2023 Spring OOP Assignment Report

과제 번호 : 4
학번 : 20220778
이름 : 표승현
Povis ID : hyeony312

명예서약 (Honor Code)

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.
I completed this programming task without the improper help of others.

프로그램을 하다 보면 결정해야 할 세부 사항이 많은데, 이러한 세부 사항을 처리한 방법과 이유를 보고서에 쓰십시오.
독창적인 아이디어와 추가 기능은 보너스 점수를 받을 수 있으므로, 보고서에 명확히 기재하십시오.
문제가 여러 개인 경우, 각 문제별로 정리해서 작성합니다.

아래 문항별 설명은 편의를 위한 것으로, 삭제하고 제출한다.

1. 프로그램 개요

1) SharedPtr 템플릿 클래스를 구현하는 것을 목표로 한다. SharedPtr은 동적으로 할당된 객체에 대한 소유권을 공유하는 스마트 포인터이다. 여러 개의 SharedPtr 객체가 동일한 객체를 참조할 수 있으며, 객체의 수명은 참조 카운터를 통해 추적된다. 이러한 스마트 포인터를 사용하면 메모리 누수를 방지하고 자동으로 객체의 수명을 관리할 수 있다.

2) 이미지를 나타내기 위한 Image 클래스와 RGB 픽셀 타입을 정의한다. 또한, 이미지의 픽셀 타입을 변환하거나 BMP 파일로 저장하고 로드하는 몇 가지 보조 함수도 포함되어 있다. 이미지의 정보를 저장하기 위한 공간을 1)에서 구현한 스마트 포인터를 통해 할당하여 사용한다. 이러한 헤더파일을 완성하여 이미지 처리 장치를 만든다.

2. 프로그램의 구조 및 알고리즘

SharedPtr.h의 구성은 아래와 같다.

1. Deallocator: 동적으로 할당된 개체를 해제하는 함수이다. Deallocator는 일반 객체에 대한 메모리 해제를 처리하는 Deallocator(ObjectType* ptr) 함수와 배열 객체에 대한 메모리 해제를 처리하는 ArrayDeallocator(ObjectType* ptr) 함수로 구현된다. 이 함수들은 객체의 메모리를 해제하기 위해 delete 또는 delete[] 연산자를 사용한다.

2. DeallocatorFuncType: Deallocator 함수 포인터 타입을 정의하는 별칭 템플릿이다.

3. SharedPtr 클래스: 동적으로 할당된 객체에 대한 소유권을 공유하는 스마트 포인터를 구현한다. 해당 클래스에는 다음과 같은 멤버가 있다:

- m_object: 포인터로 동적으로 할당된 객체를 가리킨다.

- m_ref_counter: 객체를 참조하는 SharedPtr 인스턴스의 수를 추적하는 참조 카운터이다.

- 생성자 및 소멸자: 기본 생성자, 명시적 생성자 및 소멸자를 구현한다.

- 대입 연산자: SharedPtr 객체 간의 대입 연산을 처리한다.

- 포인터 연산자: operator-> 및 operator*를 오버로딩하여 객체에 대한 포인터 연산을 지원한다.

- 첨자 연산자: operator[]를 오버로딩하여 배열 객체에 대한 요소 접근을 지원한다.

- 형 변환 연산자: operator ObjectType*()를 오버로딩하여 SharedPtr 객체를 해당 객체 유형에 대한 포인터로 변환할 수 있도록 한다.

4. SharedArray: SharedPtr 템플릿 클래스를 사용하여 배열에 대한 스마트 포인터를 표현한다. 이 클래스는 ArrayDeallocator 함수를 사용하여 배열 객체의 메모리를 해제한다.

Image.h 의 구성은 아래와 같다.

1. RGB 구조체: RGB 픽셀 값을 나타내는 구조체이다. RGB<ValType> 형식으로 정의되며, ValType은 픽셀 값의 데이터 유형을 나타낸다. 구조체에는 r, g, b 필드 또는 data 배열을 사용하여 각각 R, G, B 채널의 값을 저장할 수 있다. 또한, operator[]를 사용하여 인덱스를 통해 각 채널에 접근할 수 있다.

2. Image 클래스 템플릿: 이미지 데이터를 나타내는 클래스이다. PixelType은 이미지의 픽셀 타입을 나타내는 템플릿 매개변수이다. Image 클래스에는 다음과 같은 멤버가 있다:

- m_width: 이미지의 너비를 저장하는 변수.

- m_height: 이미지의 높이를 저장하는 변수.

- m_buff: 동적으로 할당되는 이미지 데이터를 저장하기 위한 SharedArray 포인터.

- 생성자 및 소멸자: 기본 생성자, 크기 및 초기 값으로 생성되는 생성자, 복사 생성자 및 소멸자를 구현한다.

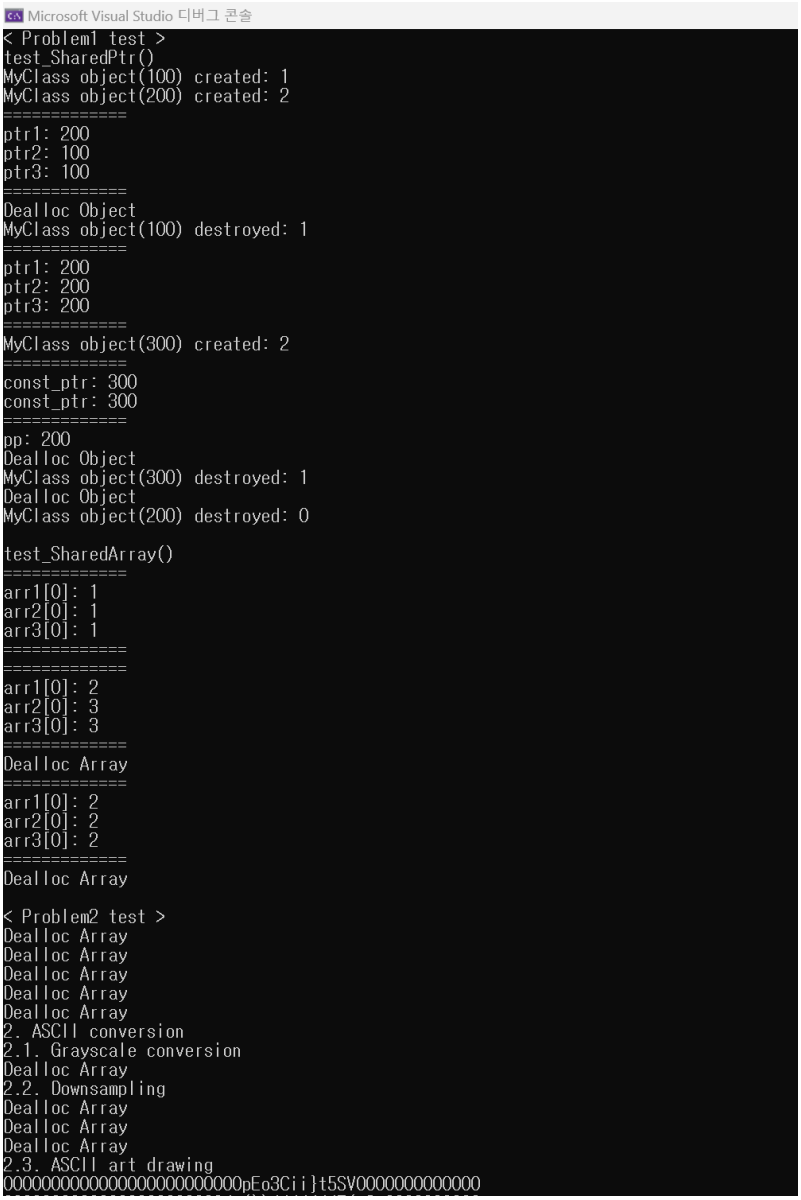
- 대입 연산자: 같은 픽셀 타입을 사용하는 다른 이미지로부터 이미지 데이터를 복사하는 대입 연산자를 구현한다.

- 첨자 연산자: operator[]를 오버로딩하여 이미지의 각 픽셀에 접근할 수 있도록 한다.

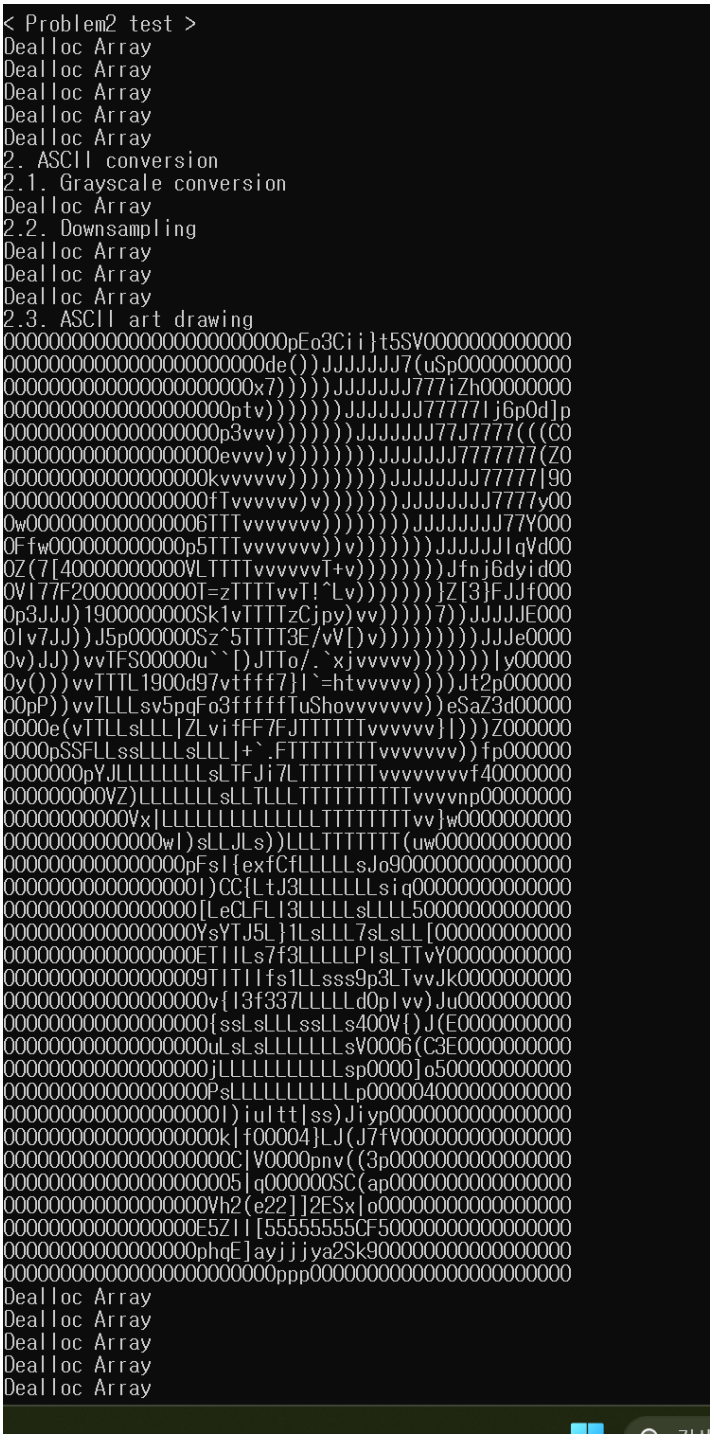
- 기타 메소드: 이미지의 너비와 높이를 반환하는 width()와 height() 메소드를 구현한다.

3. convert_pixel_type 함수: 이미지의 픽셀 타입을 다른 픽셀 타입으로 변환하는 함수다. RGB8b에서 RGBf로의 변환 및 RGBf에서 RGB8b로의 변환을 처리한다.

4. load_bmp 및 save_bmp 함수: BMP 파일을 로드하거나 저장하는 함수이다. load_bmp 함수는 주어진 경로의 BMP 파일을 로드하여 RGB8b 픽셀 타입의 이미지로 변환한다. save_bmp 함수는 주어진 경로에 RGB8b 픽셀 타입의 이미지를 BMP 파일로 저장한다.



Test 1을 실행한 결과이다.



Test 2를 실행한 결과이다.

개선할 점:

1. SharedPtr

Null을 검사하는 기능을 추가한다. 현재의 SharedPtr 클래스는 null 포인터를 지원하지 않

는다. 개선을 위해 null 포인터 검사를 추가하여 프로그램의 안전성을 높일 수 있다. 예를 들어, 포인터가 null인 경우에 대한 예외 처리를 추가하여 잘못된 참조를 방지할 수 있다.

2. Image class

형식 및 채널 지원을 확장하여 개선할 수 있다. 현재의 Image 클래스는 RGB 이미지를 지원한다. 그러나 다른 형식 (예: 그레이스케일, RGBA)이나 다양한 채널 수를 지원하도록 개선할 수 있다. 이를 위해 템플릿 매개변수를 활용하여 유연한 이미지 형식 및 채널 지원을 구현할 수 있다.

3. 참고 문헌