자료구조와 알고리즘

한국기술교육대학교
온라인평생교육원

학습내용

- 자료구조
- 알고리즘
- 알고리즘 기술 방법
- 추상 자료형

학습목표

- 자료구조의 개념을 설명할 수 있다.
- 알고리즘의 개념을 설명할 수 있다.
- 다양한 알고리즘 기술 방법을 이해하고 파이썬 표기의 특징을 설명할 수 있다.
- 추상 자료형의 개념을 설명할 수 있다.

자료구조

자료구조와 알고리즘 자료구조

1 자료구조의 개념

◉ 일상 생활에서의 다양한 자료들

자료구조와 알고리즘 자료구조

1 자료구조의 개념

◉ 자료의 효율적인 정리 방법

해야 할 일	순서대로 수첩에 기록
빌린 책들	반납일 순으로 책꽂이에 보관
영화관 매표소	먼저 온 사람이 먼저 살 수 있음
영어사전	단어들이 알파벳 순으로 정렬
컴퓨터 저장장치	폴더들이 계층적으로 구성됨
도시들의 연결 상태	지도에 표시

자료구조

자료구조와 알고리즘 자료구조

1 자료구조의 개념

자료구조
(Data Structure)

컴퓨터에서 자료를 정리하고
조직화하는 다양한 구조

자료구조와 알고리즘 자료구조

1 자료구조의 개념

◉ 일상생활의 자료들과 적절한 자료구조의 예

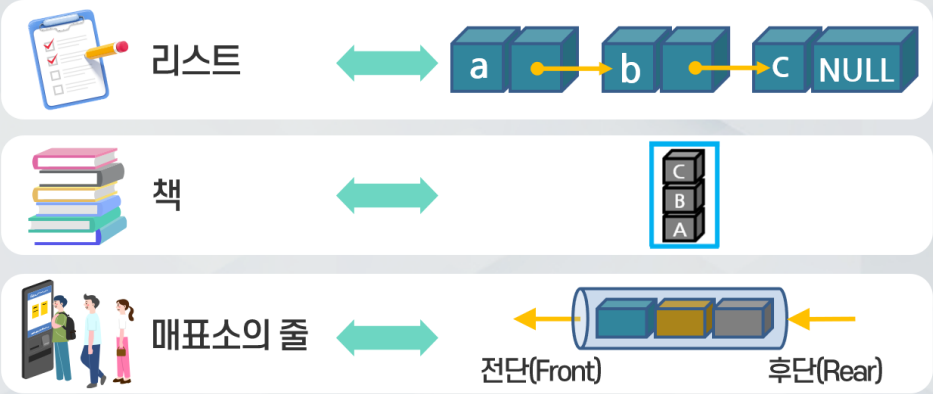
일상생활에서의 예	자료구조
해야 할 일	리스트
물건을 쌓아두는 것	스택
영화관 매표소의 줄	큐
영어사전	사전(Dictionary)
폴더 구조나 조직도	트리
지도	그래프

자료구조

자료구조와 알고리즘 자료구조

1 자료구조의 개념

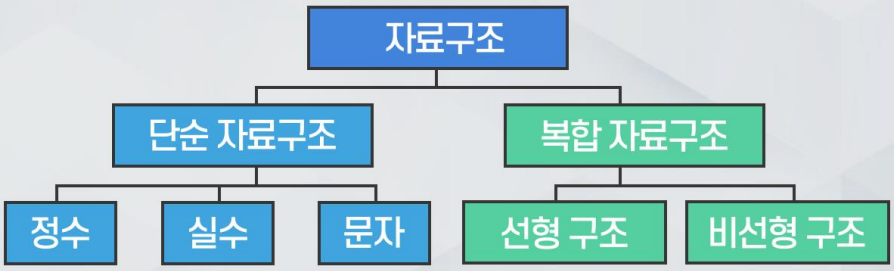
◉ 자료구조(리스트, 스택, 큐)의 실제 구조 예



자료구조와 알고리즘 자료구조

2 자료구조의 종류

◉ 자료구조의 분류



자료구조

자료구조와 알고리즘 자료구조

2 자료구조의 종류

◉ 자료구조의 분류

단순 자료구조	복합 자료구조
숫자, 문자 등의 단일 자료	여러 자료들을 한꺼번에 관리하는 컨테이너

자료구조와 알고리즘 자료구조

2 자료구조의 종류

◉ 복합 자료구조의 종류

선형 자료구조	비선형 자료구조
항목들을 순서적으로 관리	항목들이 보다 복잡한 연결관계를 표현

자료구조

자료구조와 알고리즘 자료구조

2 자료구조의 종류

◉ 선형 자료구조

선형 자료구조

항목들을 순서적으로 나열하여
저장하는 창고(Containner)

- 선형 구조
  - 리스트
  - 스택
  - 큐
  - 덱

자료구조와 알고리즘 자료구조

2 자료구조의 종류

◉ 선형 자료구조

▸▸ 항목 접근 방법에 따른 세분화

리스트	스택, 큐, 덱
가장 자유로운 선형 자료구조	항목의 접근이 맨 앞(전단)이나 맨 뒤(후단)로 제한

자료구조

자료구조와 알고리즘 자료구조

2 자료구조의 종류

◉ 비선형 자료구조

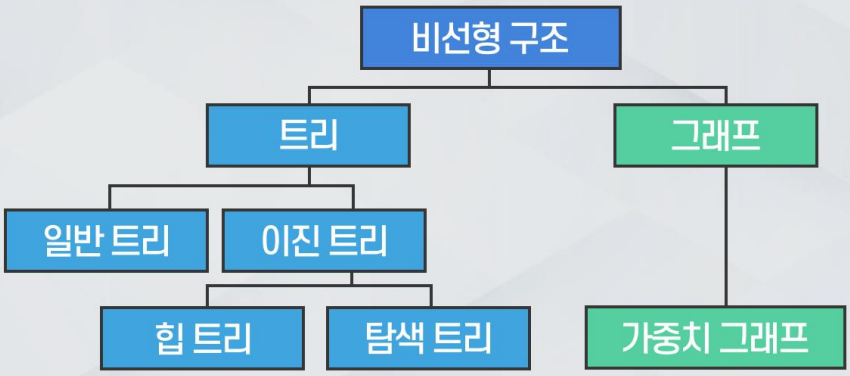
비선형 자료구조

항목들이 보다 복잡한 연결 관계를 표현

자료구조와 알고리즘 자료구조

2 자료구조의 종류

◉ 비선형 자료구조의 종류



자료구조

자료구조와 알고리즘 자료구조

2 자료구조의 종류

◉ 비선형 자료구조

트리

- 회사의 조직도나 컴퓨터의 폴더와 같은 계층 구조 표현에 적합
- 힙 트리는 우선순위 큐를 구현하는 가장 좋은 방법
- 이진 탐색트리나 AVL트리: 탐색을 위한 트리 구조

자료구조와 알고리즘 자료구조

2 자료구조의 종류

◉ 비선형 자료구조

그래프

- 지도나 SNS망과 같이 가장 복잡한 연결 관계를 갖는 자료 표현에 적합
- 다양한 문제를 해결하기 위한 기본 구조로 사용됨

자료구조

자료구조와 알고리즘 자료구조

2 자료구조의 종류

◉ 자료구조의 구현 방법에 따른 분류

배열 구조

연결된 구조

알고리즘

자료구조와 알고리즘 알고리즘

1 알고리즘의 개념

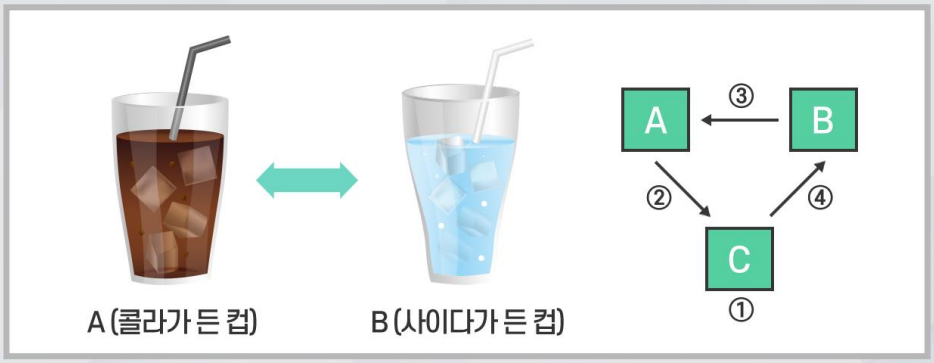
알고리즘(Algorithm)

- 해결해야 할 어떤 문제가 주어졌을 때, 이 문제의 해답을 구하기 위한 절차를 순서대로 명확하게 나타낸 것
- 아부 압둘라 무함마드 이븐 무사 알-콰리즈미 (Al-Khwarizmi) 대수학의 아버지(AD780~AD847)

자료구조와 알고리즘 알고리즘

1 알고리즘의 개념

◉ 알고리즘의 예: 두 컵의 음료를 바꾸는 문제



1 새로운 컵 C를 준비한다.	2 A의 콜라를 모두 C에 붓는다.
3 B의 사이다를 A에 모두 붓는다.	4 C의 내용물을 B에 모두 붓는다.

알고리즘

자료구조와 알고리즘 알고리즘

2 알고리즘과 자료구조의 관계

알고리즘 + 자료구조 = 프로그램

예 최댓값 탐색 프로그램 = 배열 + 순차탐색

자료구조

score[]

80 70 90 … 30

알고리즘

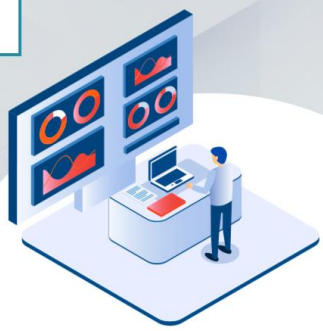
```
tmp←score[0];
for i ← 1 to n do
        if score[i]>tmp
                then tmp←score[i];
```

자료구조와 알고리즘 알고리즘

3 알고리즘의 조건

1 **입력(Well-defined Inputs)**

- 알고리즘이 입력을 받는다면 모호하지 않고 잘 정의된 입력이어야 함
- 알고리즘은 0개 이상의 입력을 가짐



알고리즘

3 알고리즘의 조건

2 출력(Well-defined outputs)

- 출력은 명확하게 정의되어야 함
- 1개 이상의 출력이 반드시 존재해야 함

3 알고리즘의 조건

3 명확성(Clear and Unambiguous)

- 각 명령어의 의미는 모호하지 않고 명확해야 함

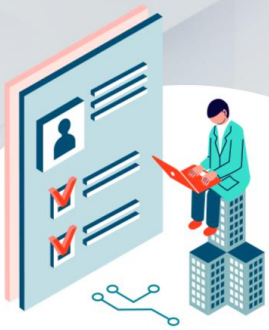
알고리즘

자료구조와 알고리즘 알고리즘

3 알고리즘의 조건

4 유한성(Finiteness)

- 한정된 수의 단계 후에는 반드시 종료되어야 함
- 무한루프나 이와 유사한 상태로 끝나서는 안 됨



자료구조와 알고리즘 알고리즘

3 알고리즘의 조건

5 유효성(Feasible)

- 명령어들은 현재 실행 가능한 연산이어야 함
- 미래에 개발될 기술 등을 포함해서는 안 됨

알고리즘 기술 방법

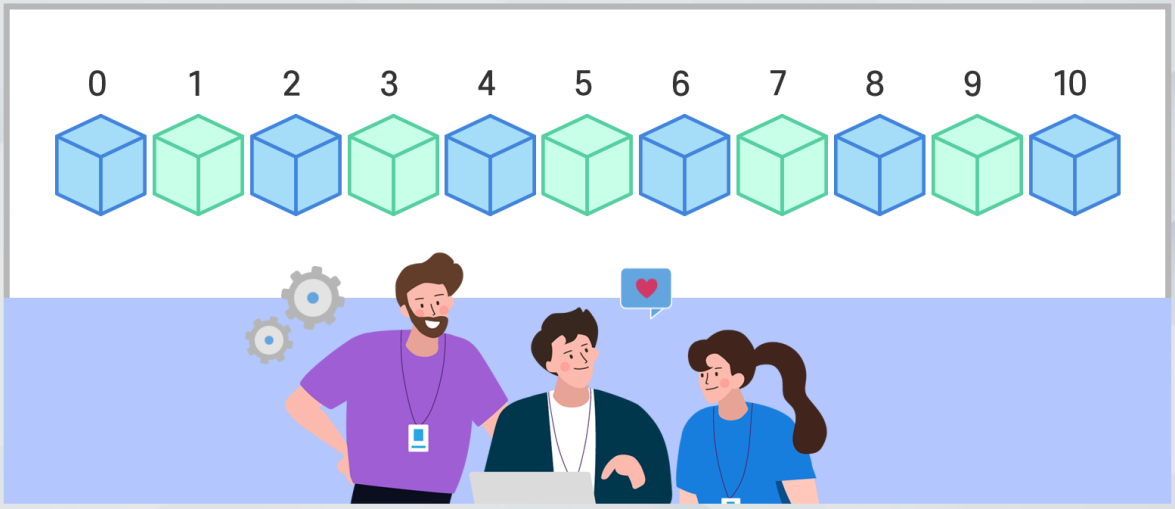
1 알고리즘의 기술 방법

1. 영어나 한국어와 같은 자연어
2. 흐름도(Flow Chart)
3. 유사 코드(Pseudo-code)
4. C나 파이썬과 같은 프로그래밍 언어

자료구조와 알고리즘 알고리즘 기술 방법

1 알고리즘의 기술 방법

예 배열에서 최댓값 찾기 알고리즘



알고리즘 기술 방법

자료구조와 알고리즘 알고리즘 기술 방법

1 알고리즘의 기술 방법

❶ 영어나 한국어와 같은 자연어를 사용하는 방법

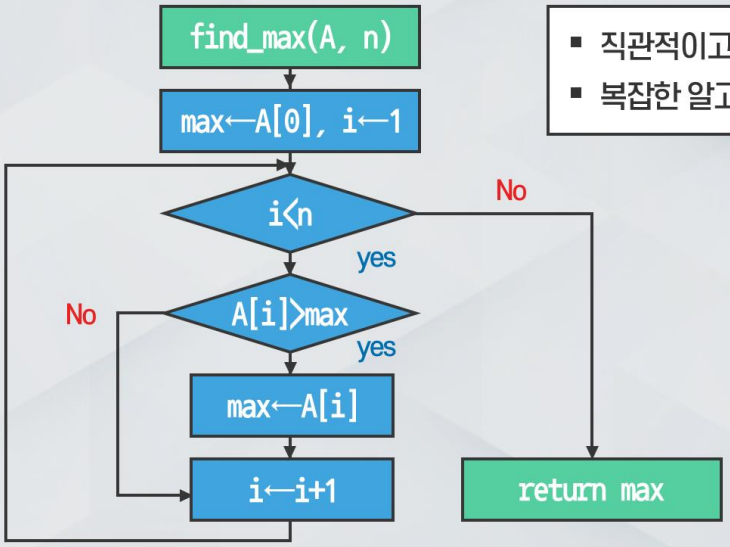
```
find_max(A)

1. 배열 A의 첫 번째 요소를 변수 tmp에 복사한다.
2. 배열 A의 다음 요소들을 차례대로 tmp와 비교하여, 더 크면 그 값을 tmp로 복사한다.
3. 배열 A의 모든 요소를 비교했으면 tmp를 반환한다.
```

- 읽기 쉬움
- 단어들을 정확하게 정의하지 않으면 의미 모호

자료구조와 알고리즘 알고리즘 기술 방법

1 알고리즘의 기술 방법

❷ 흐름도(flowchart)로 표시하는 방법



- 직관적이고 명확하며 이해하기 쉬움
- 복잡한 알고리즘 → 상당히 복잡

알고리즘 기술 방법

1 알고리즘의 기술 방법

❸ 유사 코드(Pseudo-code)로 기술하는 방법

```
find_max(A)
tmp ← A[0]
for i←1 to size(A) do
    if A[i] > tmp then
        tmp ← A[i]
return tmp
```

- 프로그램을 구현할 때의 여러 가지 문제들을 감출 수 있음
- 알고리즘의 핵심적인 내용에만 집중 가능

1 알고리즘의 기술 방법

❹ 특정한 프로그래밍 언어(예: C언어)

```
int find_max(int A[], int n) {          // 배열의 길이도 전달해야 함
    int i, tmp=A[0];                    // 사용할 변수를 미리 선언해야 함
    for( i=1 ; i<n ; i++ ) {            // 블록의 시작을 '{'로 표시 함
        if( A[i] > tmp ) {
            tmp = A[i];                 // = 를 대입 연산자로 사용하고,
        }                               // 비교 연산자로는 ==를 사용함
    }                                   // 중괄호 { }를 맞추어 주어야 함
    return tmp;
}
```

- 알고리즘의 가장 정확한 기술이 가능
- 실제 구현시 많은 구체적인 사항들이 알고리즘의 핵심적인 내용들의 이해를 방해할 수 있음

알고리즘 기술 방법

2 파이썬을 이용한 알고리즘 기술방법

◉ 파이썬 코드와 유사 코드 표현 알고리즘 비교

파이썬 코드

```
def find_max( A ):
    tmp = A[0]
    for item in A :
        if item > tmp :
            tmp = item
    return tmp
```

유사 코드

```
find_max(A)
tmp ← A[0]
for i←1 to size(A) do
    if A[i] > tmp then
        tmp ← A[i]
return tmp
```

- 유사 코드 표현과 큰 차이가 없음
- C언어에서와 같은 문제가 크게 줄어듦

2 파이썬을 이용한 알고리즘 기술방법

◉ 파이썬으로 표현된 알고리즘

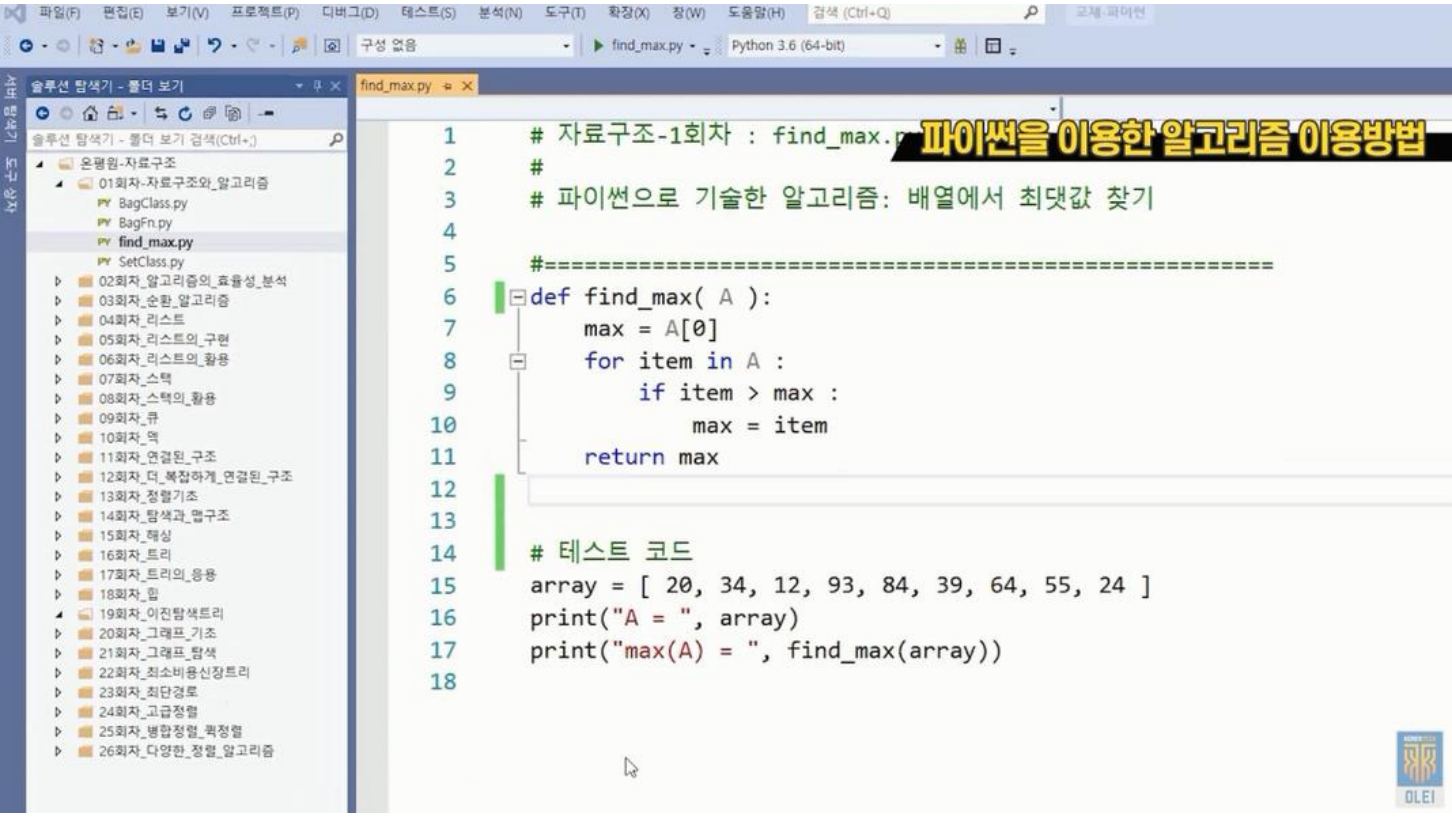
예 알고리즘 테스트_최댓값 찾기

```
array = [ 20, 34, 120, 93, 84, 39, 64, 55, 24 ]
print("A = ", array)
print("max(A) = ", find_max(array))
```

```
C:₩WINDOWS₩system32₩cmd.exe
A =  [20, 34, 120, 93, 84, 39, 64, 55, 24]
max(A) =  120
계속하려면 아무 키나 누르십시오 . . .
```

▶▶ 바로 실행하여 결과를 확인할 수도 있음

알고리즘 기술 방법



실습 단계
Visual Studio 2019
통합 개발 환경
알고리즘 개발 과정을 처리할 수 있는 환경 제공
파이썬은 인터프리터(Interpreter, IDE) 방식 ⇨ 바로 실행 가능
C언어, C++ 는 컴파일러(Complier) 방식
디버깅
왼쪽의 솔루션 탐색기에서 파일 선택
편집기 화면에 파일이 나타남
스크립트 모드 ⇨ 파이썬 코드들을 파일에 묶어 한 번에 실행
무조건 def로 시작
함수 이름, 매개변수가 들어감
블록 기호를 사용하지 않음
파이썬에선 '들여쓰기' 너무 중요! → 모든 블록을 표시
리스트
Just print
Ctrl + F5로 실행

알고리즘 기술 방법

실습 단계
일반적인 알고리즘을 설계할 때 많이 사용하는 방법은 아님
왼쪽 화살표는 'A0를 tmp에 대입한다' 를 의미하는 연산자를 의미
논문이나 책에서 많이 사용되는 알고리즘 방식
코드가 상당히 복잡함
C언어의 경우 반환형 자료구조, 배열의 길이, 블록기호가 들어감.
파이썬도 특정 프로그래밍 언어
프로그래밍 언어에서는 왼쪽 화살표를 사용할 수 없기때문에 대입 연산자 = 를 사용
파이썬 코드와 유사 코드는 상당히 좋은 알고리즘 기술방법
파이썬 코드의 최대 장점

추상 자료형

1 추상 자료형의 개념

◉ 자료형과 추상 자료형

자료형(Data Type)

- 데이터의 집합과 이러한 데이터에 적용할 수 있는 연산의 집합

  예 C언어에서 제공하는 int 자료형

  int 자료형 { 데이터: { …,-2,-1,0,1,2,…}
  연산: +, -, /, *, % }

- 제공하지 않는 자료형

  예 자연수, BigInteger, Bag, Stack, Queue, List, …

자료구조와 알고리즘 추상 자료형

1 추상 자료형의 개념

◉ 자료형과 추상 자료형

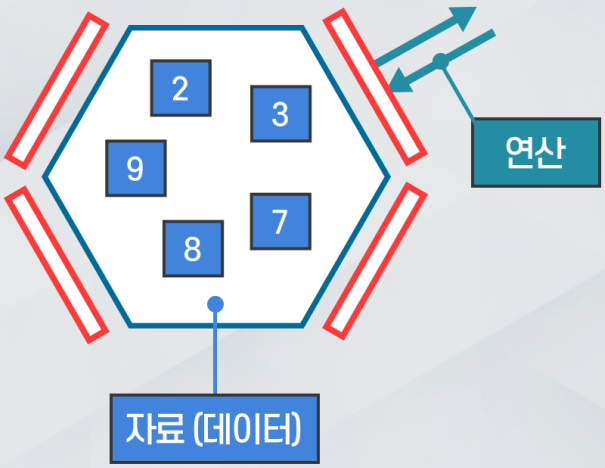
추상 자료형(Abstract Data Type, ADT)

- 프로그래머가 추상적으로 정의한 자료형
  - 자료형을 추상적(수학적)으로 정의한 것
  - 데이터나 연산이 무엇(What)인가를 정의함
  - 데이터나 연산을 어떻게(How) 구현할 것인지는 정의하지 않음
- 시스템의 정말 핵심적인 구조나 동작에만 집중

추상 자료형

자료구조와 알고리즘 추상 자료형

◉ 추상 자료형(ADT)의 형태



데이터

- 추상 자료형에 속하는 자료(데이터) 정의

연산

- 데이터 사이의 연산들이 정의됨
- 추상 자료형과 외부를 연결하는 인터페이스의 역할을 함

자료구조와 알고리즘 추상 자료형

2 추상 자료형의 예

◉ Bag 자료구조

- Bag에 어떤 것들을 넣을까? 어떤 조건이 있을까?
- Bag으로 어떤 일들을 할까?

추상 자료형

2 추상 자료형의 예

◉ Bag 자료구조

▸▸ 가방(Bag)을 위한 추상 자료형

데이터
▪ 중복된 항목을 허용하는 자료들의 저장소 ▪ 항목들은 특별한 순서가 없이 개별적으로 저장되지만 항목간의 비교는 가능해야 함

자료구조와 알고리즘 추상 자료형

2 추상 자료형의 예

◉ Bag 자료구조

▸▸ 가방(Bag)을 위한 추상 자료형

연산
▪ Bag( ): 비어있는 가방을 새로 만듦 ▪ insert(e): 가방에 항목 e를 넣음 ▪ remove(e): 가방에 e가 있는지 검사하여 있으면 이 항목을 꺼냄 ▪ contains(e): e가 들어 있으면 True를, 없으면 False를 반환함 ▪ count( ): 가방에 들어 있는 항목들의 수를 반환함

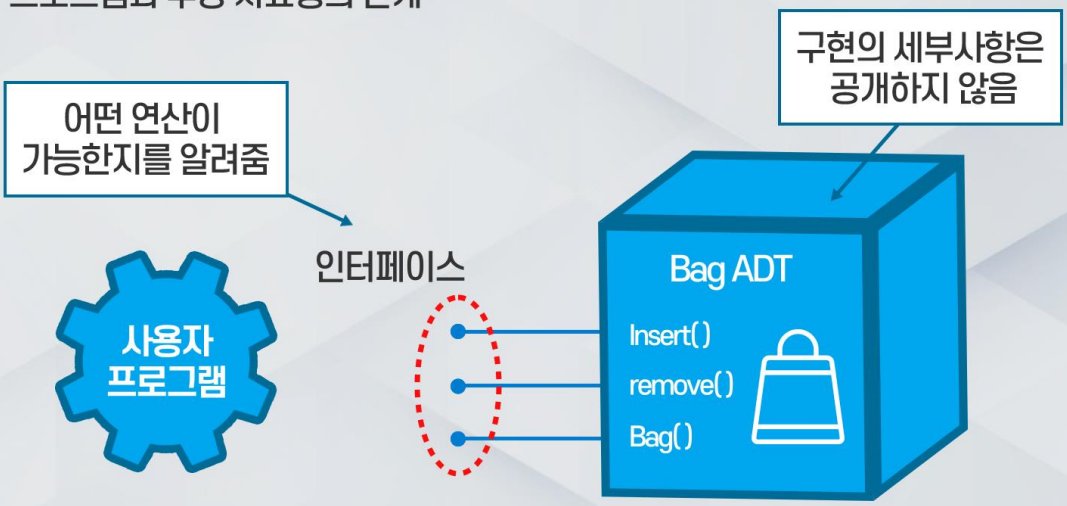
추상 자료형

자료구조와 알고리즘 추상 자료형

2 추상 자료형의 예

◉ Bag 자료구조

▸▸ 사용자 프로그램과 추상 자료형의 관계



자료구조와 알고리즘 추상 자료형

2 추상 자료형의 예

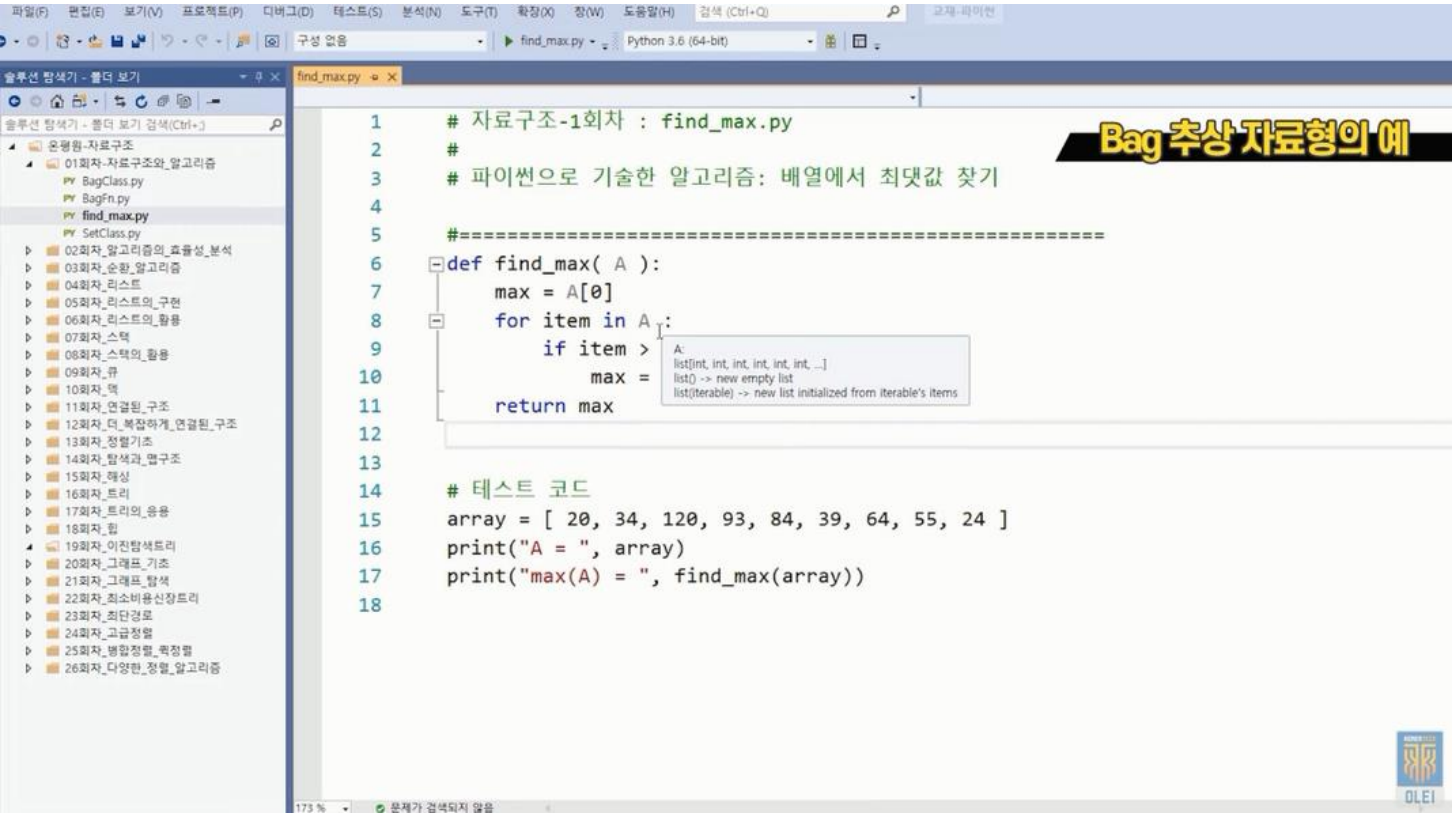
◉ Bag 자료구조의 구현과 활용

▸▸ 함수를 이용한 Bag ADT의 구현

▸▸ 클래스를 이용한 Bag ADT의 구현

▸▸ Bag ADT의 활용

추상 자료형



실습 단계
오른쪽 버튼 클릭
BagFn.py 파일 실행
구현 방법 2가지
함수를 이용하는 방법 ⇨ 연산을 함수로 구현
매개변수를 사용하는 방법 ⇨ 데이터
Bag의 사용 방법
공백 리스트를 만들기
테스트 시작 → Ctrl + F5로 실행
결과 출력
클래스(class)로 구현
자료구조를 구현하는 가장 좋은 방법은 클래스(class)
클래스(class)로 만든 추상 자료형 사용 방법
객체 만들기
객체에서는 점(.)을 이용하여 멤버 함수를 호출
추상 자료형을 구현하는 가장 좋은 방법은? ⇨ 클래스(class)