LValue RValue

남동강

LValue RValue

LValue

- **메모리의 식별자**가 있어 **값의 변경이 가능**
- 값의 변경이 불가능한 lvalue : **배열, const 한정자가 붙은 Type**

RValue

- 메모리의 식별자가 없어 값의 **변경이 불가능한 임시 값**
- ex) **상수, 연산에 의하여 생성된 결과 값, enum**

RValue Reference(&&)

```
int lvalue = 10;
enum { rvalue = 10 };
int& a = lvalue;
int& b = rvalue;  // 비const 참조에 대한 초기 값은 lvalue여야 합니다.
int&& c = lvalue; // rvalue 참조를 lvalue에 바인딩할 수 없습니다.
int&& d = rvalue;
```

LValue RValue

- **이동 생성자**
- **얕은 복사**로 **메모리 이사**를 한다
- 이동 연산자(=) 또한 만들 수 있다

```
class Type
{
private:
    int* data;

public:
    Type() : data(new int(0)) { std::cout << data << " 기본 생성자 호출" << std::endl; }
    Type(const Type& t) : data(new int(0))
    {
        if (nullptr != t.data) *data = *t.data;
        std::cout << data << " 복사 생성자 호출" << std::endl;
    }
    Type(Type&& t) noexcept : data(t.data) // 이동 생성자
    {
        t.data = nullptr;
        if (nullptr == data) data = new int(0);
        std::cout << data << " 이동 생성자 호출" << std::endl;
    }

    ~Type() { if (data) { std::cout << "소멸자 호출" << std::endl; delete data; data = nullptr; } }
};
```

LValue RValue

```
#define _CRTDBG_MAP_ALLOC
#include <crtdbg.h> // 메모리 누수 확인.

int main()
{
    _CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF | _CRTDBG_LEAK_CHECK_DF);

    Type t = std::move(Type()); // std::move() 객체가 이동할 수 있음을 알려준다.
                                // 이동을 수행하지 않는다!!
    std::cout <<"-----------------------------------" << std::endl;
    std::vector<Type> v;
    v.push_back(t);
    v.push_back(t);
    v.push_back(t);
    std::cout <<"-----------------------------------" << std::endl;

    return 0;
}
```

LValue RValue

- **연습문제**

① 이동 생성자의 **t.data = nullptr 코드**를 **주석**하여 실행 후 **결과**와 **이유를 유추**하여 봅시다

② **이동 생성자를 주석**하여 실행 후 **결과**와 **이유를 유추하**여 봅시다