

클래스

클래스의 종류

클래스

특성

- **public** : 접근할 수 있는 권한
- **protected** : 클래스의 내부에서만 접근할 수 있는 권한
- **private** : 클래스의 외부에서 접근할 수 없는 권한 **struct**는 접근할 수 있는 **class**와 유사하다

클래스의 구성

- 클래스의 구성 요소(멤버)
 - 멤버 함수(*void*)
 - 멤버 변수
- 클래스의 생성자
 - 생성자(*const person &p*) : 클래스의 객체를 생성할 때 호출되는 함수
- 클래스의 소멸자
 - 소멸자(*person()*)
 - 클래스의 객체가 소멸될 때 호출되는 함수
 - 멤버 변수
- 클래스의 정적 멤버
 - 정적 멤버 함수(*void*)
 - 정적 멤버 변수

클래스의 사용

- **c++**의 클래스는 3가지로 나뉜다
 - 1. 클래스의 선언
 - 2. 클래스의 정의
 - 3. 클래스의 사용
- 클래스의 선언은 **c++**의 헤더 파일에서 이루어진다
- 클래스의 정의는 **c++**의 소스 파일에서 이루어진다

클래스의 생성

- 클래스의 생성 방법
 - 클래스의 객체 생성
- 클래스의 생성 방법
 - `new`와 `new`를 사용하여 생성

클래스의 소멸

클래스의 소멸 방법: `1(1) 2(2) ... { }` 형태로 생성

클래스

- 클래스의 `static` 멤버

- 構造体
 - 構造体変数
 - 構造体ポインタ
 - 構造体配列
 - 構造体
 1. 構造体
 2. 構造体変数: 構造体
- 構造体
 - 構造体変数
 - 構造体ポインタ(構造体変数ポインタ)

C++ 構造体

- 構造体変数
- 構造体ポインタ((*)構造体)構造体変数ポインタ.構造体変数ポインタ

this

this構造体変数ポインタ

- 構造体変数ポインタthis
- 構造体変数ポインタ(return *this) 構造体変数ポインタ 構造体変数ポインタ
- 構造体変数const構造体this
- 構造体変数mutable(構造体変数)

友元(friend)

友元関数

1. 友元関数
 - 友元関数friend
2. 友元関数
 - 友元関数friend
3. 友元関数
 - 友元関数

友元関数

友元関数(友元関数) **operator** 友元関数(友元関数){友元関数}

友元関数

友元関数

- 友元関数cout
- 友元関数

```
ostream &operator<<(ostream &cout, person &p)
{
    cout << p.m_A << p.m_B;
    return cout; //coutにpの値を出力
}
```

C++ 構造体

- 前置++运算符: `operator++()` {};
- 后置++运算符: `operator++(int){}`;

例

```
class Counter{  
public:  
    int value = 0;  
    Counter(){}  
    Counter(int v): value(v){}  
    void increment() {value++;}  
    void decrement() {value--;}  
    int get() const {return value;}  
};
```