|  |  |
| --- | --- |
|  |  |
| 자료구조와C++프로그래밍 |  |
|  |  |
|  | **이름 : 장혁수****학과 : 소프트웨어학과****학번 : 2019125055****수업시간 : 월 10:30, 수09:00** |
|  | **과제번호 : 과제 1** |

### 문제 정의

먼저 1-1 문제는 먼저 코드를 자세히 보고 이후 코드의 각 부분에 주석을 작성하고 이후 실행 결과물이 어떤 단계에서 출력 된 것인지 설명하는 문제이다.

이에 대한 해결방법은 먼저 어떠한 클래스가 있고, 그 클래스에 어떠한 데이터 멤버, 멤버 함수가 있는지 자세히 살피는 것이겠다. 그리고 또한 main함수에서 어떤 변수, 객체, 함수를 호출하고 활용하는지 잘 보면 되겠다.

다음 1-2문제를 크게 쪼개 보면 4가지 문제가 나온다고 할 수 있다.

1. 입력연산자 >>에 대한 오버로딩을 구현해라.
2. 이렇게 만든 입력연산자 >>를 이용해 Rectangle 클래스의 r1, r2객체를 생성하고 x, y, width,

height의 값을 런타임에 직접 입력 받도록 해라.

1. 객체의 넓이를 구하는 함수 등을 클래스 내부에 추가하고 구현해라.
2. 구현한 함수를 이용해 r1, r2객체가 겹치는 영역을 나타내는 새로운 r3객체를 생성하고 그 내용을 출력해라.

그리고 간단하게 위 4가지 1-2문제를 해결하기 위한 방법론을 생각해보면 각각 다음과 같다.

1. 연산자 <<은 ostream과 연관 있지만, 입력연산자 >>은 istream과 연관 되어있으니 주어진 코드의 <<에 대한 오버로딩에서 ostream을 istream으로 바꾸면서 시작하자.
2. 먼저 Rectangle생성자의 default value를 넣어서 혹시 발생할 수 있는 오류(초기화하지 않았는데도 데이터멤버에 접근하는 경우)를 방지하고 이후 만든 입력연산자 >>를 통해 값을 입력할 때 입력할 값(x, y, width, height)이 무엇인지 알 수 있도록 차례차례 무엇을 입력하라고 출력하면서 입력 받자.
3. 객체의 넓이를 구하는 함수는 객체의 width값과 height값을 곱한 값을 반환해 주게 만들고 또한 연산자 ==의 다중화를 이용해서 두 객체가 겹치는 영역을 나타내는 새로운 객체를 반환하는 함수를 만들자.
4. 위에 새롭게 만든 연산자 ==을 이용해서 r3에 두 객체가 겹치는 영역을 저장하고, 그 내용을 출력하는데 이 때 객체의 x, y, width, height, 넓이와 같은 영역을 표현해주는 값들을 출력해 주는 새로운 함수를 또한 만들어서 나타내자

### 문제 해결

첫 번째로 문제 1-1에 따라 주어진 코드의 각 부분에 주석을 넣은 코드를 보여주자면 다음과 같다.

○Rectangel.h

텍스트이(가) 표시된 사진

자동 생성된 설명

○main.cpp

텍스트, 화면이(가) 표시된 사진

자동 생성된 설명

○Rectangle.cpp

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

이후 실행결과에 대한 해석은 다음과 같다

○실행결과

텍스트이(가) 표시된 사진

자동 생성된 설명

먼저 main함수에서 r1이라는 Rectangle 객체를 정적할당 했고, 그 과정에서 r1 객체의 데이터 멤버 값인 xLow, yLow, width, height 값을 각각 1,1,3,4로 초기화해줬고 이후 Rectangle의 정적 데이터 멤버인 id값의 +1 된 값이 출력되면서 1번째 Rectangle 객체가 생성되었다고 출력됐다. 이후, 출력 연산자 <<의 다중화를 사용해서 r1 객체의 데이터 멤버 값들을 각각 출력됐다.

이후 새로운 Rectangle 객체가 new 연산자를 통해 동적할당 되었고, 그 과정에서 앞과 비슷하게 데이터 멤버 값이 2,3,5,5로 초기화 되었고, id값이 2가 됨으로써 2번째 Rectangle 객체가 생성되었다고 출력됐다. 그리고 포인터 변수 r2를 만들고 이 객체를 가리키게 했다. 이어서 r2가 가리키는 객체의 데이터 멤버 값을 각각 출력했다.

그리고 연산자 + 다중화를 r1 객체와 r2가 가리키는 객체 사이에 실행했고, 이에 r1, \*r2 객체의 xLow, yLow, width, height 값 끼리 서로 더한 값으로 새로운 Rectangle 객체가 초기화 되었고, id값이 3이 돼 3번째 Rectangle 객체가 생성되었다고 출력됐다. 하지만 그 이후 r1 객체에 데이터 맴버 값은 복사하고 소멸되고 그 과정에서 파괴자가 실행되어서 좌표 (3, 4) Rectangle 객체 소멸이라고 출력된 것이다.

이어서 새롭게 데이터 맴버 값이 바뀐 r1 객체의 값들을 각각 출력했다.

마지막으로 delete 연산자를 사용해 r2가 가리키는 동적할당된 객체의 메모리를 해제 시켜줌으로 파괴자가 실행되어서 좌표 (2,3) Retangle 객체 소멸이라고 출력되었고, 마지막에 main함수가 끝나고 정적할당된 r1객체가 소멸되면서 파괴자가 실행되어서 좌표 (3,4) Rectangle 객체 소멸이라고 출력되었다.

다음 문제 1-2의 주요 문제들에 대해서 작성한 코드 및 프로그램 기법이나 알고리즘을 설명하겠다.

★입력연산자 >>에 대한 오버로딩 구현하고 이를 이용해서 r1, r2 객체의xLow, yLow, width, height값 입력 받기 및 생성자 default value 지정



먼저 위와 같이 Rectangle 클래스 안에 입력연산자 >>에 대한 오버로딩을 선언해준다.

이때, 출력연산자 <<에 대한 오버로딩에서 ostream을 istream으로 바꾸었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이후 Rectangle.cpp에서 입력연산자 >>에 대한 오버로딩을 구현해보면 위와 같다.

여기서 중요한 것은 먼저 매개변수로 입력 스트림과 Rectangle자료형의 레퍼런스를 가져오고 입력 스트림 매개변수 in을 통해 참조하는 객체의 height, width, xLow, yLow값을 차례차례 입력 받게 한 것이다.

텍스트, 시계, 설정이(가) 표시된 사진

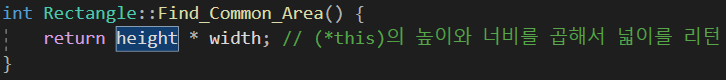
자동 생성된 설명

그리고 위에서 보다시피 메인 함수에서 r1, r2 객체를 만들고 방금 만든 입력연산자 >>에 대한 오버로딩을 이용해서 력 받으면 된다.



여기서 또한 구현한 것은 위와 같이 입력연산자 >>을 이용하지 않고 초기화되지 않은 데이터 멤버에 접근해서 나오는 오류를 생각해 생성자를 통해 default value를 0으로 지정해준 것이다.

★Rectangle 객체 영역의 넓이를 반환, 연산자 ==의 다중화를 이용해서 두 객체가 겹치는 영역을 나타내는 새로운 객체를 반환하는 함수 구현



일단 객체 영역의 넓이를 반환하는 함수는 위와 같이 객체의 높이와 너비를 가지고 있는 데이터멤버 height와 width를 곱하고 이를 반환하면 된다.

그리고 이제 가장 중요한 것이 연산자 ==의 다중화를 이용해서 두 객체가 겹치는 영역을 나타내는 새로운 객체를 반환하는 함수를 구현하는 것이다.

일단 먼저 생각해 볼 수 있는 게 각 객체가 나타내는 영역은 직사각형이다. 이에 만약 두 객체가 겹치는 영익이 존재한다면 그것은 또한 직사각형일 거고 이를 Rectangle 객체로 반환하면 될 것이다.

이후 공통된 영역을 생각해 볼 때 한번에 직사각형 전체를 보지 말고 width, height 따로따로 보도록 하는데 그 중 width(너비)만 먼저 보자. 각각 객체의 width의 범위를 보면 x좌표 범위 [xLow, xLow+width]로 볼 수 있다. 즉, 공통된 영역의 width는 각 객체의 x좌표 범위 [xLow, xLow+width]의 교집합인 것을 알 수 있다. 그리고 이 교집합은 세 가지 경우로 나뉠 수 있다.

1. 교집합이 없는 경우, 즉 겹치는 x좌표 범위가 없는 경우
2. 교집합이 있는데 겹치는 x좌표 범위가 두 객체의 width보다는 작은 경우
3. 교집합이 있는데 겹치는 x좌표 범위가 한 객체의 width와같은 경우

이를 그림을 표현하면 다음과 같다. 선은 각 객체의 밑면을 의미하고 왼쪽부터 대표적인1 2 3 경우를 나타낸다.

이때 노란색 칠한 부분이 겹치는 x좌표의 범위를 나타낸다 여기서 선들의 y좌표는 생각하지 않고 보자

경우1 경우2 경우 3



이렇게 보면 공통점이 하나 나온다. 즉, 공통된 영역의 xLow의 값은 두 객체 중 더 큰 xLow의 값이라는 것이다. 다음에 각 경우에서 공통된 width값을 살펴보면 다음과 같다.

2 경우 : 두 객체 중 더 작은 xLow를 가지고 있는 객체의 xLow값과 width값을 더하고 이를 더 큰 xLow를 가지고 있는 객체의 xLow값을 뺀 것과 같다.

1경우 : 2경우로 구한 공통된 width값이 0보다 작거나 같게 된다.

3경우 : 두 객체 중 더 작은 xLow를 가지고 있는 객체의 xLow값과 width 값을 더한 게 더 큰 xLow를 가지고 있는 객체의 xLow값과 width값을 더한 것보다 크다. 결국 공통된 width값은 더 큰 xLow를 가지고 있는 객체의 width값이다.

공통된 height를 구하는 경우도 잘 생각해보면 결국 공통된 width를 구하는 경우와 똑같다.

그리고 생각해볼 수 있는게, 공통된 width, height를 구할 때 둘 중 하나라도 1경우를 만족한다면 공통된 영역이 없다는 것이다. 즉, 2경우로 구한 공통된 width, height 값 둘 중 하나는 0 보다 작거나 같은 경우이다.

이후 Rectangle.cpp에서 위의 알고리즘을 토대로 연산자 == 의 다중화를 구현해보자면 다음과 같다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트, 벽, 모니터, 화면이(가) 표시된 사진

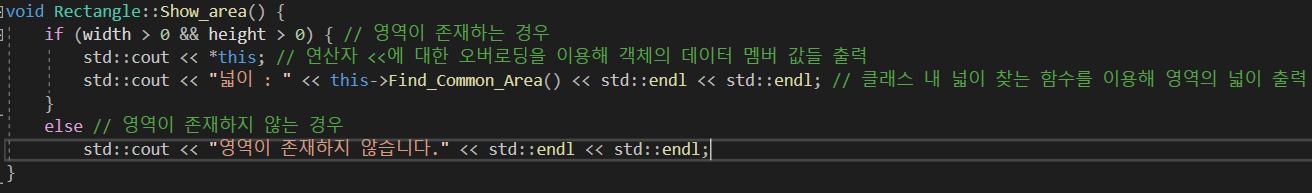
자동 생성된 설명

★ 겹치는 영역을 나타내는 r3 객체를 만들고 그 영역을 출력해라

공통된 영역은 위에서 연산자 == 다중화를 이용해서 구하도록 하고, 이때 Rectangle 객체의 영역을 출력해주는 함수 Show\_area()를 Rectangle 클래스 내에 하나 더 구현하고 Show\_area함수를 이용할 때 영역의 넓이를 반환하는 함수도 같이 이용해주자.



추가로 아까 공통된 width 또는 height가 존재하지 않는 경우에는 공통된 영역이 없으므로 그런 경우에는 “영역이 존재하지 않습니다”라고 출력해주자



이때 주어진 코드의 출력연산자 <<에 대한 오버로딩을 통해 다음과 같이 height, width, xLow, yLow의 값을 출력해준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

위의 코드들을 main에서 활용해 본다면 다음과 같다.

텍스트, 시계, 탑재, 설정이(가) 표시된 사진

자동 생성된 설명

마지막으로 아래 표와 같이 각 객체의 값을 입력할 시 출력 결과는 다음과 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | height | width | xLow | yLow |
| r1 | 4 | 5 | 1 | 3 |
| r2 | 3 | 6 | 4 | 1 |

텍스트이(가) 표시된 사진

자동 생성된 설명

### 결론

2차원에서 어떤 직사각형의 영역을 표현하라고 한다면, 다양한 방법이 있겠지만, 이 문제와 같이 너비, 높이, 또 기준이 되는 한 점이 기본적인 방법인 것 같다. 그래서 두 직사각형의 공통된 영역이 또한 직사각형이라는 것을 알고 이 영역 또한 이와 같은 방법으로 표현하자는 생각이 들었다. 그리고 나아가 간단하지만 넓이 또한 표현함으로 마침표를 찍은 것 같다. 하지만 이것은 영역이 직사각형이기에 가능한 것이지 만약 삼각형이나 다른 도형들이었다면 다른 방법을 생각해야 할 것 같다.

과제를 수행하면서 연산자의 다중화에 대해서 좋은 문법이라는 생각이 들었다. 특히나 연산자 <<, >>에 대한 오버로딩을 구현하는 과정만 좀 힘들지 이후 매우 쉽게 이용한 것 같다.

아마 전에 C언어를 배울 때 출력하거나 입력 받을 때 거의 printf , scanf함수를 썼었는데 그 때마다 쓸 것도 많고, 문법도 복잡해서 그런 것 같다.

그리고 특히 어려웠던 점은 역시 클래스라는 새로운 개념에 대해서 활용하는 점이 어려웠다.

그래도 struct와 유사한 점이 어느정도 있어서 그렇게 까지 크게 힘들지는 않았고, 지금까지 배운 내용을 정리하고 복습한 좋은 시간이었던 것 같다.