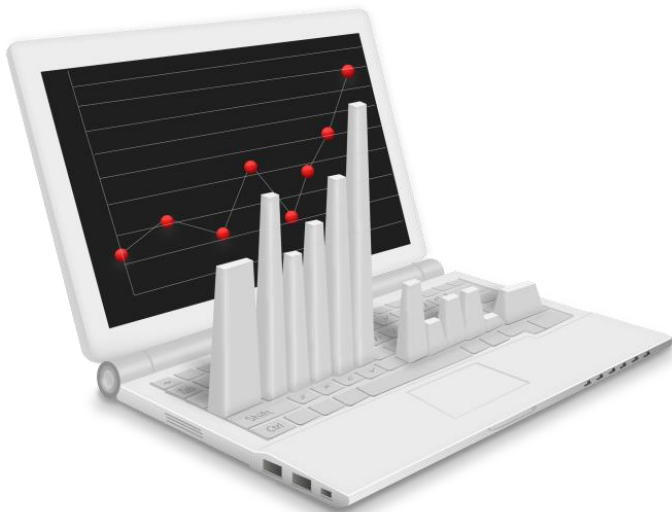


프로그래밍 언어론

다양한 추상데이터타입 예

컴퓨터공학과
조은선



다양한 추상데이터타입 예

학 습 목 표

- 다양한 프로그래밍 언어에서 추상 데이터 타입이 제공되는 방식에 대해 이해한다.

학 습 내 용

- 다양한 프로그래밍 언어에서의 추상데이터 타입
- Ada 의 추상데이터 타입
- C++ 의 추상데이터 타입
- Java 의 추상데이터 타입
- C# 의 추상데이터 타입



목 차

- 들어가기
- 학습하기
 - 다양한 프로그래밍 언어에서의 추상 데이터 타입
 - Ada 의 추상 데이터 타입
 - C++ 의 추상 데이터 타입
 - Java 의 추상 데이터 타입
 - C# 의 추상 데이터 타입
- 평가하기
- 정리하기



알고가기



C 프로그램 내에 다음과 같은 문장이 있다면 이에 대한 설명으로 옳지 않은 것은?

```
void f(int i, int j);
```

- ① f 라는 이름의 함수의 선언이다.
- ② 인자는 두 개인데 모두 `int` 타입이다.
- ③ 함수의 실제 구현의 시작을 나타낸다.
- ④ 함수를 호출할 때 사용하는 인터페이스만을 가지고 있다.

확인



| 다양한 프로그래밍 언어에서의 추상 데이터 타입

➔ 추상 데이터 타입의 예

➔ 내장된 기본 타입들은 넓게 보면 추상 데이터 타입의 일종

예 `int` = 숫자값 + { `+`, `-`, `*`, `/` }

➔ 대부분의 1980년대 이후 언어들은 사용자 정의 추상 데이터 타입 지원

➔ Simula 67

- 기본적인 캡슐 형태 최초로 제공
- 정보 은닉 (information hiding) 제공되지 않음



| Ada의 추상 데이터 타입

Ada : “패키지 (package)”

인터페이스와 구현의 분리

패키지 (package) = 명세 패키지 (specification p.) + body 패키지 (body p.)

Information Hiding 지원

➔ 은닉될 타입은 명세 패키지에서 아래를 수행

1. 키워드 **private** 을 써서 전용 타입으로 선언하고

> 사용자를 위해

2. 그 아래서 **private** 구문 내에서 정의

> 컴파일러를 위해 (사용자는 열람 불가)

➤ **private**으로 선언하면 기본연산 외의 여타 연산은 불허

기본 연산 : assignment, 동등비교, 비동등비교

> **private** 대신 **limited private**이라고 하면 기본 연산까지 불허

package가 정의하는 것만 사용

```
package ... is
  type NODE_TYPE is private;
  ...
  private
    type NODE_TYPE is
      record
        ...
      end record;
  ...
end record;
```

A

P

| C++의 추상데이터 타입

클래스

C의 `struct`에 기반한 추상 데이터타입

한 클래스 타입에 속하는 개체 (instance) 들

- ➔ 연산 (멤버함수) 을 공유하고, 데이터 (멤버)는 따로 가짐
- ➔ stack에 정적, 동적으로 heap에 동적으로 생성 가능
- ➔ **생성자 연산**
 - 클래스 이름과 동일한 멤버함수, 개체생성시 초기화 루틴이 들어있음
 - `new` 등으로 개체 생성시, 메모리 할당 직후 수행됨- 별도로 호출 불가
- ➔ **소멸자 연산**
 - 클래스 이름에 앞에 `~`를 붙인 멤버함수로, 개체 소멸시 자동으로 수행될 내용이 들어있음
 - 별도로 호출불가



| C++의 클래스와 Information Hiding

클래스를 구성하는 데이터나 연산의 information hiding

➤ 클래스를 구성하는 데이터나 연산의 정보 은닉

> **public** : 모든 곳에서 접근 가능

> **private** : 같은 클래스에 속하는 개체끼리만 접근 가능

> **friend** 함수, **friend** 클래스 : 다른 클래스에 속하는 개체끼리 접근하는 것도 허용, 밀접한 클래스를 나타냄.

➡ Ada는 데이터 타입이기 보다는 encapsulation에 중점을 둠

➡ C++은 하나의 타입을 캡슐 단위 (클래스) 로 일치 시킴



| Java의 추상데이터 타입

클래스

- C++의 클래스와 비슷하나,
- 모든 사용자 정의 타입은 클래스이고
- 소속된 모든 개체는 heap에 저장되어 참조에 의해 접근
- 클래스가 모여서 패키지(package)가 됨

클래스를 구성하는 데이터나 연산의 정보 은닉

public : 모든 곳에서 접근 가능

private : 같은 클래스에 속하는 개체끼리만 접근 가능

package : 동일 패키지에 속하는 클래스에 속하는 개체끼리만 접근 가능



| C#의 추상데이터 타입

C# : 클래스 (class)

클래스

- ➔ Java의 클래스와 비슷하나,
 - 소속된 모든 개체는 힙에 저장
- ➔ 클래스가 모여 어셈블리 (Assembly)

클래스를 구성하는 데이터나 연산의 정보 은닉

- ➔ `private`, `public` 외에 `internal`, `protected internal` 등이 추가
 - `internal` : 하나의 어셈블리에 의해 속한 모든 클래스에 대해 가시적
- ➔ `getter/setter` : 복잡한 계산보다는 데이터 값을 그대로를 접근, 갱신하는 작업이 더 의미가 있는 클래스에서 사용
 - `property` 를 사용해서 구현

(예)



| C#의 추상데이터 타입

C# : 클래스 (class)

클래스

- ➔ Java의 클래스
- 소속된 모든
- ➔ 클래스가 모

클래스를 구성하

- ➔ **private, public**
- **internal**
- ➔ **getter/setter**
- **property**

```
public class Weather {
    public int DegreeDays {
        /** DegreeDays is a property
        get {      /** 코드 추가 가능
            return degreeDays;
        }
        set {      /** 코드 추가 가능
            degreeDays = value;
        }
    }
    private int degreeDays;
    ...
}

...
Weather w = new Weather();
int degreeDaysToday, oldDegreeDays;
...
w.DegreeDays = degreeDaysToday;
...
oldDegreeDays = w.DegreeDays;
```

이 추가
가시적

업근, 갱신하는 작업이

(예)



평가하기

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

START



평가하기 1

1. 다음 중 각 프로그래밍 언어와 그 언어에서 지원하는 추상 데이터 타입에 관한 정보 은닉 제어 키워드가 올바르게 연결된 것은?

- ① Ada - limited (private)
- ② C++ - internal
- ③ Java - friend
- ④ C# - package

확인



평가하기 2

2. 추상 데이터 타입에 대한 설명 중 거리가 먼 것은?

- ① Ada의 패키지는 명세 패키지와 Body 패키지로 이루어져 있다.
- ② C#의 클래스가 모여서 된 것을 어셈블리라고 한다.
- ③ C++ 의 클래스의 객체는 stack에 생성될 수도 있고, heap에 생성될 수도 있다.
- ④ information hiding과 encapsulation이 최초로 지원된 프로그래밍 언어는 Simula67이다.

확인



정리하기

• 추상 데이터 타입의 예

- ➔ Ada는 package라는 추상데이터 타입을 제공하나 데이터 타입개념 보다는 encapsulation 개념이 강하다. package는 명세와 body의 정의를 분리하고 있으며, information hiding 을 위한 `private` 등을 제공하고 있다.
- ➔ C++, Java, C# 등의 class는 encapsulation과 information hiding 을 지원하기 위해 각기 특수한 키워드 집합을 제공하고 있다.



“ 강의를 마치겠습니다. 수고하셨습니다. ”

