들) 의 묶음
- ➔ 프로그램 수정 용이
  - 구현에 독립적인 인터페이스
- ➔ 재사용 용이 :
  - 표준 인터페이스로 다른 프로그램에서도 재사용





# | 추상 데이터 타입은 information hiding의 지원 단위

## 2. 정보은닉 (information hiding) 지원 단위

➔ 이 타입 데이터의 사용은 반드시 함께 묶여있는 연산들을 통해서만 가능하다. 

➔ 보안성



> 구현의 세부사항을 프로그램 다른 부분에서 접근 금지



# 평가하기

마지막으로 내가 얼마나 이해했는지를 한번 확인해 볼까요?  
총 1문제가 있습니다.

START



# 평가하기 1

## 1. 다음 중 추상 데이터 타입에 관한 설명과 거리가 먼 것은?

- ① 구현에 독립적인 인터페이스 제공으로 프로그램 수정이 용이해진다.
- ② goto 로 표현할 수 있는 것을 부프로그램으로 표현하는 것이 대표적인 예이다.
- ③ 표준 인터페이스를 사용하면 다른 프로그램에서도 재사용하기 쉽다.
- ④ Information hiding을 추가하면 추상 데이터 타입으로 정의된 데이터의 사용은 반드시 함께 묶여있는 연산들을 통해서만 가능하다.

확인



## 정리하기

추상화는 구체적인 사항을 단순하게 바라보게 하여 크고 복잡한 문제를 풀 수 있게 한다.

Control flow 추상화는 goto 대신 부프로그램을 사용하여 단순화 시키는 것이다.

Data 추상화는 이에 더해 연산과 데이터를 함께 묶는 추상데이터타입이 기본단위가 된다.

추상데이터타입은 capsule화 뿐 아니라, 해당 데이터를 해당 연산외의 경로로 접근 할 수 없도록 해서 information hiding을 제공하기도 한다.



“ 강의를 마치겠습니다. 수고하셨습니다. ”

