

# 28장. 네임스페이스

## 01\_ 네임스페이스 사용하기

# 네임스페이스 안에서 정의하기(1)

- 네임스페이스 안에서 변수, 함수, 구조체, 클래스를 정의하는 예

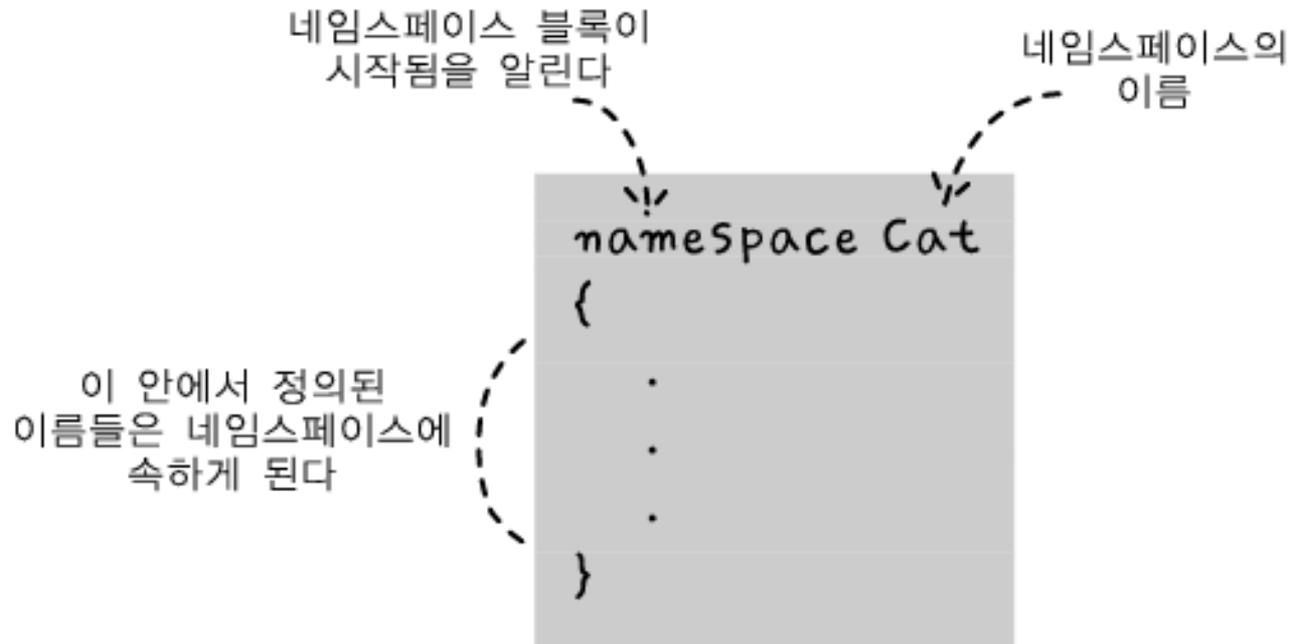
```
namespace Dog
{
    struct Info
    {
        char name[20];
        int age;
    };
    Info dogs[20];        // 멍멍이 리스트
    int count;           // 전체 멍멍이들의 수
    void CreateAll();    // 모든 멍멍이 생성 함수
}

namespace Cat
{
    class Info
    {
    public:
        void Meow();
    protected:
        char name[20];
    };
    Info cats[20];       // 야옹이 리스트
    int count;           // 전체 야옹이 들의 수
    void CreateAll();   // 모든 야옹이 생성 함수
}

// 전체 멍멍이와 야옹이들의 수
int count;
```

# 네임스페이스 안에서 정의하기(2)

- 네임스페이스를 사용하기



# 네임스페이스 지정하기

- 네임스페이스를 지정해서 네임스페이스 안에 정의한 이름 사용하기

```
// 모든 야옹이를 생성한다.  
Cat::CreateAll();  
  
// 야옹이 배열에 접근한다.  
Cat::cats[0].Meow();  
  
// 모든 멍멍이를 생성한다.  
Dog::CreateAll();  
  
// 멍멍이의 개수를 얻어온다.  
int dog_count = Dog::count;
```

# using 키워드 사용하기

- using 키워드로 사용할 네임스페이스 지정하기

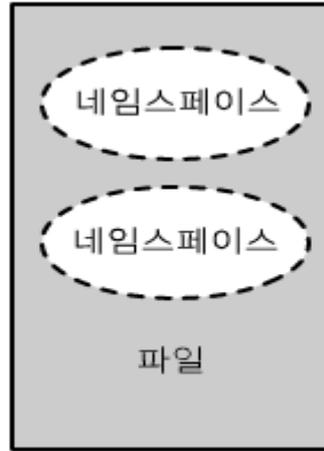
```
using namespace Cat;  
  
// 모든 야옹이를 생성한다.  
CreateAll();  
  
// 야옹이 배열에 접근한다.  
cats[0].Meow();
```

- using 키워드로 사용할 이름 지정하기

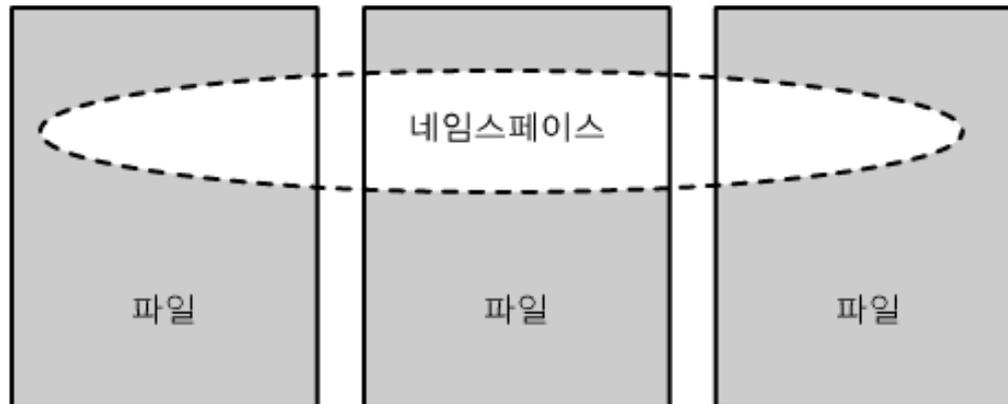
```
using Cat::CreateAll;  
  
// 모든 야옹이를 생성한다.  
CreateAll();
```

# 네임스페이스와 소스 파일

- 하나의 파일에 여러 개의 네임스페이스가 등장할 수 있다.



- 여러 파일에 걸쳐서 하나의 네임스페이스가 등장할 수 있다.



# 이름 없는 네임스페이스

- 이름 없는 네임스페이스 안에 정의된 이름은 다른 소스 파일에서 접근할 수 없다. (static 전역 변수, 함수와 동일한 효과)

```
// Example.h
// a.cpp에서 정의된 g를 사용하기 위한 준비
extern int g;

int main()
{
    g = 200; // 실패

    return 0;
}
```

```
namespace
{
    int g;
}

void Func()
{
    g = 200; // 성공
}
```

# 중첩된 네임스페이스

- 네임스페이스 안에 또 다른 네임스페이스를 만들 수 있다.

```
namespace Data
{
    namespace User
    {
        int number;
    }
}

int main()
{
    int user_number = Data::User::number;

    return 0;
}
```

# 네임스페이스를 별칭으로 부르기

- 네임스페이스에 별칭을 붙여줄 수 있다.

```
namespace this_namespace_has_a_very_long_name
{
    int n;
}

// 별칭을 붙여준다.
namespace oh = this_namespace_has_a_very_long_name;

int main()
{
    oh::n = 100;

    return 0;
}
```

# Q&A