

닷넷프로그래밍

과제2

2019305059

이현수

```

1  using System;
   참조 2개
2  abstract class Figure
3  {
   참조 4개
4      public abstract void Area();
   참조 4개
5      public abstract void Girth();
   참조 4개
6      public abstract void Draw();
7  }
   참조 3개
8  class Rect : Figure
9  {
10     int width, height;
   참조 1개
11     public Rect(int width, int height)
12     {
13         this.width = width;
14         this.height = height;
15     }
   참조 4개
16     override public void Area()
17     {
18         Console.WriteLine("사각형의 넓이는 {0}입니다.", width * height);
19     }
   참조 4개
20     override public void Girth()
21     {
22         Console.WriteLine("사각형의 둘레는 {0}입니다.", 2 * width + 2 * height);
23     }
   참조 4개
24     override public void Draw()
25     {
26         Console.WriteLine("사각형을 그립니다.");
27         for (int i = 0; i < width; i++) Console.Write("*"); Console.WriteLine();
28         for (int i = 0; i < height-2; i++)
29         {
30             Console.Write("*");
31             for (int j = 0; j < width - 2; j++) Console.Write(" ");
32             Console.WriteLine("*");
33         }
34         for (int i = 0; i < width; i++) Console.Write("*"); Console.WriteLine();
35     }
36 }
   참조 3개
37 class Circle : Figure
38 {
39     int radius;
   참조 1개
40     public Circle(int radius)
41     {
42         this.radius = radius;
43     }
   참조 4개
44     override public void Area()
45     {
46         Console.WriteLine("원의 넓이는 {0:F2}입니다.", Math.PI * Math.Pow(radius, 2));
47     }
   참조 4개
48     override public void Girth()
49     {
50         Console.WriteLine("원의 둘레는 {0:F2}입니다.", 2 * Math.PI * radius);
51     }
   참조 4개
52     override public void Draw()
53     {
54         Console.WriteLine("원을 그렸습니다.");
55     }
56 }

```

```

참조 0개
57 class Ch8_8
58 {
    참조 0개
59     public static void Main()
60     {
61         Rect r = new Rect(5, 3);
62         Circle c = new Circle(10);
63
64         r.Area(); r.Girth(); r.Draw();
65
66         Console.WriteLine();
67
68         c.Area(); c.Girth(); c.Draw();
69     }
70 }

```

(실행결과)

```

Microsoft Visual Studio 디버그 콘솔
사각형의 넓이는 15입니다.
사각형의 둘레는 16입니다.
사각형을 그립니다.
*****
*   *
*****

원의 넓이는 314.16입니다.
원의 둘레는 62.83입니다.
원을 그렸습니다.

C:\Users\nec\Desktop\닷넷프로그래밍_중간고사이후\닷넷프로그래밍_중간고사이후\bin\Debug\netcoreapp3.1\닷넷프로그래밍_중간고사이후.exe(프로세스 6468개)이(가)
종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요.

```

(설명)

추상클래스 Figure이 있다. 여기에는 추상메소드 Area, Girth, Draw가 존재한다.

Figure 클래스를 상속받는 Rect클래스를 선언한다. 필드 width, height을 선언한다. 각각 사각형의 가로, 세로 길이를 의미한다.

(코드11-15)Rect클래스 생성자를 만들어 준다. 매개변수 2개를 지정어 this를 이용해 Rect클래스 필드 두개를 각각 초기화 해준다.

*추상클래스의 abstract 수정자는 virtual 수정자의 의미가 포함되어 있어서 추상클래스를 상속받는 파생클래스는 override 수정자를 사용해 추상 메소드를 재정의한다.

(코드16-19)사각형넓이를 구하는 메소드이다. 넓이=가로*세로 이므로 width*height로 계산해 출력한다.

(코드20-23)사각형둘레길이를 구하는 메소드이다. 2*width + 2*height로 계산해 결과를 출력한다.

(코드24-35)사각형을 그리는 메소드이다. 직접 그릴 수 없어서 * 모양으로 그린다. 처음에 for반복문으로 width만큼 *을 출력하고 height-2만큼 *출력+width-2만큼의공백+*출력 후 마지막에 for반복문으로 width개 *을 출력한다.

(코드40-43)Circle 클래스 생성자를 만들었다. 매개변수 radius에서 값을 받아 Circle 클래스 필드 radius를 초기화한다. 이때 radius 이름이 같으므로 this 지정어를 이용해 this.radius = radius; 값을 초기화한다.

(코드44-47)원넓이를 구하는 메소드이다. 넓이=PI* radius² 이다. PI값은 Math.PI를 이용하고 제곱은 Math.Pow(radius,2)를 이용해 계산한다. 원 넓이를 출력할 때 소수점 2번째까지 출력하게 만들었다.

(코드48-51)원둘레길이를 구하는 메소드이다. 둘레길이는 2*PI*radius이다. PI값은 Math.PI를 이용한다. 원 둘레를 출력할 때 소수점 두번째까지 출력하게 만들었다.

(코드52-55)원을 그리는 메소드이다. 직접 그릴 수 없어서 글자로 원을 그리는 메소드임을 나타냈다.

(코드59~) 클래스 ch8_8내 Main메소드에서 width=5, height=3인 Rect클래스 객체 r과 radius=10인 Circle클래스 객체 c를 만들어준다. 그리고 r.Area(), r.Girth(), r.Draw(), c.Area(), c.Girth(), c.Draw()를 호출한다.

(코드캡처)

```

1  using System;
2  interface IOperation
3  {
4      void Insert(string str);
5      string Delete();
6      bool Search(string str);
7      string GetCurrentElt();
8      int NumOfElements();
9  }
10 class Stack : IOperation
11 {
12     string[] stack;
13     int sp = -1;
14     int size = 0;
15     public Stack(int size = 20)
16     {
17         stack = new string[size];
18     }
19     public void Insert(string str)
20     {
21         stack[++sp] = str;
22         size++;
23     }
24     public string Delete()
25     {
26         size--;
27         return stack[sp--];
28     }
29     public bool Search(string str)
30     {
31         for(int i = 0; i < size; i++)
32         {
33             if (stack[i] == str) return true;
34         }
35         return false;
36     }
37     public string GetCurrentElt()
38     {
39         return stack[sp];
40     }
41     public int NumOfElements()
42     {
43         return size;
44     }
45     public void printStack()
46     {
47         for(int i = 0; i < size; i++)
48         {
49             Console.Write("{0} ", stack[i]);
50         }
51         Console.WriteLine();
52     }
53 }
54 class Queue : IOperation
55 {
56     string[] queue;
57     int front = 0;
58     int rear = -1;
59     int size = 0;
60     public Queue(int size)
61     {
62         queue = new string[size];
63     }
64     public void Insert(string str)
65     {
66         queue[++rear] = str;
67         size++;
68     }

```

```

69     public string Delete()
70     {
71         size--;
72         return queue[front++];
73     }
74     public bool Search(string str)
75     {
76         for (int i = front; i <= rear; i++)
77         {
78             if (queue[i] == str) return true;
79         }
80         return false;
81     }
82     public string GetCurrentElt()
83     {
84         return queue[front];
85     }
86     public int NumOfElements()
87     {
88         return size;
89     }
90     public void printQueue()
91     {
92         for(int i = front; i <= rear; i++)
93         {
94             Console.Write("{0} ", queue[i]);
95         }
96         Console.WriteLine();
97     }
98 }
99 class Exercise5_9
100 {
101     public static void Main()
102     {
103         Console.WriteLine("[스택 테스트]");
104         Stack s = new Stack(10);
105         s.Insert("a"); s.Insert("b"); s.Insert("c"); s.Insert("d"); s.Insert("e"); s.Insert("f");
106         s.printStack();
107         Console.WriteLine("스택 원소 삭제 : {0}", s.Delete());
108         s.printStack();
109         if (s.Search("b")) Console.WriteLine("스택에 원소 'b'가 존재 합니다.");
110         if (!s.Search("z")) Console.WriteLine("스택에 원소 'z'가 존재하지 않습니다.");
111         Console.WriteLine("스택 탑에 있는 원소는 {0}", s.GetCurrentElt());
112         Console.WriteLine("스택에 존재하는 원소 개수 {0}개\n", s.NumOfElements());
113
114         Console.WriteLine("[큐 테스트]");
115         Queue q = new Queue(10);
116         q.Insert("a"); q.Insert("b"); q.Insert("c"); q.Insert("d"); q.Insert("e"); q.Insert("f");
117         q.printQueue();
118         Console.WriteLine("큐 원소 삭제 : {0}", q.Delete());
119         q.printQueue();
120         if (q.Search("b")) Console.WriteLine("큐에 원소 'b'가 존재 합니다.");
121         if (!q.Search("z")) Console.WriteLine("큐에 원소 'z'가 존재하지 않습니다.");
122         Console.WriteLine("큐 front에 있는 원소는 {0}", q.GetCurrentElt());
123         Console.WriteLine("큐에 존재하는 원소 개수 {0}개", q.NumOfElements());
124     }
125 }

```

(실행캡처)

```

Microsoft Visual Studio 디버그 콘솔

[스택 테스트]
a b c d e f
스택 원소 삭제 : f
a b c d e
스택에 원소 "b"가 존재 합니다.
스택에 원소 "z"가 존재하지 않습니다.
스택 탑에 있는 원소는 e
스택에 존재하는 원소 개수 5개

[큐 테스트]
a b c d e f
큐 원소 삭제 : a
b c d e f
큐에 원소 "b"가 존재 합니다.
큐에 원소 "z"가 존재하지 않습니다.
큐 front에 있는 원소는 b
큐에 존재하는 원소 개수 5개

```

(설명)

(코드2-9) 인터페이스 IOperation이 구현되었다. 총 5개 메소드가 있다.

클래스 Stack을 선언하고 인터페이스 IOperation을 상속받는다. stack[] stack;을 선언해준다. 이 클래스의 경우 다루는 자료형은 string이다. 그리고 스택포인트를 나타내는 sp=-1로 초기화 해주고 스택에 들어있는 원소개수를 나타내는 size변수를 0으로 초기화한다.

(코드15-18) 매개변수 size를 가지는 생성자를 만든다. size크기의 string형 배열을 만들어준다.

(코드19-23) public으로 Insert메소드를 구현한다. 매개변수 string형 str을 스택에 삽입한다. 우선 전위연산자 ++sp를 통해 sp를 1증가시키고 str을 배열에 저장한다. 그리고 size++을 해줘 크기를 한 개 증가시킨다.

(코드24-28) public으로 Delete메소드를 구현한다. 우선 size를 한 개 줄이고 배열 인덱스 sp에 있는 원소를 반환하고 sp를 1 감소시킨다.

(코드29-36) public으로 Search메소드를 구현한다. 매개변수 str이 스택배열에 있는지를 체크해 있으면 true를, 없으면 false로 반환한다. for반복문으로 0부터 size-1까지 반복하면서 if조건문으로 검사한다.

(코드37-40) public으로 GetCurrentElt메소드를 구현한다. 스택의 탑에 있는 원소를 반환한다. 스택배열 stack의 sp 인덱스 원소를 반환한다.

(코드41-44)public으로 NumOfElements메소드를 구현한다.스택에 들어있는 원소개수를 반환한다. Size를 반환한다.

(코드45-52) printStack()메소드를 public으로 구현한다. 스택에 들어있는 원소를 출력한다. for반복문으로 0부터 size-1까지 출력해준다.

클래스 Queue를 선언하고 인터페이스 IOperation을 상속받는다. 큐는 배열로 구현한다. String[] queue;를 선언하고 front, rear를 각각 0,-1로 초기화해주고 size 원소개수는 0으로 초기화한다.

(코드60-63) 매개변수 size를 가지는 생성자를 만든다. size크기의 큐 배열을 만든다.

(코드64-68) public으로 Insert메소드를 만든다. 매개변수 str을 큐에 삽입한다. 우선 ++rear 전위연산자를 통해 rear를 1 증가시키고 그 인덱스에 str을 저장하고 size변수를 1 증가시킨다.

(코드69-73) public으로 Delete메소드를 구현한다. 우선 size변수를 1 감소시키고, front인덱스에 있는 원소를 반환하고 front를 1 증가시킨다.

(코드74-81) public으로 Search메소드를 구현한다. 매개변수 str이 큐 배열에 있는지를 검사해 있으면 true, 없으면 false를 반환한다. for반복문으로 front인덱스부터 rear인덱스까지 반복하고 if조건문으로 검사한다.

(코드82-85) public으로 GetCurrentElt메소드를 구현한다. 큐 배열 front인덱스 원소를 반환한다.

(코드86-89) public으로 NumOfElements메소드를 구현한다. 큐에 있는 원소개수를 반환한다. size를 반환한다.

(코드90-97) public으로 printQueue메소드를 구현한다. 큐에 있는 원소를 출력한다. for반복문을통해 front인덱스부터 rear인덱스까지 출력한다.

(코드101~) s.Insert("원소");를 통해 스택 배열에 a,b,c,d,e,f를 삽입한다. 그리고 출력한다. 그후 s.Delete();를 통해 스택배열의 f를 삭제하고, 다시 출력한다. s.Search("b"), !s.Search("z")코드를 if조건문에 넣어서 원소유무를 판단한다. "b"는 있기 때문에 있다고 출력되고, "z"는 없기 때문에 없다고 출력된다. 그리고 s.GetCurrentElt()를 통해 스택 탑에 있는 원소 e를 출력한다. s.NumOfElements()를 통해 스택배열에 존재하는 원소개수 5를 출력한다.

q.Insert("원소");를 통해 큐 배열에 a,b,c,d,e,f를 삽입한다. 그리고 출력한다. 그후 q.Delete();를 통해 큐배열의 a를 삭제하고, 다시 출력한다. q.Search("b"), !q.Search("z")코드를 if조건문에 넣어서 원소유무를 판단한다. "b"는 있기 때문에 있다고 출력되고, "z"는 없기 때문에 없다고 출력된다. 그리고 q.GetCurrentElt()를 통해 큐 front에 있는 원소 b를 출력한다. q.NumOfElements()를 통해 큐 배열에 존재하는 원소개수 5를 출력한다.

5-11

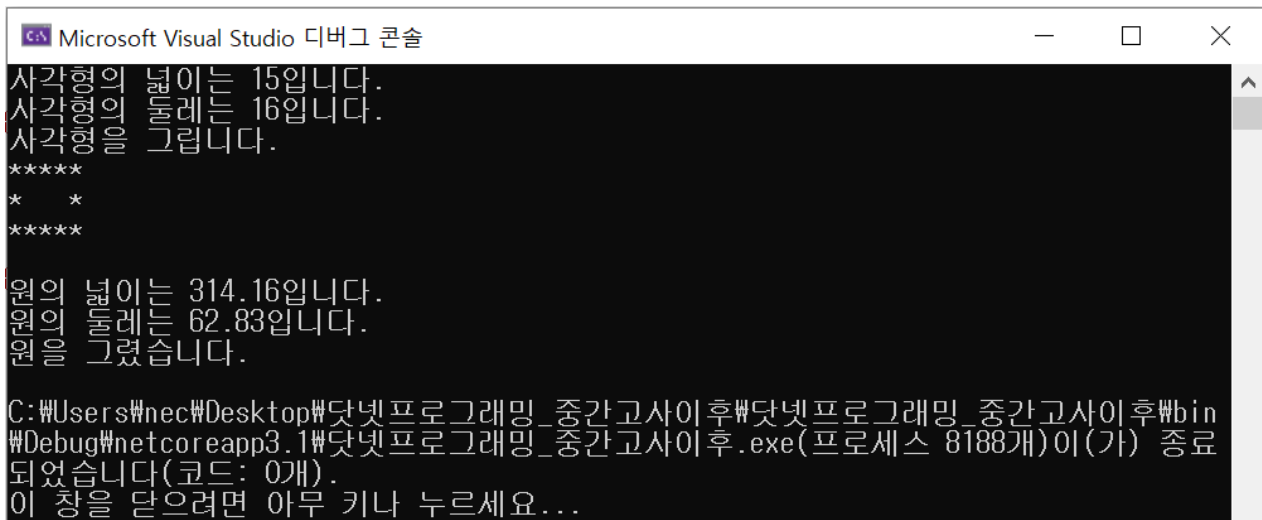
(코드캡처)

```

1  using System;
2  public interface IFigure
3  {
4      void Area();
5      void Girth();
6      void Draw();
7  }
8  class Rect : IFigure
9  {
10     int width, heighth;
11     public Rect(int width, int heighth)
12     {
13         this.width = width;
14         this.heighth = heighth;
15     }
16     public void Area()
17     {
18         Console.WriteLine("사각형의 넓이는 {0}입니다.", width * heighth);
19     }
20     public void Girth()
21     {
22         Console.WriteLine("사각형의 둘레는 {0}입니다.", 2 * width + 2 * heighth);
23     }
24     public void Draw()
25     {
26         Console.WriteLine("사각형을 그립니다.");
27         for (int i = 0; i < width; i++) Console.Write("*"); Console.WriteLine();
28         for (int i = 0; i < heighth - 2; i++)
29         {
30             Console.Write("*");
31             for (int j = 0; j < width - 2; j++) Console.Write(" ");
32             Console.WriteLine("*");
33         }
34         for (int i = 0; i < width; i++) Console.Write("*"); Console.WriteLine();
35     }
36 }
37 class Circle : IFigure
38 {
39     int radius;
40     public Circle(int radius)
41     {
42         this.radius = radius;
43     }
44     public void Area()
45     {
46         Console.WriteLine("원의 넓이는 {0:F2}입니다.", Math.PI * Math.Pow(radius, 2));
47     }
48     public void Girth()
49     {
50         Console.WriteLine("원의 둘레는 {0:F2}입니다.", 2 * Math.PI * radius);
51     }
52     public void Draw()
53     {
54         Console.WriteLine("원을 그렸습니다.");
55     }
56 }
57 class Exercise5_11
58 {
59     public static void Main()
60     {
61         Rect r = new Rect(5, 3);
62         Circle c = new Circle(10);
63
64         r.Area(); r.Girth(); r.Draw();
65         Console.WriteLine();
66         c.Area(); c.Girth(); c.Draw();
67     }
68 }

```

(실행캡처)



```
Microsoft Visual Studio 디버그 콘솔

사각형의 넓이는 15입니다.
사각형의 둘레는 16입니다.
사각형을 그립니다.
*****
*   *
*****

원의 넓이는 314.16입니다.
원의 둘레는 62.83입니다.
원을 그렸습니다.

C:\Users\wec\Desktop\닷넷프로그래밍_중간고사이_후\닷넷프로그래밍_중간고사이_후\bin
\Debug\netcoreapp3.1\닷넷프로그래밍_중간고사이_후.exe(프로세스 8188개)이(가) 종료
되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

(설명)

(코드2-7) 인터페이스 IFigure가 구현되었다. Void Area()는 넓이를 구하는 메소드, void Girth() 메소드는 둘레를 구하는 메소드, void Draw()은 도형을 구하는 메소드이다.

클래스 Rect를 선언하고 인터페이스 IFigure을 상속받는다. Rect 즉 사각형의 가로, 세로 길이를 나타내는 변수 width, height를 선언해준다.

(코드11-15) 매개변수 두개를 가진 생성자를 만든다. 매개변수와 클래스 필드 변수이름이 같으므로 지정어 this로 구분해준다.

(코드16-19) public 접근 수정자로 void Area()를 구현한다. 사각형 넓이는 가로*세로 이므로 width*height으로 구할 수 있다.

(코드20-23) public 접근 수정자로 void Girth()를 구현한다. 사각형 둘레는 2*width + 2*height으로 구할 수 있다.

(코드24-35) public 접근 수정자로 void Draw()를 구현한다. 직접 그릴 수 없어서 * 모양으로 그린다. 처음에 for반복문으로 width만큼 *을 출력하고 height-2만큼 *출력+width-2만큼의공백+*출력 후 마지막에 for반복문으로 width개 *을 출력한다.

클래스 Circle를 선언하고 인터페이스 IFigure을 상속받는다. Circle는 원이므로 반지름을 저장하기 위한 변수 radius를 선언한다.

(코드40-43) 매개변수 radius를 가지는 생성자함수를 만든다. 매개변수로 받은 값을 클래스 필드 radius에 저장한다. 매개변수와 클래스 필드가 같으므로 지정어 this를 이용한다.

(코드44-47) public 접근 수정자로 void Area()를 구현한다. 넓이= $\pi * radius^2$ 이다. π 값은 Math.PI를 이용하고 제곱은 Math.Pow(radius,2)를 이용해 계산한다. 원 넓이를 출력할 때 소수점 2번째까지 출력하게 만들었다.

(코드48-51) public 접근 수정자로 void Girth()를 구현한다. 원 둘레길이는 $2 * \pi * radius$ 이다. π 값은 Math.PI를 이용한다. 원 둘레를 출력할 때 소수점 두번째까지 출력하게 만들었다.

(코드52-55) public 접근 수정자로 void Draw()를 구현한다. 원을 그리는 메소드이다. 직접 그릴 수 없어서 글자로 원을 그리는 메소드임을 나타냈다.

(코드59~) 클래스 Rect의 객체 r를 생성자함수를 통해 width=5, height=3으로 만든다. 클래스 Circle의 객체 c를 생성자함수를 통해 radius=10으로 만든다. 그리고 r.Area(), r.Girth(), r.Draw(), c.Area(), c.Girth(), c.Draw()를 호출한다.