



객체 지향 프로그래밍

1. 추상 데이터 타입
2. 상속
3. 동적 바인딩

01_객체 지향 프로그래밍의 개요

❖ 객체지향 프로그래밍의 개념

- 1960년대 크리스텐 니가드(Kristen Nygaard)와 올레 조한 달(Ole-Johan Dahl)에 의해 개발된 SIMULA에서부터 시작
- 1967년에 클래스, 객체, 그리고 상속 개념을 포함한 SIMULA 67이 개발되면서 최초의 객체지향 개념이 도입
- 진정한 최초의 객체지향 언어는 1980년에 앨런 케이(Alan Kay)에 의해 개발된 Smalltalk 80 : 동적 바인딩을 지원
- C++, Java, C#
- 객체지향 언어는 추상 데이터 타입, 상속, 동적 바인딩이라는 개념을 지원

02_추상 데이터 타입

❖ 추상 데이터 타입

- 임의의 타입의 데이터 표현과 그와 관련된 연산을 제공하는 부프로그램들을 함께 묶어 캡슐화한 것
- 캡슐화한 하나의 데이터 타입과 관련된 모든 정의를 한 곳에 모으고 그곳에서 정의된 연산을 통해서만 이 데이터 타입을 사용하도록 제한하는 것을 의미
- 객체 : 추상 데이터 타입의 실체

❖ 추상 데이터 타입의 구조

| 추상 데이터 타입 이름 |
|-----------------------------|
| 데이터1 데이터2 ... |
| 부프로그램1() 부프로그램2() ... |

3

02_추상 데이터 타입

❖ 추상 데이터 타입의 정의

- 데이터 타입과 그 타입과 관련된 연산이 동시에 정의되어야
 - 정의들은 한 곳에 모여 있어야 하며, 연산과 타입은 직접적으로 연계되어야 함
- 데이터 타입의 세부 사항은 그 타입에 마련된 연산만이 접근할 수 있음
- 그 타입을 사용하는 프로그램이 직접적으로 접근할 수 없음
 - 추상 데이터 타입에서 제공하는 연산을 호출해야만 객체의 상태를 변경시킬 수 있다.
- C로 작성한 사원의 급여를 관리하는 프로그램

• emp 구조 정의

```
struct employee {
    char name[20];
    int basicsalary;
    int bonus;
    int annualsalary;
};
```

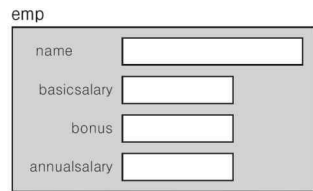
• emp 선언 후 이름, 기본급, 보너스, 연봉에 값 대입

```
struct employee {
    struct employee emp;
    strcpy (emp.name, "abc");
    emp.basicsalary = 150;
    emp.bonus = 400;
    emp.annualsalary = emp.basicsalary*12 +
        emp.basicsalary*emp.bonus*0.01;
```

4

02_추상 데이터 타입

▪ 레코드 타입 변수 emp의 구조



- emp의 데이터인 name, basicsalary, bonus, annualsalary가 감춰지지 않음. 기본급 또는 보너스가 수정될 때 연봉까지 수정해야 하는 부담 존재

[그림 9-2] 레코드 타입 변수 emp의 구조

- emp에 대한 이름과 연봉을 출력하는 문장
 > `printf("%s %d\n", emp.name, emp.annualsalary);`
- 기본급이 200으로 변경
 > `emp.basicsalary = 200;`
 `emp.annualsalary = emp.basicsalary*12 + emp.basicsalary*emp.bonus*0.01;`
- 보너스가 500%로 변경
 > `emp.bonus = 500;`
 `emp.annualsalary = emp.basicsalary*12 + emp.basicsalary*emp.bonus*0.01;`

5

02_추상 데이터 타입

▪ C++로 작성한 사원의 급여를 관리하는 프로그램

- 이름과 연봉을 출력 : emp 객체의 데이터인 name, annualsalary를 감춤
 > `cout << emp.getName() << " " << emp.getAnnualSalary();`
- 기본급이 200으로 변경
 > `emp.changeBasicSalary(200);`

추상 데이터 타입

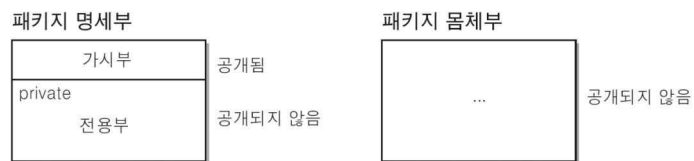
- 임의의 타입의 데이터 표현과 그와 관련된 부프로그램들을 함께 묶어 캡슐화
- 이 타입에서 제공하는 부프로그램만을 통해 동작이 이루어지므로 관리하기 편하고 안정성이 뛰어나

6

02_추상 데이터 타입

❖ Ada의 추상 데이터 타입

- 패키지를 이용하여 추상 데이터 타입을 정의
- 명세부 : 패키지를 사용하는 프로그램에게 인터페이스를 제공
 - 가시부(visible part) : 가시부에 선언된 내용은 외부에 공개됨
 - 전용부(private part) : 전용부에 선언된 내용은 외부에 공개되지 않음
- 몸체부 : 명세에서 명명된 연산에 대한 구현을 제공



[그림 9-3] 패키지 명세부와 패키지 몸체부

7

02_추상 데이터 타입

- 패키지의 명세부 형태

```
package 패키지이름 is
:
end 패키지이름;
```

- 패키지 몸체부의 형태

```
package body 패키지이름 is
:
end 패키지이름;
```

8

02_추상 데이터 타입

▪ employeePack 패키지 명세부

```
package employeePack is
  type employee is private;
  procedure initEmp(emp: out employee; name: string; basicsalary,
    bonus, annualsearly: integer);
  function getName(emp: employee) return string;
  function getAnnualSalary(emp: employee) return integer;
  procedure changeBasicSalary(emp: in out employee; basicsalary: integer);
  procedure changeBonus(emp: in out employee; bonus: integer);
private
  type employee is
    record
      name: string(1..3);
      basicsalary: integer;
      bonus: integer;
      annualsearly: integer;
    end record;
end employeePack;
```

9

02_추상 데이터 타입

▪ employeePack 패키지 몸체부

```
with TEXT_IO;
use TEXT_IO;
package body employeePack is
  package INT_IO is new TEXT_IO.INTEGER_IO (integer);
  use INT_IO;
  procedure initEmp(emp: out employee; name: string; basicsalary, bonus,
    annualsearly: integer) is
  begin
    emp := (name, basicsalary, bonus, annualsearly);
  end initEmp;
  function getName(emp: employee) return string is
  begin
    return emp.name;
  end getName;
  function getAnnualSalary(emp: employee) return integer is
  begin
    return emp.annualsearly;
  end getAnnualSalary;
  procedure changeBasicSalary(emp: in out employee; basicsalary: integer) is
  begin
```

10

02_추상 데이터 타입

```

emp.basicsalary := basicsalary;
emp.annualsalary := emp.basicsalary*12 +
emp.basicsalary*emp.bonus/100;
end changeBasicSalary;
procedure changeBonus(emp: in out employee; bonus: integer) is
begin
    emp.bonus := bonus;
    emp.annualsalary := emp.basicsalary*12 +
                        emp.basicsalary*emp.bonus/100;
end changeBonus;
end employeePack;

```

- 정의된 employee 추상 데이터 타입의 실체인 객체 생성 방법

```
emp : employee;
```

11

02_추상 데이터 타입

- 객체 emp의 구조

| |
|---|
| emp: employee |
| name basicsalary bonus annualsalary |
| initEmp() getName() getAnnualSalary() changeBasicSalary() changeBonus() |

- 객체 emp의 각 데이터에 값을 저장

```
initEmp(emp, "abc", 150, 400, 2400);
```
- emp 객체의 name과 annualsalary를 알아내기

```
put(getName(emp));
```

```
put(getAnnualSalary(emp));
```
- emp의 기본급을 200으로 변경

```
changeBasicSalary(emp, 200);
```

12

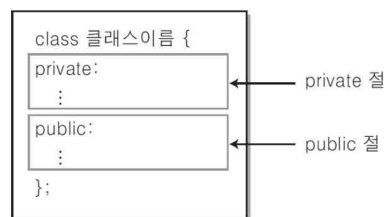
02_추상 데이터 타입

❖ C++의 추상 데이터 타입

- 클래스(class)를 이용해서 추상 데이터 타입을 지원
- 멤버 데이터 : 클래스에서 정의된 데이터
- 멤버 함수 : 클래스에서 정의된 부프로그램

```
class 클래스이름 {
    :
};
```

- 클래스
 - private 절에 속한 멤버들은 외부로 공개되지 않음
 - public 절에 속한 멤버들은 외부로 공개



13

02_추상 데이터 타입

▪ employee 클래스를 정의한 예 1

```
class employee {
private:
    char *name;
    int basicsalary;
    int bonus;
    int annualsalary;
public:
    employee(char *na, int bs, int bon, int as) {
        name = new char[strlen(na)+1];
        strcpy(name, na);
        basicsalary = bs;
        bonus = bon;
        annualsalary = as;
    }
    ~employee() {
        delete []name;
    }
    char* getName() {
        return name;
    }
};
```

14

02_추상 데이터 타입

```
int getAnnualSalary() {
    return annualsealary;
}
void changeBasicSalary(int bs) {
    basicsalary = bs;
    annualsealary = basicsalary*12 + basicsalary*bonus*0.01;
}
void changeBonus(int bon) {
    bonus = bon;
    annualsealary = basicsalary*12 + basicsalary*bonus*0.01;
}
};
```

15

02_추상 데이터 타입

▪ employee 클래스를 정의한 예 2

```
class employee {
private:
    char *name;
    int basicsalary;
    int bonus;
    int annualsealary;
public:
    employee(char *na, int bs, int bon, int as);
    ~employee();
    char* getName();
    int getAnnualSalary();
    void changeBasicSalary(int bs);
    void changeBonus(int bon);
};
employee::employee(char *na, int bs, int bon, int as) {
    name = new char[strlen(na)+1];
    strcpy(name, na);
    basicsalary = bs;
    bonus = bon;
    annualsealary = as;
}
```

16

02_추상 데이터 타입

```

employee::~~employee() {
    delete []name;
}

char* employee::getName() {
    return name;
}

int employee::getAnnualSalary() {
    return annualsearly;
}

void employee::changeBasicSalary(int bs) {
    basicsalary = bs;
    annualsearly = basicsalary*12 + basicsalary*bonus*0.01;
}

void employee::changeBonus(int bon) {
    bonus = bon;
    annualsearly = basicsalary*12 + basicsalary*bonus*0.01;
}

```

- employee 클래스의 실체인 객체를 생성

```
employee emp( "abc" , 150, 400, 2400);
```

17

02_추상 데이터 타입

- 객체 emp의 구조

| | |
|---|---------|
| emp: employee | |
| name = "abc" basicsalary = 150 bonus = 400 annualsearly = 2400 | 공개되지 않음 |
| employee() ~employee() getName() getAnnualSalary() changeBasicSalary() changeBonus() | 공개됨 |

- emp 객체의 name과 annualsearly를 알아내는 문장

```
cout << emp.getName() << " " << emp.getAnnualSalary();
```
- emp 객체의 기본급을 200으로 변경

```
emp.changeBasicSalary(200);
```

18

02_추상 데이터 타입

❖ Java의 추상 데이터 타입

▪ 클래스

- 멤버 변수 : 클래스에서 정의된 데이터
- 메소드 : 클래스에서 정의된 부프로그램

```
class 클래스 이름 {
    :
}
```

▪ 멤버 변수와 메소드에 대한 접근 변경자

```
class 클래스 이름 {
    private 멤버변수 또는 메소드;
    :
    public 멤버변수 또는 메소드;
    :
};
```

▪ 소멸자가 없음

19

02_추상 데이터 타입

▪ Java로 표현한 employee 클래스

```
class employee {
    private String name;
    private int basicsalary;
    private int bonus;
    private int annualsalary;

    public employee(String name, int basicsalary, int bonus, int annualsalary) {
        this.name = name;
        this.basicsalary = basicsalary;
        this.bonus = bonus;
        this.annualsalary = annualsalary;
    }

    public String getName() {
        return name;
    }

    public int getAnnualSalary() {
        return annualsalary;
    }
}
```

20

02_추상 데이터 타입

```
public void changeBasicSalary(int basicsalary) {
    this.basicsalary = basicsalary;
    annualsalary = (int)(basicsalary*12 + basicsalary*bonus*0.01);
}

public void changeBonus(int bonus) {
    this.bonus = bonus;
    annualsalary = (int)(basicsalary*12 + basicsalary*bonus*0.01);
}
}
```

- employee 클래스의 실체인 객체를 생성

```
employee emp = new employee( "abc" , 150, 400, 2400);
```

21

02_추상 데이터 타입

- 객체 emp의 구조

| | |
|--|---------|
| emp: employee | |
| name = "abc" basicsalary = 150 bonus = 400 annualsalary = 2400 | 공개되지 않음 |
| employee() getName() getAnnualSalary() changeBasicSalary() changeBonus() | 공개됨 |

- emp 객체의 name과 annualsalary를 알아내는 문장
`System.out.println(emp.getName() + " " + emp.getAnnualSalary());`
- emp 객체의 기본급을 200으로 변경
`emp.changeBasicSalary(200);`

22