**Lab #4: MakeAGraph 설계 발표 및 평가**

**마감: 4월 1일**

객체지향 설계와 패턴

2021년 봄학기

**개 요**  
이번 실습은 자신이 설계한 MakeAGraph를 설명하고 동료가 설계한 MakeAGraph를 평가하는 것이다.

2명씩 구성된 소회의실에서 만나 다음과 같은 작업을 하여 제출한다.

1. 지난 실습에서 작성하여 제출한 자신의 MakeAGraph에 대한 설계를 상대편에게 설명한다.

2. 상대가 설명한 설계에 대하여 별도의 평가지를 이용하여 평가하라.

빈 칸에 해당되는 질문에 대하여 평가하고 평가 이유를 자세히 적어라. 피상적인 답이 아니라 깊은 사고에 의한 설득력 있는 답을 적어야 한다. 좀 더 설계를 깊게 이해하고 신중한 평가를 하여 구현에 도움이 되도록 하라.

설계 리뷰한 사람: \_\_\_\_\_\_\_\_\_\_김희수\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

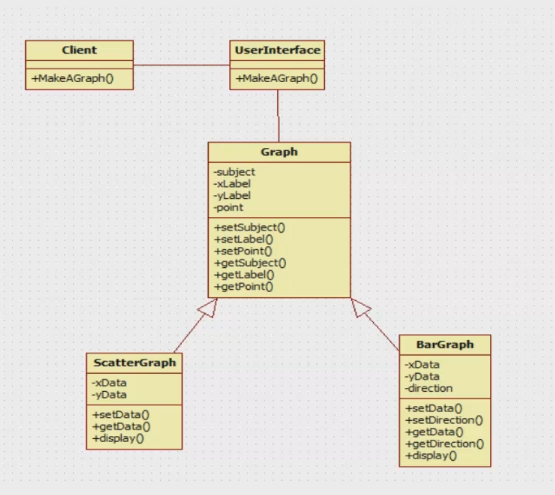
설계 작성한 사람: \_\_\_\_\_\_\_\_\_\_이서연\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

PART 1: 일반적 질문

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 번호 |  | | 1 | 2 | 3 | 4 | 5 |
| 1 | 전체적인 설계의 이해용이성을 1(low)부터 5(high)까지 매긴다면? | |  |  |  | 4 |  |
| 그 이유는? | | 설계자체는 단순하여 이해하기 편했다 | | | | | |
| 2 | 이 설계를 당신이 구현한다면 설계가 얼마나 도움이 되는가? 1(low)부터 5(high)까지 매긴다면? | | 1 |  |  |  |  |
| 그 이유는? | | 변경에 너무 민감하다. 변경에 대해 열려있고, 구체적인것에 의존하고 있는 설계이다. 구현을 자체는 쉽게 되겠지만 미래에 구현을 변경할 일이 생긴다면 설계가 도움되지 않을 것 같다. | | | | | |
| 3 | 이상적인 설계는 구현자가 볼 때 상위 설계 결정을 잘 반영하여 빠뜨리지 않아야 한다. 전반적인 완벽성 수준을 매긴다면? | |  | 2 |  |  |  |
| 그 이유는? | | 설계자는 ScatterGraph, BarGraph등을 Graph를 상속하여 설계하였는데, 이 부분은 괜찮다고 생각하지만 전반적으로 보면 UserInterface에서 바로 Graph 클래스가 연관되는건 좋지 않다고 본다. 파이차트 같은걸 반영하려면 Graph클래스보다 상위의 인터페이스가 있어야할 것 같다. | | | | | |
| 4 | 분할을 적절히 하였는지 그 수준을 평가하라(예, 얼마나 쉽게 담당자들이 설계의 각 부분을 각각 구현할 수 있는지) 1 (low) 부터 5 (high): | |  |  | 3 |  |  |
| 그 이유는? | | 구현자체는 쉽게 할 수 있을 것 같다. 다만, 구현’만’ 쉬울 것이라 예상된다. 또한, ScatterGraph, BarGraph에 너무 많은 책임이 들어있다 생각된다. ScatterGraph의 데이터포인트나 BarGraph의 막대는 각각 Scatter, Bar클래스로 책임을 분할할 수 있다고 생각된다 | | | | | |

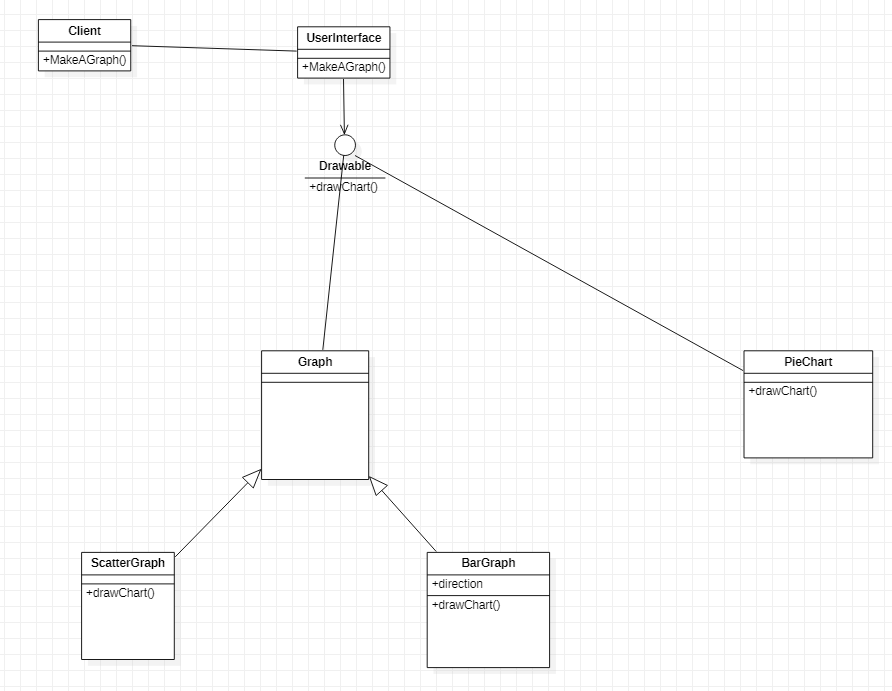
PART 2: 구현 전에 일어난 변경

설계자의 클래스 다이어그램은 다음과 같다.



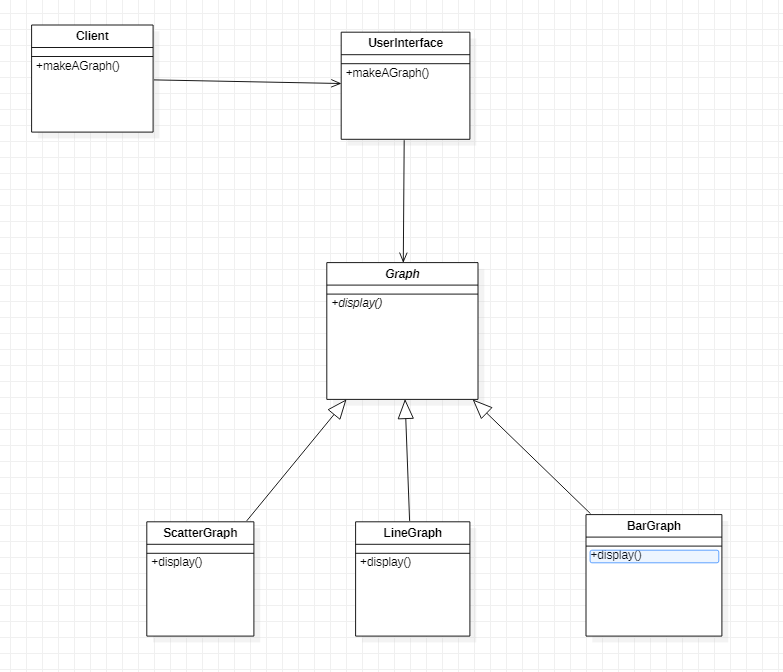
5. 설명 받은 설계에 대하여 아직 구현하지 않았다 그런데 새로운 그래프의 타입, 예를 들어 파이 챁, 같은 것을 추가하여야 한다고 하자. 설계에 어떤 영향을 줄 것인지 적어보라(예를 들어 어떤 클래스와 인터페이스가 어떻게 왜 영향을 받을지). 설명에 도움이 된다면 다이어그램을 그려도 좋다.

새로운 그래프(예를 들어 파이차트)가 추가된다면, 현재 존재하는 Graph 클래스로는 부족하다고 생각한다. 새로운 클래스를 정의해야한다. 파이차트는 x축, y축이 없을텐데 Graph클래스엔 xLabel, yLabel이 있기 때문이다. 파이차트 클래스를 정의하고 UserInterface에 붙인다고 가정할 시, UserInterface클래스에 MakeAPie메소드도 정의해야 한다. 하지만 이런 식으로 클래스가 새롭게 늘어날때마다 메소드를 추가해야 한다면 의존관계 역전에 어긋난다고 생각된다. 구체적인 클래스 자체에 의존하기 때문이다. 차트를 그리는 역할을 하는 Drawable인터페이스를 만들고 ScatteGraph와 BarGraph의 수퍼크클래스인 Graph클래스를 Drawable을 인터페이스 상속해서 만들고, 파이차트 또한 Drawable을 구현해서 만든다면 추상적인 것(Drawable인터페이스)에 의존하게 되어 의존관계 역전을 지킬 수 있다.

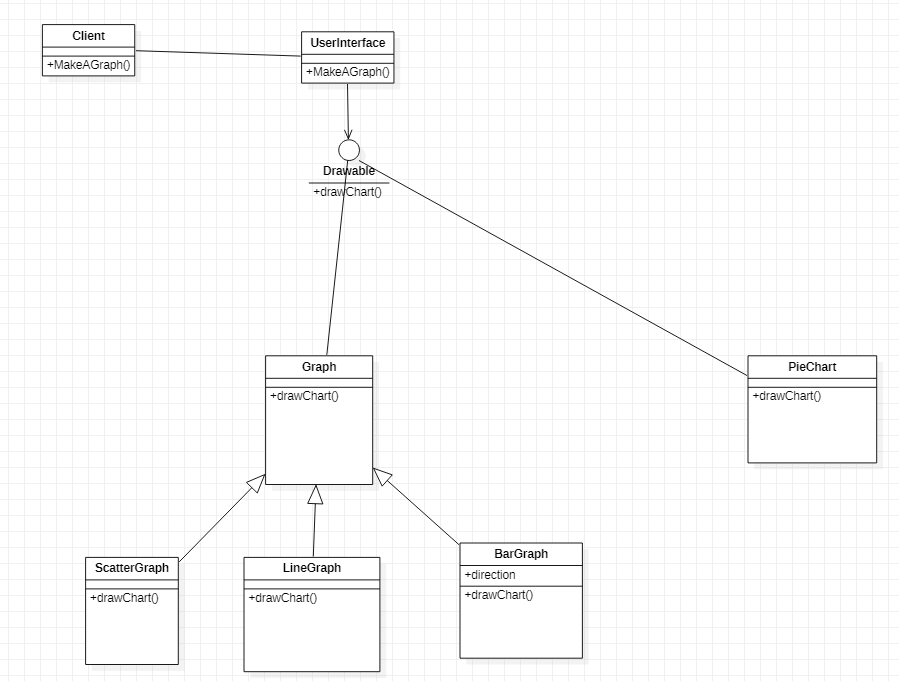


6. 설명 받은 설계에 대하여 아직 구현하지 않았다 그런데 고객이 꺽은선 그래프를 추가해 주길 원하고 있다 하자. 이런 변경이 설계에 어떤 영향을 줄 것인지 기술하라. 설명에 도움이 된다면 다이어그램을 그려도 좋다.

꺾은선그래프는 Scatter나 Bar와 같이 x축, y축이 존재할것이고 데이터 포인트 또한 존재할 것이다. Graph클래스를 상속받아서 만든다면 큰 변경은 없을 것이라 예상된다. 하지만 이는 꺾은선 그래프만을 추가할 때만 해당된다. x축,y축이 존재하지 않는 그래프를 그릴땐(위의 파이차트) Graph클래스를 상속받아선 안된다.



5번을 고려해서 Drawable인터페이스가 추가된 설계에 LineGraph클래스를 추가한다면 다음과 같을 것이다.



7. 설명 받은 설계에 대하여 아직 구현하지 않았다고 가정한다. 고객이 이번엔 GUI를 추가하길 원한다고 하자. 예를 들어 그래프의 색상, 데이터 포인트의 그래픽 표현을 바꾸고 싶어한다. 이런 변경이 설계에 어떻게 영향을 주는지 기술하라. 필요하면 다이어그램을 그려 설명해도 좋다.

현재의 설계에 색상, 그래픽표현에 대한 내용은 담겨있지 않다. Graph 클래스에 Graphic인터페이스를 aggregation하면 그래픽 표현을 바꿀 수 있을 것이라 생각된다. 하지만 Graph의 틀(x축, y축, xLabel, yLabel, 제목 등)의 색깔과 그래프 안의 데이터 포인트의 그래픽 표현은 분리되어야 한다. 따라서 ScatterGraph, BarGraph의 책임을 분산하기 위해 산포도의 점을 담당하는 Scatter클래스와 막대를 담당하는 Bar클래스를 생성하자. 5번과 6번을 고려해서 파이차트와 꺾은선그래프의 경우도 마찬가지로 Line, Pie클래스를 생성하자. Scatter, Bar, Line, Pie 클래스에 Graphic 인터페이스를 aggregation해준다면 해당 변경을 커버할 수 있을 것이다. 현재 설계에는 그래픽 표현을 담당하는 부분이 없어 많은 영향을 줄 것이다.

