**2022 Spring OOP Assignment Report**

과제 번호 : 4

학번 : 20210054

이름 : 정하우

Povis ID : howru0321

**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.

각 문항별 설명은 편의를 위한 것으로, 삭제하고 제출한다.

1. **프로그램 개요**
   * 본 프로그램은 assn3 과제를 통해 배운 class initializer 개념과 기존에 사용할 수 없었던 여러 STL을 적절히 사용하는 프로그램이다.
   * Problem1은 canvas class를 통해 우리가 직접 rectangle이나 point를 그릴 canvas를 설정하고 그 위에 위치정보를 받아 rectangle class와 point class를 통해 canvas위에 그리고 이를 출력하는 프로그램이다.
   * Problem2는 좌표평면위에 두 사각형의 위치 정보를 받아서 사각형 간의 관계를 파악하는 프로그램을 구현하는 것이다.
2. **프로그램의 구조 및 알고리즘**

* class canvas2d : public vector<geometry\*> STL vector를 상속 받아 구현되며 보유한 객체를 stdout으로 그려내는 class

void draw2stdout() 보유한 객체를 stdout으로 출력

* class geometry 모든 geometry 구성 요소들의 abstract class

class rectangle :public geometry 2차원 평면을 가정한 사각형 class

void draw2canvas(vector<vector<int>>& board) {

int max\_x\_temp = (int)max(this->x\_1, this->x\_2);

int max\_y\_temp = (int)max(this->y\_1, this->y\_2);

int min\_x\_temp = (int)min(this->x\_1, this->x\_2);

int min\_y\_temp = (int)min(this->y\_1, this->y\_2);

int max\_x = min(max\_x\_temp, (int)board[0].size()-1);

int max\_y = min(max\_y\_temp, (int)board.size()-1);

int min\_x = max(min\_x\_temp, 0);

int min\_y = max(min\_y\_temp, 0);

for (int j = min\_x; j <= max\_x; j++) {

for (int i = min\_y; i <= max\_y; i++) {

board[i][j] += 1;

}

}

}

bool is\_equal(const rectangle<T>& other) {//멤버함수를 호출하는 instance와 other이 정확히 4개의 point를 공유하는 경우 true, 그렇지 않은 경우 false

bool is\_intersect(const rectangle<T>& other) {//멤버함수를 호출하는 instance와 other이 equal이 아니면서 영역이 겹치는 경우 true, 그렇지 않은 경우 false

bool is\_line\_overlap(const rectangle<T>& other) {//멤버함수를 호출하는 instance와 other이 equal 및 intersect가 아니면서 한 변을 공유하는 경우 true, 그렇지 않은 경우 false

bool is\_point\_match(const rectangle<T>& other) {//멤버함수를 호출하는 instance와 other이 intersect가 아니면서 오직 한 점을 공유하는 경우 true, 그렇지 않은 경우 false

bool is\_disjoint(const rectangle<T>& other) {//멤버함수를 호출하는 instance와 other이 만나지 않는 경우 true, 그렇지 않은 경우 false

void draw2canvas(vector<vector<int>>& board) {

if ((float)x > (float)((int)board[0].size() - 1) || (float)y > (float)((int)board.size() - 1))

return;

int pos\_x = (int)x;

int pos\_y = (int)y;

board[pos\_y][pos\_x] += (int)num;

}

class point :public geometry {//포인트 class로 T는 사용될 numeric type, N은 coordinate dimension을 의미한다

void get\_pos(T \_x, T \_y) point의 위치를 설정해 주는 함수. 위치좌표가 private이므로 public함수를 통해 구현해 주었다.

* class parser string을 받아 객체의 instance를 생성하는 기능을 제공하는 class

1. **토론 및 개선**
   * + 기본적으로 오류는 이번에 새로배운 개념에서 나온 것이 아니라 힙 개념 부족으로 인해 동적할당을 하지 않고 정적할당을 하거나 pure virtual function개념 부족으로 인해 나타는 오류가 많았다. 기존에 배웠던 개념을 확실히 복습을 해야겠다고 생각했고, 이번 개념 역시 복습을 꾸준히 해야겠다는 다짐을 하게 되었다.
2. **참고 문헌**