

## CSED 232 Object-Oriented Programing (Spring 2022)

### Programming Assignment # 3

#### - Inheritance & Polymorphism -

Due date: 4 월 29 일

담당 조교: 이지형 (jihyung.lee@postech.ac.kr)

#### 주의사항

- 입출력을 위한 **iostream** 을 제외한 라이브러리의 사용은 허용하지 않습니다. 단, 기존 과제(Assignment2 에서 구현했던 **vector** 등)에서 직접 짜신 코드는 사용하셔도 됩니다.
- 각 클래스마다 header file 과 cpp file 을 나누어 구현하여야 합니다. **AShape** 관련 클래스들의 경우 아래 그림과 같이 **ashape.h** 와 **ashape.cpp** 에 선언과 정의를 합쳐서 작성하여도 괜찮습니다.

ashape.h	ashape.cpp
<div>⟨선언⟩ AShape APoint AHorizontalLine AVerticalLine ARectangle</div>	<div>⟨정의⟩ AShape APoint AHorizontalLine AVerticalLine ARectangle</div>












- 문제에 명시되어 있지 않더라도 **각 클래스마다 생성자(Constructor), 소멸자(Destructor)**는 필수입니다. 기본 생성자의 경우, 별다른 언급이 없다면 멤버 변수에 int 는 0, char 는 'a', string 은 "", pointer 는 NULL 로 초기화해 주십시오.
- 클래스에 대한 헤더파일과 cpp 파일의 파일명을 통일하여 주십시오. (대소문자 유의)
- 모든 클래스의 멤버 변수는 protected, 혹은 private 으로 선언하여야 합니다.
- 채점 시 사용되는 정확한 테스트 케이스는 공개되지 않습니다. 이는 테스트 케이스에 맞춰 필요한 부분만 프로그래밍하는 경우를 방지하기 위한 것이니 양해해주시길 바랍니다.

#### 감점

- 제출 기한이 지나면 얻은 총점의 20% 감점
- 추가로 하루(24 시간) 늦을 때마다 20%씩 감점
  - 1 일 이내 지연: 20% 감점, 2 일 이내 지연: 40% 감점, 5 일 이상 지연: 0 점
- 컴파일이 정상적으로 이루어지지 않을 경우 0 점

## 제출방식

채점은 Windows Visual Studio 2022(윈도우 사용자의 경우) 및 Ubuntu 20.04(lts)와 gcc version 9.4.0 (맥 사용자의 경우) 환경에서 이루어집니다. VS 로 작업했을 경우 작업하신 환경이 있는 visual studio 프로젝트 폴더에 Report 를 포함하여 zip 파일로 압축 후 제출해 주시기 바랍니다. (x64 및 .vs 폴더는 전부 지워주십시오) 마찬가지로, 맥 이용자의 경우 소스 코드, 보고서, Makefile 을 포함한 폴더를 압축해서 제출해주시면 됩니다. 폴더명은 '#학번'으로 만들어 주시고, Report 는 docx 나 pdf 형식으로 제출해주세요. 반드시 PLMS 를 통해 제출해주시기 바랍니다. 이메일 제출은 인정되지 않습니다. 5 일 이 지날 경우 0 점입니다.

[윈도우 이용자]	[맥 이용자]
<div data-bbox="448 748 528 835"> x64  .vs</div> <p data-bbox="213 875 767 1003">VS 프로젝트로 제출 시 위 폴더를 전부 <b>지우고</b> 제출해주시십시오. x64 폴더는 두개 있습니다. (.vs 폴더는 숨겨져있음)</p> <div data-bbox="209 1037 759 1545"><p data-bbox="555 1037 632 1070">폴더 안</p><div data-bbox="209 1077 759 1545"><div data-bbox="209 1122 392 1216"> assn3  assn3.sln  assn3_report.pdf</div><div data-bbox="448 1115 722 1514"><ul style="list-style-type: none"><li>*+ ashape.cpp</li><li>ashape.h</li><li>*+ ashapelist.cpp</li><li>ashapelist.h</li><li>assn3_reborn.vcxproj</li><li>assn3_reborn.vcxproj.filters</li><li>assn3_reborn.vcxproj.user</li><li>*+ main.cpp</li><li>*+ menu.cpp</li><li>menu.h</li><li>*+ whiteboard.cpp</li><li>whiteboard.h</li></ul></div></div></div> <p data-bbox="233 1570 751 1648">위쪽 파일들을 포함하여 학번을 폴더명으로 하여 압축해주세요</p>	<div data-bbox="986 757 1201 1205"><ul style="list-style-type: none"><li> whiteboard.h</li><li>*+ whiteboard.cpp</li><li> menu.h</li><li>*+ menu.cpp</li><li> Makefile</li><li>*+ main.cpp</li><li> assn3_report.pdf</li><li> ashapelist.h</li><li>*+ ashapelist.cpp</li><li> ashape.h</li><li>*+ ashape.cpp</li></ul></div> <p data-bbox="871 1256 1390 1335">위와 같이 Makefile 파일을 포함시켜 학번을 폴더명으로 하여 압축해주세요</p>

## 채점 기준

### 1. 프로그램 기능

- 프로그램이 요구 사항을 모두 만족하면서 올바르게 실행되는가?

### 2. 프로그램 설계 및 구현

- 요구 사항을 만족하기 위한 변수 및 알고리즘 설계가 잘 되었는가?
- 설계된 내용이 요구된 언어를 이용하여 적절히 구현되었는가?

### 3. 프로그램 가독성

- 프로그램이 읽기 쉽고 이해하기 쉽게 작성되었는가?
- 변수 명이 무엇을 의미하는지 이해하기 쉬운가?
- 프로그램의 소스 코드를 이해하기 쉽도록 주석을 잘 붙였는가?

### 4. 보고서 구성 및 내용, 양식

- 보고서는 적절한 내용으로 이해하기 쉽고 보기 좋게 잘 작성되었는가?
- 보고서의 양식을 잘 따랐는가?

다른 사람의 프로그램이나 인터넷에 있는 프로그램을 복사(copy)하거나 간단히 수정해서 제출하면 학점은 무조건 'F'가 됩니다. 이러한 부정행위가 발견되면 학과에서 정한 기준에 따라 추가의 불이익이 있을 수 있습니다.

## 알파벳 도형 그리기

본 과제에서는 '알파벳 도형(Alphabet Shape, 통칭 AShape)'을 WhiteBoard 에 그리는 기능을 직접 구현해보면서 Class Inheritance 와 Polymorphism 개념을 익 혀보도록 한다.

### • 프로그램 설명



```
C:\Users...
[WhiteBoard]
=====
*****
*****
*ttttt*****
*ttttt*****
*ttttt*****
*****
*****
*z*****
*****
*****
*****
*****
*****qqq*
*****
=====
[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>
```

'알파벳 도형 그리기' 프로그램은 15x15 사이즈의 WhiteBoard 에 **알파벳을 값으로 가지는 도형 네 종류(점, 수평선, 수직선, 사각형)**을 그리는 프로그램이다. WhiteBoard 는 처음에는 비어있으며, 비어있는 칸은 \*을 출력한다. 이 프로그램은 다음 세 가지 기능을 지원해야한다.

1. WhiteBoard 에 새로운 도형 추가
2. WhiteBoard 에 현재 그려져 있는 도형 삭제
3. 프로그램 종료

아래 설명을 잘 읽고, 요구 사항에 맞게 프로그램을 구현해보도록 하자.

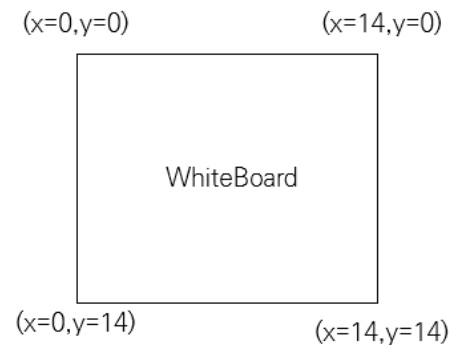
## 1. 프로그램 기능

### 1.1 메뉴



```
C:\Users\#... [WhiteBoard]
=====
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
=====
[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>
```

프로그램 시작 화면



WhiteBoard 좌표에 대한 그림

프로그램 시작 시 비어있는 WhiteBoard 와 함께 메뉴가 출력된다. '1'을 입력할 시 도형 추가, '2'를 입력하면 도형 삭제, '3'을 입력하면 프로그램이 종료된다.

WhiteBoard 는 15x15 의 고정된 크기를 지닌다. 오른쪽 그림과 같이 x 값은 왼쪽부터 오른쪽으로 갈수록 커지고, y 값은 위쪽에서 아래로 내려올수록 커진다. 가장 왼쪽 위 좌표는 (0,0), 가장 오른쪽 아래 좌표는 (14,14)이다.

## 1.2 WhiteBoard 에 새로운 도형 추가

```
C:\Users\W... - □ ×
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
=====
[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>1
[Add New AShape]
Enter Alphabet >>q
Enter x_start >>1
Enter x_end >>3
Enter y_start >>2
Enter y_end >>2
```

유저로부터 알파벳, x 좌표 및 y 좌표들을  
입력받는 화면.

```
C:\Users\W... - □ ×
Enter x_start >>1
Enter x_end >>3
Enter y_start >>2
Enter y_end >>2
New AHorizontalLine added w
ith size:3
[WhiteBoard]
=====
*****
*****
*qqq*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
=====
[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>_
```

‘q’를 알파벳으로 가지고 좌표가 (1~3,2),  
길이가 3 인 수직선이 추가된 화면

메뉴 1 번, 도형 추가로 들어가면 프로그램은 유저로부터 (1)알파벳 및 (2) x 시작좌표와 (3) x 끝좌표 (4) y 시작좌표와 (5) y 끝좌표 총 다섯번을 입력 받는다.

프로그램은 유저가 입력한 좌표 값이 어느 도형에 해당하는지 판단하여, **추가되는 도형의 종류와 크기를 출력하고 WhiteBoard 에 그린다.** 새 도형이 추가될때마다 동적 할당이 일어나야 한다.

도형의 종류는 1. 점(APoint) 2. 수직선(AVerticalLine) 3. 수평선(AHorizontalLine) 4. 사각형(Rectangle) 총 네 가지이며 그 특징은 다음과 같다.

1. **점(APoint):** 크기는 1 이며, x 시작좌표와 끝좌표 및 y 시작좌표와 끝좌표가 각각 같다.
2. **수직선(AVerticalLine):** 크기는 선의 길이와 같으며, x 시작좌표와 끝좌표가 같지만 y 시작좌표와 끝좌표가 다르다.
3. **수평선(AHorizontalLine):** 수직선의 반대로, y 시작좌표와 끝좌표가 같지만 x 시작좌표와 끝좌표가 다르다.
4. **사각형(ARectangle):** 크기는 사각형의 넓이와 같으며, x 및 y 시작좌표와 끝좌표가 서로 다르다.



### 1.3 WhiteBoard 에 현재 그려져 있는 도형 삭제

```
New APoint added with size:1
[WhiteBoard]
=====
*****
*****
*qqq*****
*****
*****
***zzzz*****
***zzzz*****
***zzzz*****
***zzzz*****
*****
*****
*****
=====
[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>2
Choose index of AShape that you want to remove
[=====List of AShape=====]
Current Size: 3
0) AHorizontalLine with [alphabet: q] and [y: 2 ] [x_start: 1 ] [x_end: 3] [size: 3]
1) ARectangle with [alphabet: z] and [x_start: 4 ] [x_end: 7] [y_start:5] [y_end:9] [size: 20]
2) APoint with [alphabet: u] and [x: 7 ] [y: 7] [size: 1]
>>
```

메뉴에서 '2'를 입력할 경우 현재 WhiteBoard 에 그려져 있는 도형을 삭제할 수 있다. 도형 삭제 메뉴로 들어갈 시 **현재 WhiteBoard 에 추가되어있는 도형의 개수가 출력되어야 하며, 도형 각각의 정보(좌표, 알파벳, 크기)가 보여야 한다.**

그림의 예제에서는 1.2 에서 실행했던 것과 같이 수직선, 사각형, 점 총 세 가지의 도형이 추가되어 있으므로 리스트에는 세 종류의 도형이 출력된다. 도형 리스트 인덱스는 0 부터 시작한다.

이 때, 사용자가 삭제를 원하는 도형의 index 를 입력할 시 해당 도형이 WhiteBoard 에서 삭제된다. 도형 삭제 시 동적 할당된 메모리가 해제되어야한다.

```
>>2
Choose index of AShape that you want to remove
[=====List of AShape=====]
Current Size: 3
0) AHorizontalLine with [alphabet: q] and [y: 2 ] [x_start: 1 ] [x_end: 3] [size: 3]
1) ARectangle with [alphabet: z] and [x_start: 4 ] [x_end: 7] [y_start:5] [y_end:9] [size: 20]
2) APoint with [alphabet: u] and [x: 7 ] [y: 7] [size: 1]
>> 2
[WhiteBoard]
=====
*****
*****
*qqq*****
*****
*****
***zzzz*****
***zzzz*****
***zzzz*****
***zzzz*****
*****
*****
*****
=====
[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>
```

리스트 인덱스 '2'를 입력하여 맨 마지막에 추가한 좌표 (7,7)의 점(APoint)을 삭제한 케이스



```

=====
*****
*****
*qqq*****
*****
*****
***zzzz*****
***zzzz*****
***zzzz*****
***zzzz*****
***zzzz*****
*****
*****
*****
=====
[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>2
Choose index of AShape that you want to remove
[=====List of AShape=====]
Current Size: 2
0) AHorizontalLine with [alphabet: q] and [y: 2 ] [x_start: 1 ] [x_end: 3] [size: 3]
1) ARectangle with [alphabet: z] and [x_start: 4 ] [x_end: 7][y_start:5][y_end:9] [size: 20]
>>

```

이어서 좌표 (4-7, 5-9)에 알파벳 z 를 가지는 사각형을 지워보겠다.

```

[=====List of AShape=====]
Current Size: 2
0) AHorizontalLine with [alphabet: q] and [y: 2 ] [x_start: 1 ] [x_end: 3] [size: 3]
1) ARectangle with [alphabet: z] and [x_start: 4 ] [x_end: 7][y_start:5][y_end:9] [size: 20]
>> 1
[WhiteBoard]
=====
*****
*****
*qqq*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
=====
[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>

```

리스트에서 사각형의 index '1'을 입력하니 다음과 같이 도형이 지워진 것을 볼 수 있다.

#### 1.4 도형이 겹치는 경우

도형이 겹치는 경우를 막을 필요는 없다. 도형이 겹치는 경우 WhiteBoard 에 추가된 순서대로 그린다.  
(새 도형이 기존의 도형을 덮어씀)

단, 위를 덮은 도형을 삭제한다면 기존의 도형이 다시 보여야한다. (1.3 도형 삭제의 두번째 예제 사진 참조)

## 1.5 프로그램 종료

```
=====
[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>3
Exit program
```

메뉴에서 '3'을 입력 시 'Exit Program' 메시지 출력과 함께 프로그램이 종료된다.

## 1.6 예외 처리

다음과 같은 케이스에서 예외를 처리해야한다.

### 1. 메뉴

```
*****
=====
[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>-2392
Error: Invalid input. Type again
[WhiteBoard]
=====
*****
```

**잘못된 숫자 입력이 들어오는 경우:**

'Invalid input. Type again' 에러 메시지와 함께 WhiteBoard 및 메뉴가 재출력된다.

### 2. 도형 추가

```
[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>1
[Add New AShape]
Enter Alphabet >>t
Enter x_start >>5
Enter x_end >>3
Enter y_start >>5
Enter y_end >>5
Error: x_start exceed x_end
[WhiteBoard]
=====
*****
```

**[A] x\_start 가 x\_end 보다 크거나,**

**y\_start 가 y\_end 보다 큰 경우:**

시작좌표가 끝좌표를 넘어섰다는 에러 메시지와 함께 WhiteBoard 및 메뉴가 재출력된다.

```

[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>1
[Add New AShape]
Enter Alphabet >>u
Enter x_start >>-1
Enter x_end >>3
Enter y_start >>0
Enter y_end >>0
Error: coordinate cannot be negative value
[WhiteBoard]
=====

```

**[B] 좌표중에 하나라도 음수가 들어오는 경우:**

좌표는 음수가 되지 못한다는 에러 메시지와 함께 WhiteBoard 및 메뉴가 재출력된다.

```

[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>1
[Add New AShape]
Enter Alphabet >>k
Enter x_start >>1
Enter x_end >>3
Enter y_start >>14
Enter y_end >>17
Error: your input exceed whiteboard size
[WhiteBoard]
=====
*****

```

**[C] 좌표중에 하나라도 화이트보드의 크기를 벗어난는 경우:**

입력 좌표가 WhiteBoard 크기를 벗어났다는 에러 메시지와 함께 WhiteBoard 및 메뉴가 재출력된다.

```

[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>1
[Add New AShape]
Enter Alphabet >>t
Enter x_start >>5
Enter x_end >>5
Enter y_start >>-3
Enter y_end >>-5
Error: y_start exceed y_end
[WhiteBoard]
=====
*****

```

**여러 에러가 중복으로 발생하는 경우**

에러 메시지 우선 출력 순위는  $A > B > C$  이다.  
예를 들어,  $y\_start$  가  $y\_end$  보다 크면서 음수가 들어올 경우 에러케이스 A 에 대한 메시지가 출력되어야 한다.

### 3. 도형 삭제

```

=====
[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>2
AShapeList is empty.
[WhiteBoard]
=====
*****
*****

```

**[A] 도형 리스트가 비어있는 경우:**

리스트가 비어있다는 에러 메시지와 함께 WhiteBoard 및 메뉴가 재출력된다.

```

[=====]
[Select Menu]
[=====]
1. add new shape
2. delete existing shape
3. exit
>>2
Choose index of AShape that you want to r
[=====List of AShape=====]
Current Size: 2
0) AHorizontalLine with [alphabet: u] and
x_start: 3 ] [x_end: 5] [size: 3]
1) APoint with [alphabet: k] and [x: 10 ]
size: 1]
>> -3
Error: Index exceed size of AShapeList
[WhiteBoard]
=====
*****

```

#### [B] 도형 리스트 인덱스에서 벗어나는 입력이

##### 들어올 경우:

인덱스를 벗어났다는 에러 메시지와 함께 WhiteBoard 및 메뉴가 재출력된다. 음수가 들어오는 경우 & 리스트 사이즈를 초과하는 입력이 들어올 경우 둘다 해당된다.

에러 메시지는 각 에러 케이스가 구분되기만 한다면 예제와는 다른 메시지를 출력해도 상관없다. 단, 이 경우 보고서에 해당 에러 메시지가 어떤 케이스에 해당되는 것인지 설명해야 한다.

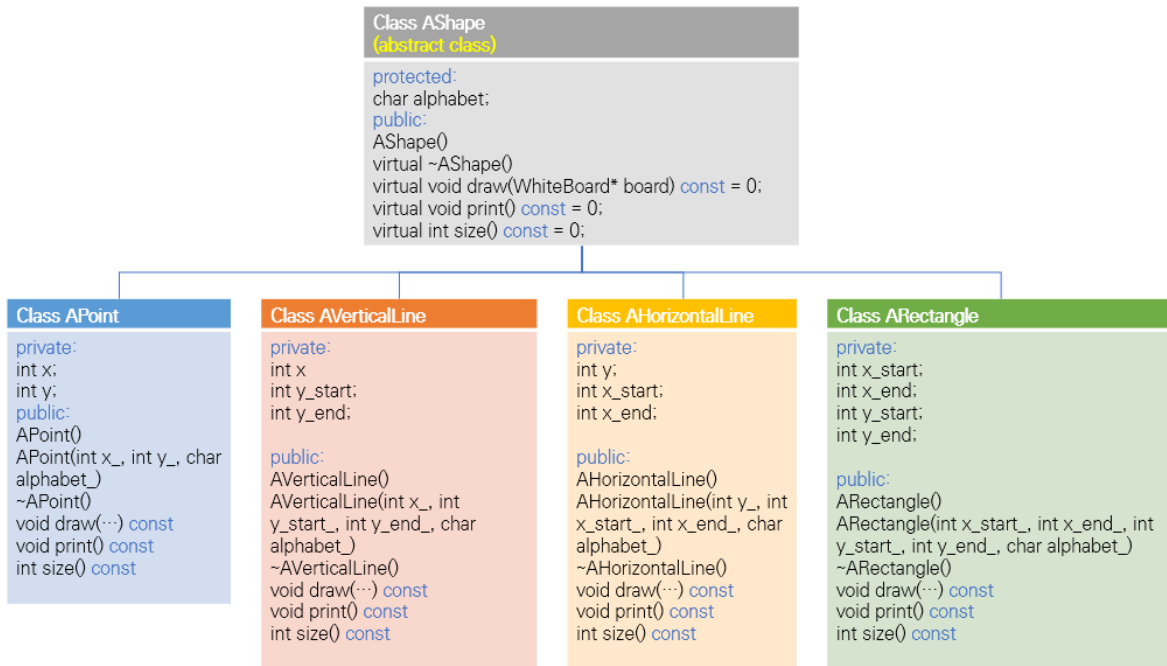
위에서 언급되지 않은 부분에 대해서는 예외 처리를 생각하지 않아도 된다.

(ex) 메뉴 입력 시 문자가 들어오는 경우, 도형 추가 시 알파벳에 이상한 값이 들어오는 경우 등... 은 고려할 필요 없다.

## 2. 구현

### 2.1 AShape (Alphabet Shape) 관련 클래스

아래 그림에 서술된 AShape 관련 멤버 함수 및 멤버 변수는 반드시 구현해야 하며, 이름을 변경해서는 안된다. 필요에 따라 자유롭게 멤버 함수를 추가해도 된다.



**AShape 클래스:** 순수 가상 함수를 가지는 abstract class 이며, 다른 shape 들의 parent class 이다. draw, print, size 를 pure virtual 함수로 정의한다.

**APoint, AVerticalLine, AHorizontalLine, ARectangle 클래스:** AShape 를 상속받는 클래스

각 AShape 관련 클래스들은 아래의 멤버 함수를 구현해야 한다.

- 생성자 및 소멸자
- **void draw(WhiteBoard\* board) const:** 화이트보드에 해당 도형의 좌표 및 알파벳의 정보를 보내 그리게 한다.
- **void print() const:** 해당 도형을 설명하는 내용을 출력하는 함수. 도형의 좌표값 및 갖고 있는 알파벳, 크기를 전부 출력한다. 도형 삭제 - 리스트 출력 시 도형을 설명하는 데에서 쓰인다.
  - ex) 수평선의 경우 AHorizontalLine with [alphabet: q] and [y: 13] [x\_start: 10 ] [x\_end: 12] [size: 3] 이와 같이 출력하면 된다. 반드시 위와 같은 양식으로 출력할 필요는 없고, 필요한 정보만 담겨있으면 된다.
- **int size() const:** 도형의 크기를 리턴하는 함수. 점의 경우 1, 선의 경우엔 선의 길이, 사각형의 경우에는 사각형의 넓이다.

## 2.2 AShapeList



도형들의 리스트를 관리하는 클래스. 예제 그림은 linked list 로 구현한 사례이며, 반드시 linked list 를 이용할 필요는 없다. 본인이 생각하기에 더 나은 자료 구조 및 방법이 있다면 그것을 이용해도 상관없다. 이 경우 필요한 멤버 함수와 멤버 변수를 자유롭게 설계해보도록 한다.

assignment2 에서 구현한 vector 의 코드를 참고하는 것도 좋은 방법이다. **단, 모든 코드는 직접 구현해야 하며 STL 라이브러리는 사용하면 안된다.** 어떤 방법을 이용하였는지 보고서에 적도록 한다.

새 도형이 추가될 때마다 동적할당이 일어나므로, AShapeList 의 소멸자에서는 할당된 메모리를 해제해주는 것을 잊지 말자.

- **void addNewAShape(AShape\* new\_shape):** 리스트에 새 도형을 추가한다.
- **AShapeNode\* popNodeByIdx(int idx):** 인덱스를 받아 리스트에서 해당 AShapeNode 를 빼내고, 빼낸 노드의 포인터를 리턴한다. (ex) idx 0 이 들어올때 첫번째 Node 를 리턴한다.
- **const AShapeNode\* getNodeByIdx(int idx) const:** 인덱스를 받아 리스트에서 인덱스에 해당하는 순번의 AShapeNode 의 포인터를 리턴한다. popNodeByIdx 와 달리, 노드를 리스트에서 빼지 않는다.
- **void displayAShapeList() const:** 현재 관리중인 모든 도형의 리스트를 출력한다. 도형 삭제 시 도형의 정보를 한꺼번에 출력할 때 쓰인다.
- **void drawAll(WhiteBoard\* board) const:** 현재 관리중인 모든 도형을 WhiteBoard 에 그린다.
- **int getSize() const:** 현재 리스트가 갖고 있는 도형의 개수를 리턴한다.

위 멤버 함수는 예제이며, AShape 의 목록을 관리할 방법에 따라 자유롭게 설계하도록 한다.

## 2.3 WhiteBoard

```
Class WhiteBoard
private:
char board[15][15];

public:
WhiteBoard()
~WhiteBoard()
void fillPoint(int x, int y, char alphabet)
void reset()
void display() const
```

WhiteBoard class 는 지정된 크기의 2 차원 char 배열을 멤버 변수로 가지며, 다른 클래스에서 호출된다.

- **void fillPoint(int x, int y, char alphabet):** 지정된 x,y 좌표에 들어온 alphabet 으로 칸을 채운다.
- **void reset():** 모든 칸을 \*로 초기화한다.
- **void display() const:** whiteboard 를 출력한다.

## 2.4 Menu

```
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() const
void addAShape() const
void deleteAShape() const
```

Menu class 는 메뉴 클래스이며, 도형 추가, 도형 삭제, 프로그램 종료 등 등 프로그램 전반에 대해 처리한다.

- **void display() const:** 메뉴를 출력한다.
- **bool getRunning() const:** 현재 프로그램이 실행 중인지의 여부를 true, false 로 리턴한다.

- **void getUserInput() const:** 유저의 입력을 받아 메뉴를 고르게 하고, 이후 addAShape, deleteAShape 등을 호출한다.
- **void addAShape() const:** 유저의 입력을 받아 새 도형을 추가한다.
- **void deleteAShape() const:** 유저의 입력을 받아 도형 삭제를 진행한다.

## 2.5 main

메인 파일의 예제이며, 반드시 아래의 코드를 사용해야하는 것은 아니다. 자신이 구현한대로 main 함수를 설계해보자.

```
#include "menu.h"

int main()
{
    WhiteBoard board;
    AShapeList shape_list;
    Menu menu(&board, &shape_list);

    while (menu.getRunning())
    {
        //WhiteBoard Display
        board.reset();
        shape_list.drawAll(&board);
        board.display();

        //Menu Display & Get Input
        menu.display();
        menu.getUserInput();
    }

    return 0;
}
```

## 채점 기준

### 1. 프로그램 기능 – 50%

- 메뉴 화면 5%
- 도형 추가 20%
- 도형 삭제 20%
- 예외 처리 5%



2. 프로그램 설계 및 구현 – 30%

- AShape 및 AShape 관련 class 10%
- AShapeList 10%
- WhiteBoard, Menu, main 10%

3. 프로그램 가독성 – 10%

4. 보고서 구성 및 내용, 양식 – 10%