

2022 Spring OOP Assignment Report

과제 번호 : ASSN #4

학번 : 20210479

이름 : 이주현

Povis ID : lanthantum14

명예서약 (Honor Code)

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

1. 프로그램 개요

- ASSN #4는 기하 객체인 point와 rectangle 객체를 정의하고, std::vector를 상속한 class인 canvas2d에 이 객체들을 저장 및 출력하는 기능을 수행하기 위해 필요한 모든 class를 구현해보는 문제이다.
- 각각의 기하 객체 point와 rectangle은 geometry라는 abstract class를 상속한 class로 구성하였다. 이 기하 객체들을 관리하는 canvas2d는 std::vector<geometry*>를 상속한 것으로 용이하게 구현할 수 있었다.
- 프로그램 실행을 위해서는 main.cpp를 실행하면 되며 따로 입력해줄 파일 등을 준비할 필요는 없다. main.cpp를 실행하고 "problem1" 또는 "problem2"를 입력하여 각 문제에 해당하는 동작 실행을 시작한다.

2. 프로그램의 구조 및 알고리즘

- geometry Class: 기하 객체를 다루는 pure virtual method를 가진 abstract class이다. 이를 이용하여 아래 두 개의 class를 구현한다. 두 개의 class 모두 geometry class를 상속받는 class이다. 또한 아래의 두 class 모두 template class이므로 헤더 파일에 함수의 내용이 같이 구현되어 있다.
 - point Class: 2~5차원의 int/float 좌표를 입력받을 수 있는 class이다.
 - rectangle Class: 4개의 int/float 좌표를 입력받을 수 있는 class이다. 이때 사각형의 면적이 0이 되는 입력, 즉 $x1 == x2$ 이거나, $y1 == y2$ 인 입력이 들어왔을 경우 throw를 통해 에러 메시지를 띄우며 프로그램을 종료(i.e., abort() 호출)한다.
- canvas2d Class: std::vector<geometry*>를 상속받은 class로, vector의 멤버 함수들을

사용할 수 있다. 추가로 width(정수), height(정수), canvas(char 2차원 배열)라는 멤버 변수들을 가지며, 이를 통해 저장된 기하 객체들을 그려 출력한다.

- parser Class: 입력받은 string을 쪼개어 canvas와 geometry 객체를 만들어주는 class 이다. 용이한 동작을 위해 stringstream을 사용한다.
- Problem1
 - 우선 canvas2d 생성을 위한 'width,height,c_empty' 순서의 입력을 해줘야 한다. 이때 canvas2d class와 parser class가 상호작용하여 canvas2d 객체가 만들어진다.
 - 그 후 '[geometry종류],typename,[좌표]' 순서의 입력을 지속하여 geometry 객체를 생성할 수 있다. 이 입력이 들어올 경우 parser class와 geometry / point / rectangle class가 상호작용하여 기하 객체가 만들어진다.
 - 입력을 종료하고 싶다면 ctrl+Z(window의 EOF)를 입력하면 된다. EOF 입력 시 지금까지 입력한 geometry 객체가 모두 그려진 canvas가 출력된 후 프로그램이 종료된다.
- Problem2
 - 'typename,[좌표]' 형태의 입력을 하여 bool 값 다섯 개를 출력받는다. rectangle의 멤버 함수를 사용하여 실행되는 동작이다. 이 입력은 제한 없이 반복할 수 있으며, 입력을 종료하고 싶다면 위와 동일하게 EOF를 입력하면 된다. EOF 입력 시 추가 statement 없이 프로그램이 종료된다.

3. 토론 및 개선

- 지난번 ASSN부터 학습한 class inheritanc를 사용하는 방법을 복습할 수 있었다. 또한 virtual method를 이용한 polymorphism에 대해서 복습할 수 있었다.
- 이번 과제를 통하여 template class를 사용하는 방법을 알 수 있었다. 특히 template class를 구현하고자 할 때는 일반 class처럼 header 파일과 cpp 파일을 분리하는 것이 아니라 header 파일 내부에 선언과 구현을 모두 해야 한다는 점을 숙지할 수 있었다.
- point와 rectangle은 template class이므로, main과 parser의 코드를 일부 변경하여 int 나 float 외의 다른 자료형을 사용할 수 있도록 만들어 기능을 확장할 수 있을 것 같다.

4. 문제 3에 관한 서술

- STL vector를 상속받을 때 예상되는 문제점은 STL vector의 소멸자가 virtual이 아니라는 것이다. 이에 상속받은 class 자체의 소멸자는 호출되지만, vector의 소멸자가 호출되지 않아 문제가 발생할 가능성이 높다. 항상 문제가 생기는 것은 아니지만 vector 소멸자 호출이 꼭 필요한 상속 class를 만들고자 할 경우 심각한 문제가 될 수 있다.

5. 참고 문헌

- <https://ya-ya.tistory.com/101>
 - c++의 new를 이용한 2차원 배열 동적할당 방법 참고
- <https://dojang.io/mod/page/view.php?id=307>
 - 2차원 배열의 인덱스 순서 참고
- <https://www.sapphosound.com/archives/389>
 - template class의 선언과 구현을 같이해야 하는 이유 참고