

03

학습 목표

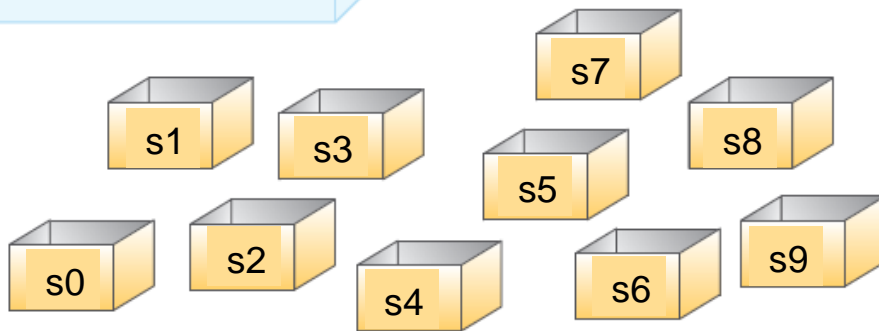
1. 실세계의 객체와 C++ 객체에 대해 이해한다.
2. C++ 클래스를 작성할 수 있다.
3. 객체를 생성하고 활용할 수 있다.
4. 생성자와 소멸자를 알고 작성할 수 있다.
5. private, protected, public 접근 지정자를 이해한다.
6. 인라인 함수의 목적을 이해하고 활용할 수 있다.

클래스의 필요성

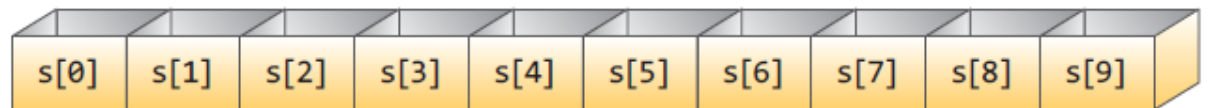
3

□ 일반 변수 -> 배열

```
// 일반 변수 사용  
int s0;  
int s1;  
...  
int s9;
```



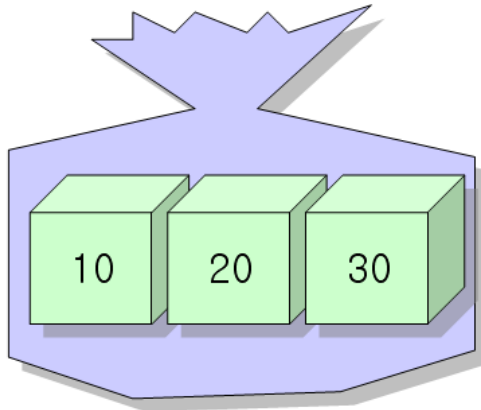
```
// 배열 사용  
int s[10];
```



클래스의 필요성

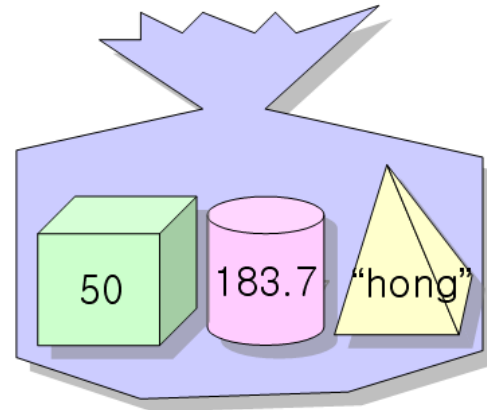
4

□ 배열 -> 구조체



배열

같은 종류의 데이터들의 집합



구조체

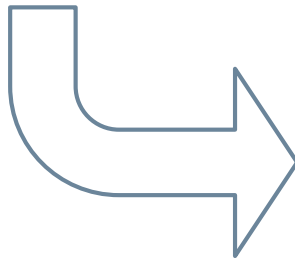
다른 종류의 데이터들의 집합

클래스의 필요성

5

- 변수(데이터 저장) + 함수(데이터 처리) -> 클래스(class)

```
struct student list[10];  
....  
struct student get_max(struct student a[]);  
struct student get_min(struct student a[]);  
void sort(struct student a[]);  
....
```



```
class TV {  
    bool on;  
    int channel;  
    int volume;  
public:  
    void powerOn() { ... }  
    void powerOff() { ... }  
    void increaseChannel() { ... }  
    void decreaseChannel() { ... }  
    void increaseVolume() { ... }  
    void decreaseVolume() { ... }  
};
```

TV 클래스

C++ 클래스와 C++ 객체

6

- 클래스(class)
 - ▣ 사용자 정의 자료형
 - ▣ 관련 있는 변수(데이터의 저장공간)와 함수(데이터를 처리하는 코드)를 모아 놓은 것 -> 구조체의 발전된 형태
 - ▣ 객체를 만들어내기 위해 정의된 설계도(틀)
- 객체(object)
 - ▣ 클래스형 변수
 - ▣ 클래스라는 설계도(틀)로 만들어낸 결과물
 - ▣ 멤버 변수와 멤버 함수로 구성
 - ▣ 메모리에 생성(할당)된 변수 -> 실체(인스턴스, instance)라고 부름
 - ▣ 클래스는 자료형을 의미하고 객체는 변수를 의미함

클래스와 객체 관계

7

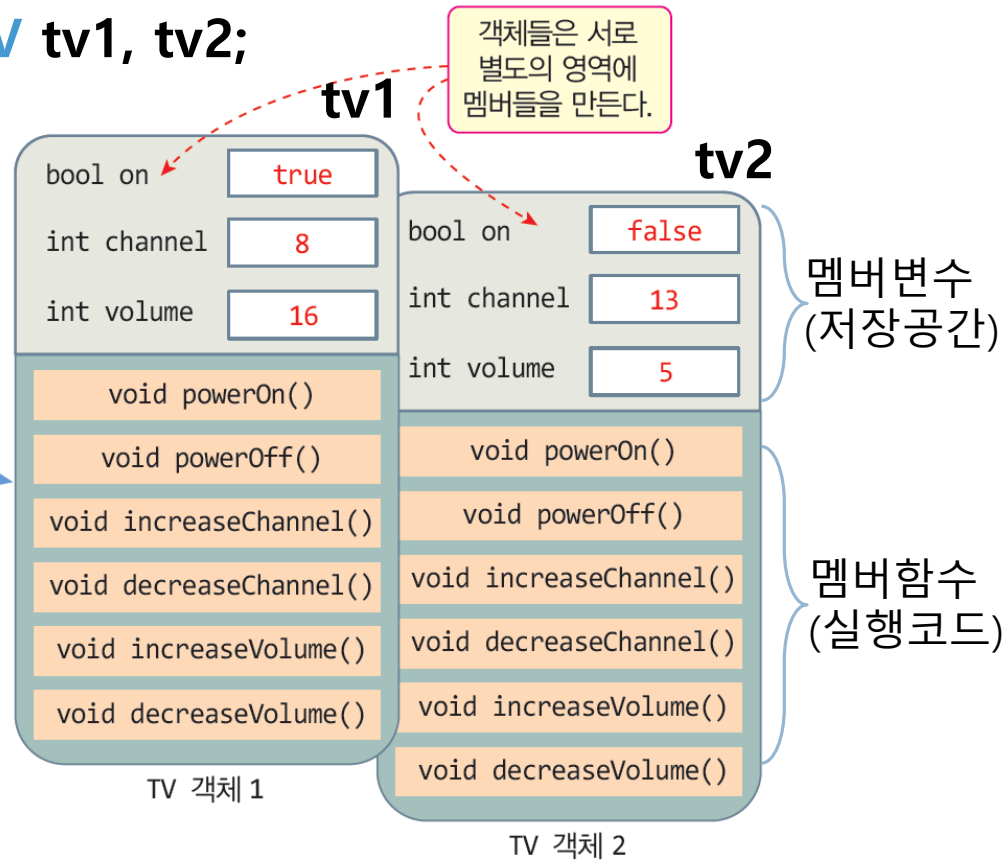
사용자 정의 자료형

```
class TV {  
    bool on;  
    int channel;  
    int volume;  
public:  
    void powerOn() { ... }  
    void powerOff() { ... }  
    void increaseChannel() { ... }  
    void decreaseChannel() { ... }  
    void increaseVolume() { ... }  
    void decreaseVolume() { ... }  
};
```

TV 클래스

TV tv1, tv2;

객체 생성



(b) C++로 표현한 TV 클래스와 TV 객체들

C++ 클래스 정의

8

- 클래스 정의
 - ▣ 멤버 변수와 멤버 함수로 구성
 - ▣ 클래스 선언부와 클래스 구현부로 나누어서 작성
 - ▣ 메인 함수 앞부분에 작성
- 클래스 선언부(class declaration)
 - ▣ class 키워드를 이용하여 클래스 선언
 - ▣ 멤버 변수와 멤버 함수 선언
 - ▣ 접근지정자 : 외부에서 멤버에 대한 접근 권한 지정
 - private, public, protected 중의 하나, 디폴트는 private
 - public : 다른 모든 객체에서 멤버의 접근이 가능함을 표시
- 클래스 구현부(class implementation)
 - ▣ 클래스 선언부에서 선언된 멤버 함수를 정의

C++ 클래스 정의

9

클래스의 선언은
class 키워드 이용

클래스 이름

멤버에 대한
접근 지정자

세미콜론
으로 끝남

```
class Circle {  
public:  
    int radius;           // 멤버 변수  
    double getArea();     // 멤버 함수  
};
```

클래스
선언부

함수의 리턴
타입

클래스
이름

범위지정연
산자

멤버 함수명과
매개변수

```
double Circle :: getArea() {  
    return 3.14*radius*radius;  
}
```

클래스
구현부

→ 멤버함수에서는 같은 클래스의 모든 멤버
변수를 접근가능

객체 생성 및 멤버 참조

10

□ 멤버변수

- ▣ 객체와 관련 있는 데이터를 저장하는 공간
- ▣ 모든 자료형의 변수 선언 가능
- ▣ 구조체의 멤버변수와 같음

□ 멤버함수

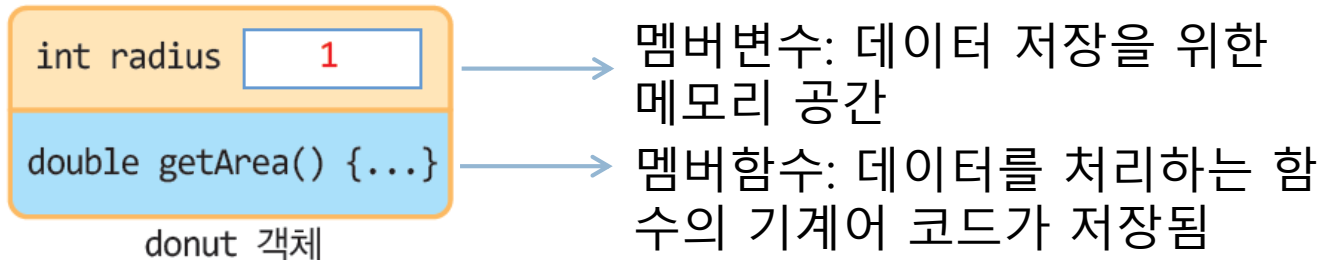
- ▣ 멤버변수를 처리하는 함수들
- ▣ 같은 클래스의 모든 멤버변수를 자유롭게 접근가능(사용가능) -> 인자로 받을 필요 없음
- ▣ 멤버변수와 묶어서 관리함으로써 편리

객체 생성 및 멤버 참조

11

- 객체 생성 -> 클래스명 객체명;
- 멤버 참조 -> 객체명.멤버변수, 객체명.멤버함수

```
Circle donut;           // 객체 생성
donut.radius = 1;       // 멤버변수 참조
double area = donut.getArea(); // 멤버함수 참조
```



예제 3-1 Circle 클래스 정의와 객체 생성

12

```
#include <iostream>
using namespace std;
```

```
class Circle {
public:
    int radius;
    double getArea();
};
```

```
double Circle::getArea() {
    return 3.14 * radius * radius;
}
```

Circle 선언부

Circle 구현부

→ 멤버함수에서는 같은 클래스의 모든 멤버 변수를 접근가능

예제 3-1 Circle 클래스 정의와 객체 생성

13

```
int main() {  
    Circle donut;           // donut 객체 생성  
    donut.radius = 1;       // donut 멤버변수 접근  
    double area = donut.getArea(); // donut 멤버함수 호출  
    cout << "donut 면적은 " << area << endl;  
  
    Circle pizza;  
    pizza.radius = 30;  
    area = pizza.getArea();  
    cout << "pizza 면적은 " << area << endl;  
    return 0;  
}
```

donut 면적은 3.14
pizza 면적은 2826

예제 3-1 Circle 클래스 정의와 객체 생성

14

(1) `Circle donut;`

객체 이름

객체가 생성되면
메모리가 할당된다.

멤버함수의 코드
가 저장된 메모리

int radius
double getArea() {...}

donut 객체

(2) `donut.radius = 1;`

int radius
double getArea() {...}

donut 객체

(3) `double area = donut.getArea();`

main()

area

int radius
double getArea() {...}

donut 객체

예제 3-1 Circle 클래스 정의와 객체 생성

15

(1) Circle pizza;

객체 이름

객체가 생성되면
메모리가 할당된다.

멤버함수의 코드
가 저장된 메모리

int radius
double getArea() {...}

pizza 객체

(2) pizza.radius = 30;

int radius
double getArea() {...}

pizza 객체

(3) area = pizza.getArea();

main()

area

int radius
double getArea() {...}

pizza 객체

예제 3-2 Rectangle 클래스 만들기

16

- 다음 main() 함수가 잘 작동하도록 너비(width)와 높이(height)를 가지고 면적 계산 기능을 가진 Rectangle 클래스를 작성하고 전체 프로그램을 완성하라.

```
....  
int main() {  
    Rectangle rect;  
    rect.width = 3;  
    rect.height = 5;  
    cout << "사각형의 면적은 " << rect.getArea() << endl;  
}
```

사각형의 면적은 15

예제 3-2 Rectangle 클래스 만들기

17

사각형의 면적은 15

```
#include <iostream>
using namespace std;
class Rectangle {
public:
    int width;
    int height;
    int getArea();
};
int Rectangle::getArea() {
    return width * height;
}
int main() {
    Rectangle rect;
    rect.width = 3;
    rect.height = 5;
    cout << "사각형의 면적은 " << rect.getArea() << endl;
}
```

// Rectangle 클래스 선언부

// 면적을 계산하여 리턴하는 함수

// Rectangle 클래스 구현부

실습과제1

18

- 구조체와 클래스의 차이점을 설명하라.
- 클래스와 객체의 차이점을 설명하라.
- `sizeof`(클래스명)의 결과값은 어떻게 계산될지 설명하라.

실습과제2

19

- 삼각형 클래스 Triangle 를 만들고 예제 3.2와 비슷하게 동작하는 프로그램을 작성하시오.
- 멤버변수의 초기값은 마음대로 하고 면적값은 실행결과와 같을 필요 없음
- (힌트) 멤버변수 -> 밑변, 높이, 면적-> $(1/2) * \text{밑변} * \text{높이}$

삼각형의 면적은 xxx

실습과제3

20

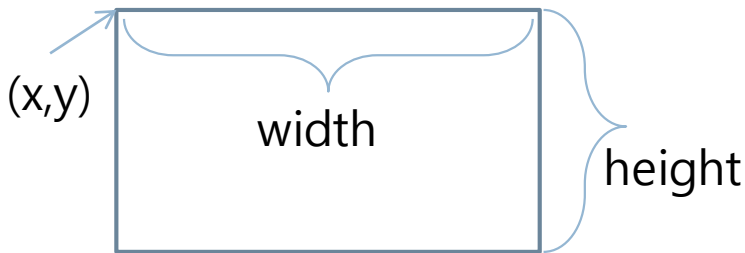
- 구 클래스 Sphere 를 만들고 예제 3.2와 비슷하게 동작하는 프로그램을 작성 하시오.
- 멤버변수의 초기값은 마음대로 하고 멤버함수는 부피를 구해주는 함수, 표면적을 구해주는 함수를 작성하라.
- (힌트) 멤버변수 -> 반지름, 부피-> $(4.0/3.0)*3.14*\text{반지름}^3$, 표면적 -> $4*3.14*\text{반지름}^2$

구의 부피는 xxx
구의 표면적은 xxx

실습과제4

21

- 예제3-2에서 다음기능을 추가하라.
 - ▣ 멤버변수에 사각형의 좌측상단의 좌표를 저장하는 2개의 변수 x, y 를 추가
 - ▣ 사각형의 둘레의 길이를 구해주는 멤버함수 추가
 - ▣ 사각형의 우측하단의 좌표를 구해주는 멤버함수 추가
 - ▣ 멤버변수의 초기값은 $x=1, y=2, width=3, height=4$ 으로 설정할 것



사각형의 면적은 12
사각형의 둘레길이는 14
사각형의 우측하단의 좌표는 (4, -2)

실습과제5

22

- 실습과제 4번에서 다음기능을 추가하라.
 - ▣ 키보드로부터 데이터를 받아서 멤버변수에 대입해주는 멤버함수 추가, 아래 실행결과를 참고할 것

사각형의 좌측상단좌표(x,y) : 1 2<엔터>
사각형의 폭과 높이(width,height) : 3 4<엔터>
사각형의 면적은 12
사각형의 둘레길이는 14
사각형의 우측하단의 좌표는 (4, -2)

과제 제출 방법

23

- 소스코드, 라인단위의 주석, 실행결과를 포함하는 pdf파일을 작성한 후 eclass 과제 게시판에 업로드, **반드시 하나의 pdf파일로 업로드할 것**
- 기한 : 과제 게시판에 마감시간 참조
- 실행결과를 캡처할 때 글자를 알아보기 쉽게 확대해서 캡처할 것.
- 소스코드의 첫 부분은 아래처럼 제목,날짜,작성자(학번,이름)를 작성할 것

```
// *****  
//   제   목   : 정수 4개의 평균을 구하는 프로그램  
//   날   짜   : 2023년 9월10일  
//   작성자   : 15010101 홍길동  
// *****  
  
// 소스코드 작성
```