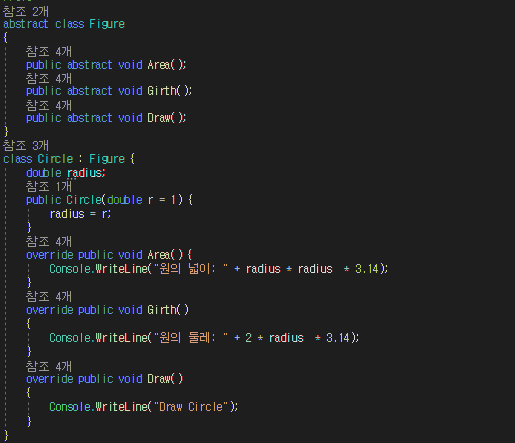
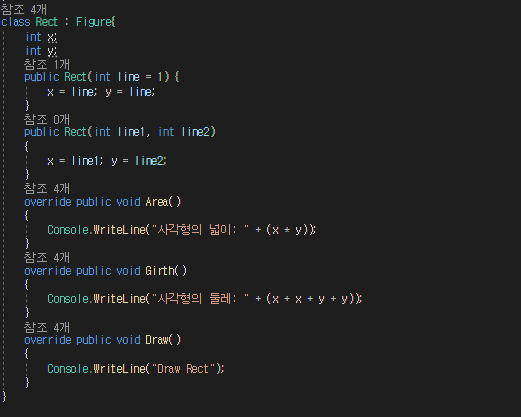
REPORT

(닷넷 2차 과제)

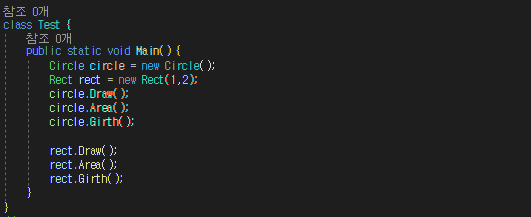


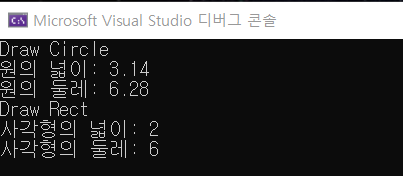
|  |  |
| --- | --- |
| 제목 | 연습문제5.8, 9, 11 |
| 제출일자 | 20.11.20 |
| 소속학과 | 컴퓨터공학과 |
| 학번 | 2017305039 |
| 성명 | 신동민 |

문제) 5.8



0



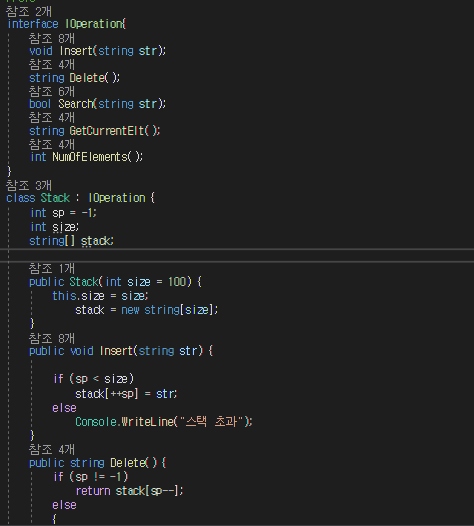
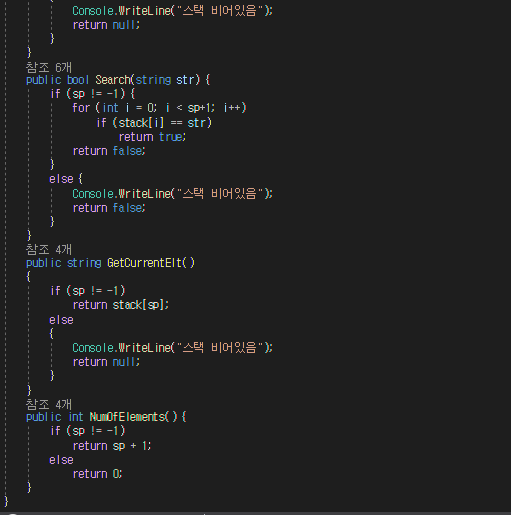


설명: 추상 클래스 Figure을 만들어주고 상속받을 원 클래스Circle와 사각형 클래스Rect에 각 메소드들을 override로 선언합니다.

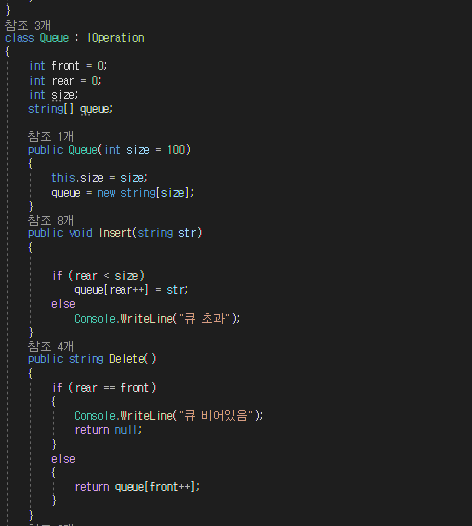
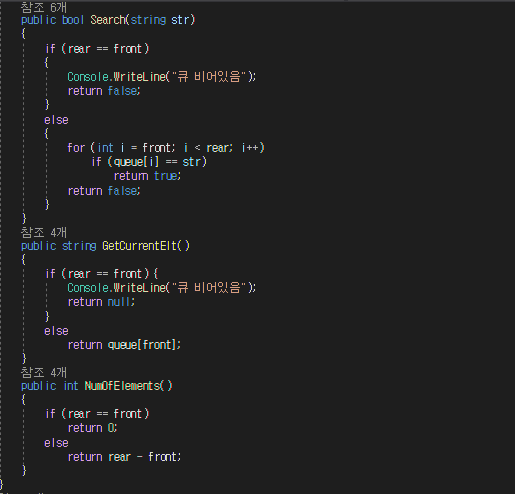
**메소드 만들어줄 때 new와 override를 선택해야 하는데 예를 들면 base 객체 = new derived();가 있다면 객체의 메소드를 실행시킬 때 new를 사용했다면 base클래스의 메소드를 override를 사용했다면 derived클래스의 메소드를 선택합니다.**

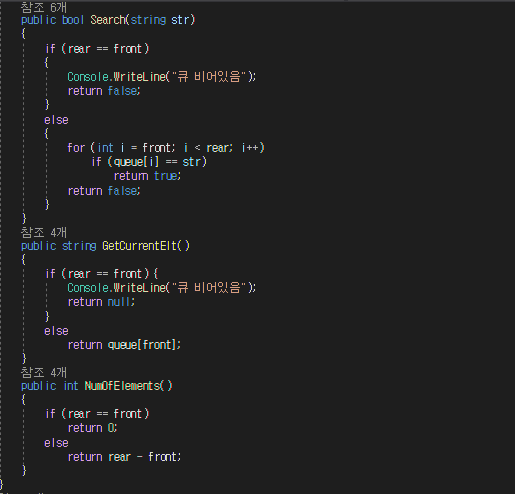
이 과정을 테스트하기 위해서 Test클래스를 만들어 메인문에서 각각의 객체를 만들어줍니다. 이때 원의 객체는 매개변수로 디폴트 값인 1이 반지름이 되고 사각형은 입력 받은 매개변수 1,2로 직사각형이 만들어집니다. 객체의 메소드를 실행하면 반지름이 1인 원의 넓이: 1x1x3.14, 원의 둘레: 2x1x3.14이며 1,2인 직사각형의 넓이: 1x2, 직사각형의 넓이: 1+1+2+2로 결과 화면이 출력됩니다.

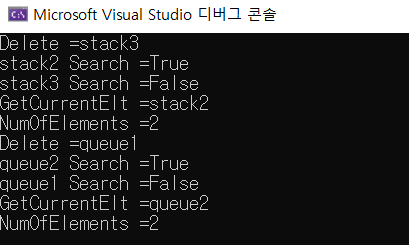
문제) 5.9



문제) 4.14







설명: 인터페이스 IOperation을 상속받는 Stack클래스와 Queue클래스을 만들어 줍니다. 만들어진 클래스가 실행되는지 알기 위해서 Test클래스에 메인문을 만들어서 Stack클래스의 객체 sk와 Queue클래스의 객체 qe의 매개변수 값을 디폴트로 받아 사이즈가 100인 배열이 선언됩니다. 먼저 sk객체에 Insert(삽입)메소드로 배열에 stack1, stack2, stack3 문자열이 들어가며 sp(스택포인터)는 마지막에 들어간 데이터의 위치 인덱스로 바뀝니다. 그리고 Delete(삭제)메소드를 실행하면 맨 뒤에 있던 stack3데이터가 삭제되며 sp의 위치 값이 바뀝니다. 스택 안에 값을 탐색하는 Search(탐색)메소드로 원하는 데이터를 찾는데 stack2데이터는 있어서 True로 나오지만 stack3데이터는 삭제해서 False로 없다고 실행되며 GetCurrentElt()메소드는 현재 sp가 지정하는 위치에 데이터 값을 알려줍니다. 마지막으로 NumOfElements()메소드는 스택에 들어가 있는 데이터의 총수를 알려줍니다. 만약 처음 정해진 사이즈가 넘어가면 ‘스택 초과’가 뜨며 배열에 데이터가 없다면 ‘스택 비어있음’이라는 문구가 출력됩니다.

**Insert 실행**

Sp=0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stack1 |  |  |  |  |

Sp=1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stack1 | Stack2 |  |  |  |

Sp=2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stack1 | Stack2 | Stack3 |  |  |

**delete 실행**

sp=1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stack1 | Stack2 |  |  |  |

다음으로 qe객체에 Insert(삽입)메소드로 배열에 queue1, queue2, queue3 문자열이 들어가며 rear(입력큐포인터)는 마지막에 들어간 데이터의 위치 인덱스로 바뀝니다. 그리고 Delete(삭제)메소드를 실행하면 맨 앞에 있던 queue1데이터가 삭제되며 front(출력큐포인터)의 위치 값이 바뀝니다. 큐 안에 값을 탐색하는 Search(탐색)메소드로 원하는 데이터를 찾는데 queue2데이터는 있어서 True로 나오지만 queue1데이터는 삭제해서 False로 없다고 실행되며 GetCurrentElt()메소드는 현재 front가 지정하는 위치에 데이터 값을 알려줍니다. 마지막으로 NumOfElements()메소드는 큐에 들어가 있는 데이터의 총수를 알려줍니다. 만약 처음 정해진 사이즈가 넘어가면 ‘큐 초과’가 뜨며 배열에 데이터가 없다면 ‘큐 비어있음’이라는 문구가 출력됩니다.

**Insert 실행**

Front=rear=0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Queue1 |  |  |  |  |

Front=0 rear=1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Queue1 | Queue2 |  |  |  |

Front=0 rear=2

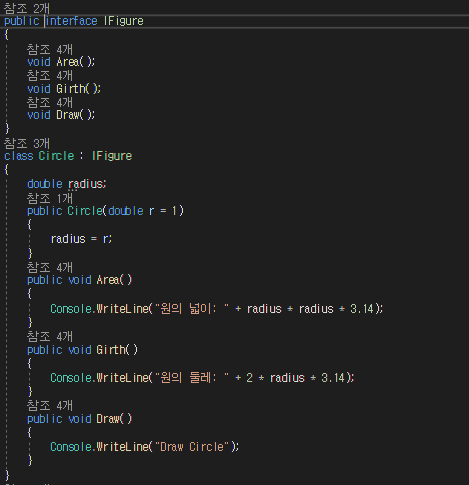
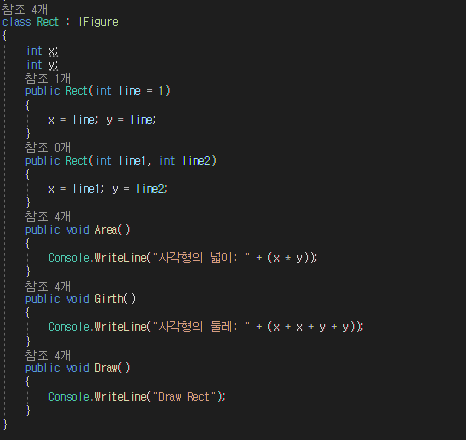
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Queue1 | Queue2 | Queue3 |  |  |

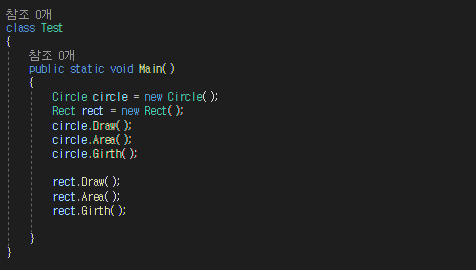
**delete 실행**

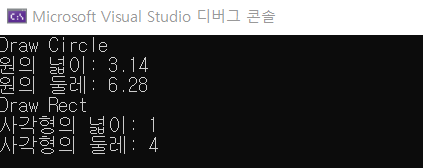
Front=1 rear=2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Queue2 | Queue3 |  |  |

문제) 5.11







설명: 인터페이스 Figure을 만들어주고 상속받을 원 클래스Circle와 사각형 클래스Rect에 각 메소드들을 선언합니다. 이 과정을 테스트하기 위해서 Test클래스를 만들어 메인문에서 각각의 객체를 만들어줍니다. 이때 원의 객체는 매개변수로 디폴트 값인 1이 반지름이 되고 또한, 사각형도 매개변수로 디폴트 값인 1을 받아 정사각형이 만들어집니다. 객체의 메소드를 실행하면 반지름이 1인 원의 넓이: 1x1x3.14, 원의 둘레: 2x1x3.14이며 1,1인 정사각형의 넓이: 1x1, 직사각형의 넓이: 1+1+1+1로 결과 화면이 출력됩니다.