윈도우프로그래밍

6. 반복문(응용예제)

2023. 4. 7.

심미나

목 차

X. 응용 예제

참고자료: 수업교재(C# 프로그래밍)

X. 응용 예제

응용예제

응용 예제 4-1 (문자열 처리)

- 대문자화와 소문자화 - ToUpper(), ToLower()
  - 알파벳 문자열을 대문자 또는 소문자로 변경

메서드	설명
ToUpper()	문자열 내부의 문자를 모두 대문자로 변경합니다.
ToLower()	문자열 내부의 문자를 모두 소문자로 변경합니다.

- C#의 모든 문자열 처리 메서드는 자신을 변경하지 않고 반환
- 비파괴적 메서드: 자신을 변경하지 않고 반환하는 메서드

Input 문자열이 변경되는가?

```
static void Main(string[] args)
{
    string input = "Potato Tomato";
    input.ToUpper();
    Console.WriteLine(input);
}
```

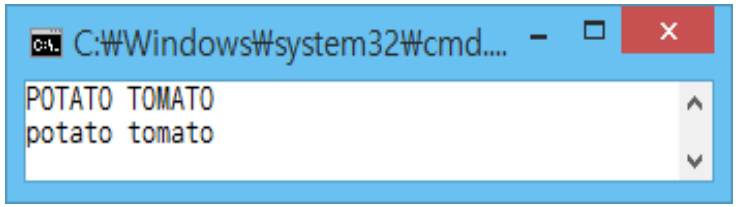
응용예제

응용 예제 4-1 (문자열 처리)

- 대문자화와 소문자화 - ToUpper(), ToLower()

```
01   static void Main(string[] args)
02   {
03         // 대문자화와 소문자화
04         string input = "Potato Tomato";
05         Console.WriteLine(input.ToUpper());
06         Console.WriteLine(input.ToLower());
07   }
08
```

StringProcess 실행 결과



응용예제

응용 예제 4-1 (문자열 처리)

- 문자열 자르기 - Split()
  - 문자열에 있는 특정한 특정한 문자를 사용해 문자열을 자르고
  - 문자열 배열로 반환해주는 메서드

메서드	설명
Split()	문자열을 특정한 문자 또는 문자열로 자릅니다.

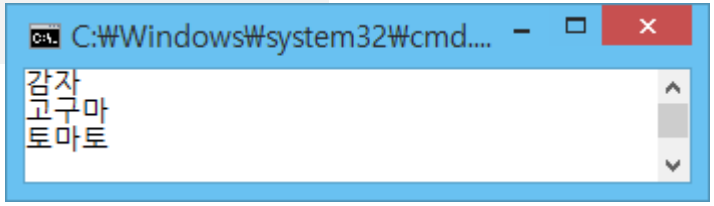
응용예제

응용 예제 4-1 (문자열 처리)

- 문자열 자르기 - Split()

```
01  static void Main(string[] args)
02  {
03      // 문자열 자르기
04      // string input = "감자 고구마 토마토";
05      input = "감자 고구마 토마토";
06      string[] inputs = input.Split(new char[] { ' ' });
07
08      foreach (var item in inputs)
09      {
10          Console.WriteLine(item);
11      }
12  }
```

StringProcess 실행 결과



응용예제

응용 예제 4-1 (문자열 처리)

- 문자열 양 옆의 공백 제거하기 - Trim(), TrimStart(), TrimEnd()
  - 문자열 양 옆, 문자열 앞, 문자열 뒤의 공백을 제거
    - 중요한 글자의 공백 삭제에 사용

메서드	설명
Trim()	문자열 양 옆의 공백을 제거합니다.
TrimStart()	문자열 앞의 공백을 제거합니다.
TrimEnd()	문자열 뒤의 공백을 제거합니다.

응용예제

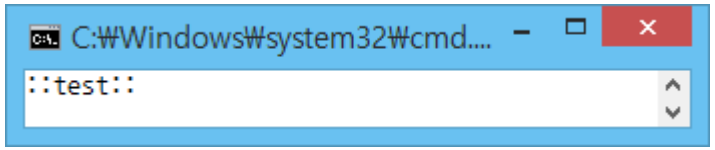
응용 예제 4-1 (문자열 처리)

- 문자열 양 옆의 공백 제거하기 - Trim(), TrimStart(), TrimEnd()

```
01  static void Main(string[] args)
02  {
03      // 문자열 양 옆의 공백 제거
04      string input = " test        \n";
05      Console.WriteLine("::" + input.Trim() + "::");
06      Console.Read();
07  }
08
```

양옆에 공백, 개행이 삭제되는지 확인하기 위한 글자

StringProcess 실행 결과



응용예제

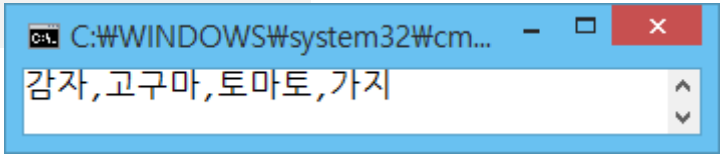
응용 예제 4-1 (문자열 처리)

- 배열을 문자열로 변환하기 - string.Join()
  - 배열에 있는 요소를 연결하여 문자열로 만들기

메서드	설명
string.Join()	배열의 요소를 뭉쳐 문자열로 변경합니다.

```
01  static void Main(string[] args)
02  {
03      // 배열을 문자열로
04      string[] array = { "감자", "고구마", "토마토", "가지" };
05      Console.WriteLine(string.Join(",", array));
06  }
07
08
```

StringProcess 실행 결과



응용예제

응용 예제 4-2 (이동하는 달팽이)

- 추가 메소드

  - Console.Clear(), Console.SetCursorPosition(), Thread.Sleep()

  – 콘솔 화면 지우기, 커서 옮기기, 스레드 정지하기

메서드	설명
Console.Clear()	콘솔 화면을 지웁니다.
Console.SetCursorPosition()	콘솔 화면의 특정한 위치로 커서를 옮깁니다.
Thread.Sleep()	특정한 시간만큼 스레드를 정지합니다.

현재는 그냥 프로그램을 정지한다고 기억해도 됩니다. 스레드와 관련된 내용은 이후에 살펴보겠습니다.

- Console.Clear()

  – 화면에 모든 글자를 지우기

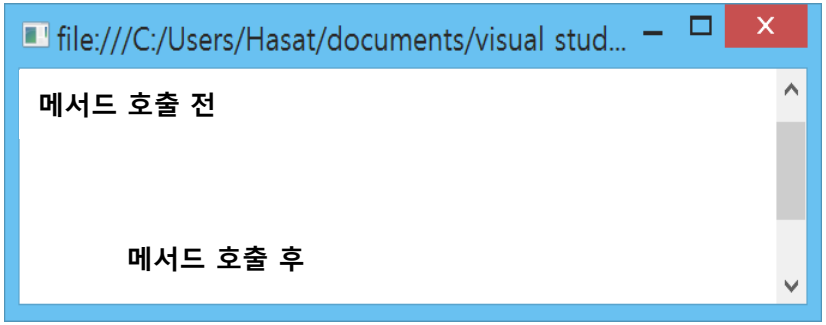
응용예제

응용 예제 4-2 (이동하는 달팽이)

- Console.SetCursorPosition()
  - 콘솔 화면의 특정 위치에 커서를 옮길 수 있음
  - 따라서 자신이 원하는 위치에 글자 출력 가능

```
01  static void Main(string[] args)
02  {
03      Console.WriteLine("메서드 호출 전");
04      Console.SetCursorPosition(5, 5);
05      Console.WriteLine("메서드 호출 후");
06  }
```

실행 결과



응용예제

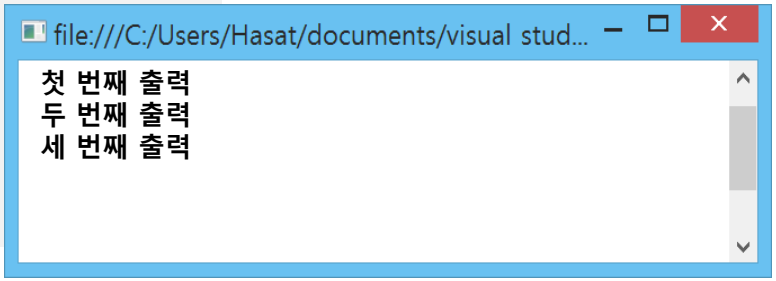
응용 예제 4-2 (이동하는 달팽이)

- Thread.Sleep()
  - 프로그램을 정지시키고 싶은 특정 시간만큼 정지시키기
  - 괄호 내부에 밀리초 단위로 입력 예) 1000 - 1초 정지

```
01 using System;
02 using System.Threading;
03
04 class Program
05 {
06   static void Main(string[] args)
07   {
08       Console.WriteLine("첫 번째 출력");
09       Thread.Sleep(1000);
10       Console.WriteLine("두 번째 출력");
11       Thread.Sleep(1000);
12       Console.WriteLine("세 번째 출력");
13   }
14 }
```

Thread 클래스를 사용하기 위해 반드시 필요한 라이브러리

실행 결과

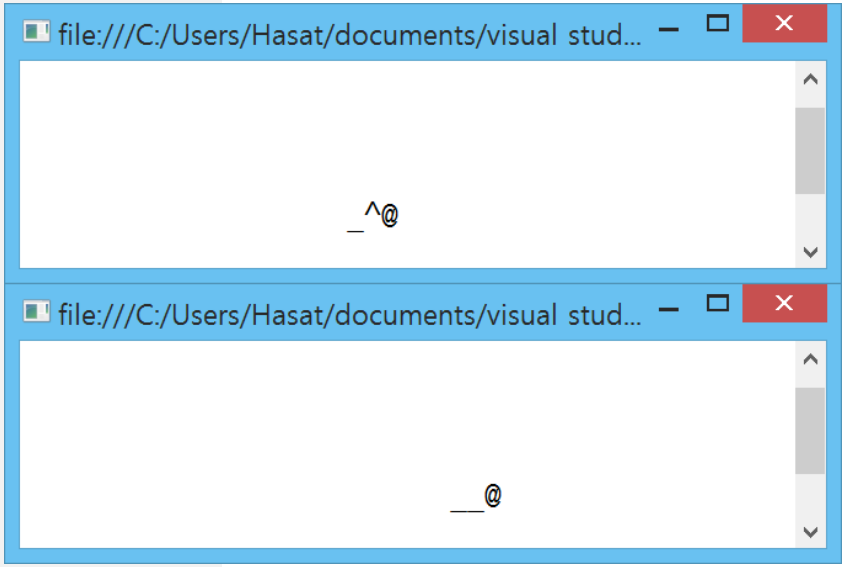


응용예제

응용 예제 4-2 (이동하는 달팽이)

```
01   static void Main(string[] args)
02  {
03        int x = 1;
04        while (x < 50)
05        {
06              // 화면을 지우고 커서를 이동합니다.
07              Console.Clear();
08              Console.SetCursorPosition(x, 5);
09
10              // 출력합니다.
11              if (x % 3 == 0)
12                  Console.WriteLine(" __@");
13              else if (x % 3 == 1)
14                  Console.WriteLine("_^@");
15              else
16                  Console.WriteLine("^_@");
17
18              // 100밀리초 정지하고 x를 증가합니다.
19              Thread.Sleep(100);
20              x++;
21                }
22  }
```

MovingAt 실행 결과



응용예제

응용 예제 4-3 (switch 조건문과 무한 반복문)

```
01  static void Main(string[] args)
02 {  bool state = true;
03    while (state)
04    {
05        ConsoleKeyInfo info = Console.ReadKey();
06        switch (info.Key)
07        {
08             case ConsoleKey.UpArrow:
09                 Console.WriteLine("위로 이동");
10                 break;
11             case ConsoleKey.RightArrow:
12                 Console.WriteLine("오른쪽으로 이동");
13                 break;
14             case ConsoleKey.DownArrow:
15                 Console.WriteLine("아래로 이동");
16                 break;
17             case ConsoleKey.LeftArrow:
18                 Console.WriteLine("왼쪽으로 이동");
19                 break;
20             case ConsoleKey.X:
21                 state = false;
22                 break;
23         }
24    }
25 }
```

ReadKey - 누른 키 정보를 표시하는데 사용

ConsolKey - 콘솔의 표준키를 지정
UpArrow 38위쪽 화살표 키입니다.
RightArrow 39오른쪽 화살표 키입니다.
DownArrow 40아래쪽 화살표 키입니다.
LeftArrow 37왼쪽 화살표 키입니다.

SwitchWithWhile

응용예제

응용 예제 4-3 (switch 조건문과 무한 반복문)

- 다음 요구사항을 고려하여 응용 예제 4-3을 변경한 예제 생각해보기
  - 방향키로 콘솔을 움직이는 골뱅이 기호 표시하기 예제(p190)

① X와 y 변수 만들기
② 방향 키를 움직일 때마다 해당 변수를 증가 또는 감소시키기
③ 반복이 한 번 끝나기 전에 화면을 지우고 x와 y 위치에 “@” 문자를 출력하기

감사합니다

mnshim@sungkyul.ac.kr