

2022 Spring OOP Assignment Report

과제 번호 : ASSN3_prob1

학번 : 20210716

이름 : 최대현

Povis ID : daehyeonchoi

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.

1. 프로그램 개요

개발환경: Visual Studio 2022

*로 채워진 15x15 보드에 도형을 그리는 프로그램을 구현한 코드이다. 클래스의 상속과 overriding 개념을 명확히 알고 있어야 구현할 수 있는 과제이다.

2. 프로그램의 구조 및 알고리즘

링크드 리스트

추상적 자료형(ADT)인 리스트를 구현한 자료구조로, 어떤 노드(Node)를 저장할 때 그 다음 순서의 자료가 있는 위치를 데이터에 포함시키는 방식으로 자료를 저장하는 자료구조이다. 자유롭게 구현할 수 있는 AShapelist를 단방향 연결리스트로 구현하였다.

사용자 정의 자료형 설명

Menu, AShpae, AShapelList, 가진 APoint, ARectangle, AVerticalLine, AHorizontalLine라는 이름을 가진 class를 구현했다. AShape class에서 가장 중요한 것은 이 클래스를 서로 다른 모양을 가진 APoint, ARectangle, AVerticalLine, AHorizontalLine으로 상속시켜서 사용했다는 점이다. 그

뒤 각 자식클래스에는 draw함수와 print 함수를 virtual로 구현하여 상황에 맞는 기능을 할 수 있도록 overriding하여 구현했다.

자료구조 설명

1) Menu class

```
#pragma once
#include<iostream>
#include "ashapelist.h"
#include "whiteboard.h"
class Menu
{
private :
    bool is_running;
    AShapeList* shape_list;
    WhiteBoard* board;

public:
    Menu(WhiteBoard* board, AShapeList* shape_list);
    ~Menu();
    void display() const;
    bool getRunning() const;
    void getUserInput();
    void addAShape() const;
    void deleteAShape() const;
};
```

Private:

멤버 변수 설명

Bool is_running: 프로그램이 구동되고 있는지 여부를 알려주는 변수이다.

AShapeList *shape_list: 도형이 담긴 리스트이다. AShapeList라는 typename을 가진 또다른 클래스이고, 단방향 연결리스트로 구현하였다.

WhiteBoard * board: 출력의 대상인 board이다. WhiteBoard라는 typename을 가진 또다른 클래스이고, 단방향 연결리스트로 구현하였다.

public:

멤버 함수 설명

void display() const: 메뉴를 출력하는 함수이다.

bool getRunning() const: is_running을 리턴한다.

void getUserInput(): user의 입력을 받아 기능을 수행하는 함수를 호출한다.

void addAShape() const: 1번을 입력할 경우 호출되는 함수로 도형을 추가한다.

void deleteAShape() const: 2번을 입력할 경우 호출되는 함수로 원하는 인덱스의 도형을 제거한다.

2) AShape class

```
class AShape
{
protected:
    char alphabet;
public:
    AShape();
    virtual ~AShape();
    virtual void draw(WhiteBoard* board) const = 0;
    virtual void print() const = 0;
    virtual int size() const = 0;
};
```

protected:

멤버 변수 설명

char alphabet: 이 도형을 구성할 알파벳, protected 멤버 변수이므로 후에 자식 클래스에 상속됨.

public:

AShape(): 기본 생성자 , 초기화는 매뉴얼 규칙을 따랐음

virtual ~AShape(): 소멸자, virtual 함수로 구현하였다.

virtual void draw(WhiteBoard* board) const = 0: virtual 함수로 구현, board에 해당 도형을 그리는 함수

virtual void print() const = 0: 도형의 정보를 출력하는 함수

virtual int size() const = 0: 도형의 size를 리턴하는 함수

```
class APoint :public AShape
{
private:
    int x;
    int y;

public:
    APoint();
    APoint(int x_, int y_, char alphabet_);
    ~APoint();
    void draw(WhiteBoard* board) const;
    void print() const;
    int size() const;
};
```

```
class AVerticalLine :public AShape
{
private:
    int x;
    int y_start;
    int y_end;

public:
    AVerticalLine();
    AVerticalLine(int x_, int y_start_, int y_end_, char alphabet_);
    ~AVerticalLine();
    void draw(WhiteBoard* board) const;
    void print() const;
    int size() const;
};
```

```

class AHorizontalLine :public AShape
{
private:
    int x_start;
    int x_end;
    int y;

public:
    AHorizontalLine();
    AHorizontalLine(int y_, int x_start_, int x_end_, char alphabet_);
    ~AHorizontalLine();
    void draw(WhiteBoard* board) const;
    void print() const;
    int size() const;
};

```

```

class ARectangle :public AShape
{
private:
    int x_start;
    int y_start;
    int x_end;
    int y_end;

public:
    ARectangle();
    ARectangle(int x_start_, int y_start_, int x_end_, int y_end_, char alphabet_);
    ~ARectangle();
    void draw(WhiteBoard* board) const;
    void print() const;
    int size() const;
};

```

3) APoint Class, AVerticalLine Class, AHorizontalLine Class, ARectangle Class

AShape class를 상속받아 추가적인 기능을 수행하는 클래스들이다.

Public:

멤버 변수 설명

x,y, 좌표가 들어있음. 각 도형의 성질에 맞게 시작 좌표와 끝 좌표가 다른 등의 특성이 반영되어 있다.

Private:

멤버 함수 설명

각각 기본 생성자, 대입 생성자가 들어 있고, derived class에서 virtual로 구현한 소멸자, draw 함수, print 함수가 들어있다.

```

struct AShapeNode {
    AShape* shape;
    AShapeNode* next;
};

class AShapeList {
private:
    AShapeNode* head;
    AShapeNode* tail;
    int size;

public:
    AShapeList();
    ~AShapeList();
    void addNewAShape(AShape* new_shape);
    AShapeNode* popNodeByIdx(int idx);
    const AShapeNode* getNodeByIdx(int idx) const;
    void displayAShapeList() const;
    void drawAll(WhiteBoard* board) const;
    int getSize() const;
};

```

4) AShapeList class

Struct AShapeNode : AShapeList의 노드가 되는 자료형이다. AShape와 다음 노드에 대한 주소를 담은 next를 멤버변수로 갖는 자기참조구조체이다.

class AShapeList

private:

멤버 변수 설명

AShapeNode* head : 리스트의 맨 처음 노드를 가리키는 포인터 변수이다.

AShapeNode* tail: 리스트의 맨 마지막 노드를 가리키는 포인터 변수이다.

int size: 리스트의 사이즈를 저장하는 변수이다.

public:

AShapeList(): 리스트의 디폴트 생성자이다. Dummy head node를 만들어서 삽입과 삭제가 용이하게 구성하였다.

~AShapeList(): 리스트의 소멸자이다. Delete를 하지 않은채로 exit 커맨드를 입력하여 프로그램을 종료하는 경우를 고려하여 리스트의 노드들을 모두 동적할당 해제 한 뒤, 리스트를 동

적할당 해제한 후 프로그램을 종료한다.

void addNewAShape(AShape* new_shape): 리스트의 맨 마지막에 노드를 삽입하는 함수이다. 이 때, 노드 자체를 가져오는 것이 아니라 AShape object를 가져와서 노드를 생성한 뒤 대입해주는 절차를 거쳐서 리스트에 삽입해준다.

AShapeNode* popNodeByIdx(int idx): 매개변수 idx를 받아와서 그 인덱스에 해당하는 노드를 떼준 뒤 주소를 반환하는 함수이다. 노드의 주소를 반환하면서도 리스트에서 노드를 떼내기 때문에, 이 함수를 호출하는 즉시 동적할당을 해제해줘야 한다.

const AShapeNode* getNodeByIdx(int idx) const: idx번째 노드를 리턴하는 함수이다. Pop 함수와 다르게 그 노드를 리스트에서 삭제하는 것이 아니고 그대로 놔둔 채 리턴한다.

void displayAShapeList() const: 리스트를 첫 노드부터 출력한다.

void drawAll(WhiteBoard* board) const: 보드에 현재 리스트에 있는 대로 도형을 그려주는 함수이다. Virtual로 지정된 draw함수를 첫 노드부터 size만큼 호출한다.

int getsize() const: size를 반환한다. Size를 외부에서 가져오기 위해 사용한다.

```
#pragma once

class WhiteBoard {
private:
    char board[15][15];
public:
    WhiteBoard();
    ~WhiteBoard();
    void fillpoint(int x, int y, char alphabet);
    void reset();
    void display() const;
};
```

5) Class WhiteBoard

private:

멤버변수 설명

char board[15][15]: 출력을 위한 15x15 보드이다.

public:

멤버변수 설명

WhiteBoard(): whiteboard의 기본 생성자이다.

~WhiteBoard(): 소멸자

void fillpoint(int x, int y, char alphabet): 보드를 채우는, 실질적으로 그려주는 함수

void reset(): 보드의 모든 인덱스 값을 *로 초기화하는 함수

void display() const: 보드를 출력하는 함수

3. 토론 및 개선

getNodeByIdx 함수를 올바르게 구현했지만 프로그래밍을 끝낸 후에 이 함수를 사용하면 효율적으로 프로그래밍할 수 있다는 것을 깨달았다. 그 점이 아쉽고, 다음부터는 멤버함수를 다른 멤버함수에 효율적으로 사용해보고 싶다. 또한, AShapelist 에 AShapeNode를 추가할 때 이것을 if문 안에서 포인터를 사용하지 않고 구현해서 의도치 않은 지역변수가 되어버리는 바람에 abort has been called() 오류가 발생하여 고생했는데, 소멸되는 지역변수를 참조해서 생기는 오류였다.

함수의 overriding과 클래스의 상속에 대해 명확히 깨달을 수 있는 의미있는 시간이었다.

4. 참고 문헌