

🏠 / C++프로그래밍과실습 (CB3500572-062) / 실습 067 - destructor (RAII, noexcept)

개요

코딩 결과

실습 067 - destructor (RAII, noexcept)

제출 마감일: 2023-04-30 23:59

업로드 가능한 파일 수: 3

제출 방식: 개인

목적

이 실습은 class 의 소멸자 정의와 RAII 개념을 연습합니다.

설명

객체의 라이프타임에서 객체의 생성은 생성자(constructor)가 담당하며, 객체의 소멸 시 소멸자(destructor) 가 사용한 자원을 해제합니다.

실습에서 정의한 스트링 클래스는 내부적으로 new 연산자를 이용하여 메모리를 할당 받아 사용합니다.

이제 스트링 클래스에 소멸자를 추가해야 할 시점입니다.

소멸자는 ~클래스명() 으로 정의하며, 클래스별로 단 하나만 존재합니다.

~String();

(C.36) core guidelines 에서는

소멸자는 실패하면 안된다고 합니다.

STL은 소멸자에서 예외를 던지지 않는 것을 요구합니다.

~String() noexcept;

C++ 의 가장 유명한 이디엄 중 하나인 RAII (Resource Acquisition Is Initialization) 은 객체가 획득한 자원은 객체 소멸 시 깨끗하게 정리해야 한다는 것입니다.

우리의 스트링 객체는 new 연산자로 배열 메모리를 획득 했으므로, 소멸자에서 delete[] 연산자로 해당 메모리를 해제해야 합니다.

```
delete[ ] s;
```

문제

String 클래스의 소멸자를 정의하시오.

단, copy/move semantics 이 구현되지 않을 경우 double free 예외가 발생할 수 있습니다.

<참고>

```
// StringDestructorTest.cpp
```

```
#include "String.h"
```

```
int main() {  
    String str1{"abc"};  
    String str2 = str1;  
    str2.at(0) = 'b';  
    str1.print("str1");  
    str2.print("str2");  
}
```

입력

없음

출력

str1: abc, size: 3

str2: bbc, size: 3

Destructor

Destructor

제출파일

String.cpp

String.h

67.csv

