

컴퓨터정보과 C# 프로그래밍

2021년도/1학기/14주차
장은미

보충자료

다중상속이 되면?

```
class A{  
    int i;  
    void TTT() { ... }  
}
```

```
class B : A  
{  
    override void TTT() { ... }  
}
```

```
class C : A  
{  
    override void TTT();  
}
```



```
interface C  
{  
    void TTT();  
}
```

```
class D : B, C  
{  
    // D의 i와 TTT()는 누구의 것?  
}
```

한 학교에서, 강의자이면서 학생인 사람이 있다.

```
class Teacher
{
    public void Teach()
    {
    }
}
```

```
class Student
{
    public void Study()
    {
    }
}
```

```
class TeacherAndStudent : Teacher, Student
{
}
}
```

클래스 불가능다중상속 금지
실제로는 사용이 불가능

```
interface ITeachable
```

```
{  
    void Teach();  
}
```

```
class Teacher : ITeachable
```

```
{  
    public void Teach()  
    {  
    }  
}
```

```
interface IStudiable
```

```
{  
    void Study();  
}
```

```
class Student : IStudiable
```

```
{  
    public void Study()  
    {  
    }  
}
```

```
class TeacherAndStudent : ITeachable, Student
```

```
{  
    public void Teach()  
    {  
    }  
}
```

```
class TeacherAndStudent : IStudiable, Teacher
```

```
{  
    public void Study()  
    {  
    }  
}
```

```
class TeacherAndStudent : IStudiable, ITeachable
```

```
{  
    public void Teach()  
    {  
    }  
  
    public void Study()  
    {  
    }  
}
```

```
TeacherAndStudent a = new TeacherAndStudent();  
a.Teach();  
a.Study();
```

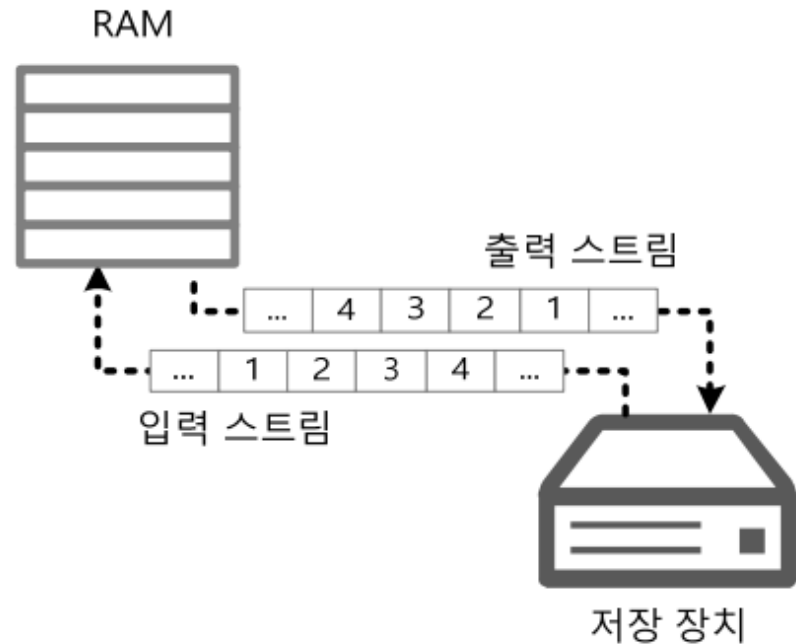
```
-----  
ITeachable b = (ITeachable)a;  
b.Teach();
```

```
-----  
Student c = (Student) a;  
c.Study();  
-----
```

File

Stream

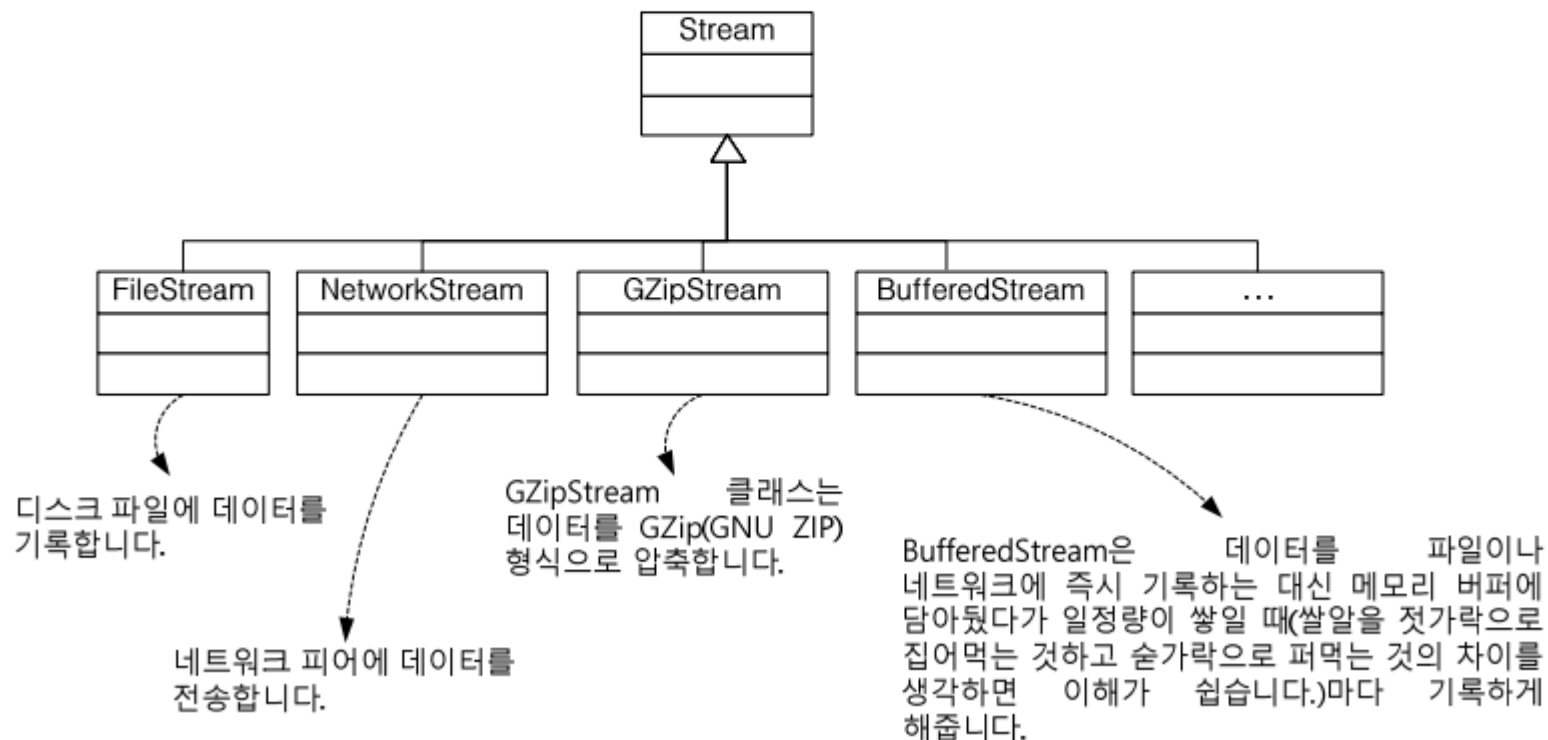
- 스트림(Stream)
 - 영어로 시내, 강, 또는 도로의 차선을 뜻함
 - 파일을 다룰 때의 스트림은 “데이터가 흐르는 통로”를 뜻함
 - 메모리에서 하드 디스크로 데이터를 옮길 때, 스트림을 만들어 둘 사이를 연결한 뒤에 메모리의 데이터를 바이트 단위로 하드 디스크로 옮김.
 - 그 반대의 경우도 마찬가지.



Stream

- System.IO.Stream 클래스

- 입력 스트림, 출력 스트림의 역할을 모두 수행
- 순차접근방식과 임의 접근 방식 모두 지원
- 단, 추상 클래스이기 때문에 이 클래스의 파생 클래스를 이용해야 함.



파일처리

- 파일(File)은 컴퓨터 저장매체에 기록되는 데이터의 묶음
- 디렉토리(Directory)는 파일이 위치하는 주소
 - 파일(서류)를 담는다는 의미에서 폴더(Folder:서류철) 라고 부르기도 함

클래스	설명1	설명2
File	파일의 생성, 복사, 삭제, 이동, 조회	정적 메소드
FileInfo		인스턴스 메소드
Directory	디렉토리의 생성, 삭제, 이동, 조회	정적 메소드
DirectoryInfo		인스턴스 메소드

폴더 다루기

기능	Directory	DirectoryInfo
생성	<code>DirectoryInfo dir = Directory.CreateDirectory("a");</code>	<code>DirectoryInfo dir = new DirectoryInfo("a"); dir.Create();</code>
삭제	<code>Directory.Delete("a");</code>	<code>DirectoryInfo dir = new DirectoryInfo("a"); dir.Delete();</code>
이동	<code>Directory.Move("a", "b");</code>	<code>DirectoryInfo dir = new DirectoryInfo("a"); dir.MoveTo("b");</code>
존재여부 확인	<code>if (Directory.Exists("a.dat")) // ...</code>	<code>DirectoryInfo dir = new DirectoryInfo("a"); if (dir.Exists) // ...</code>
속성 조회	<code>Console.WriteLine(Directory.GetAttributes("a"));</code>	<code>DirectoryInfo dir = new DirectoryInfo("a"); Console.WriteLine(dir.Attributes);</code>
하위 폴더 조회	<code>string[] dirs = Directory.GetDirectories("a");</code>	<code>DirectoryInfo dir = new DirectoryInfo("a"); DirectoryInfo[] dirs = dir.GetDirectories();</code>
하위 파일 조회	<code>string[] files = Directory.GetFiles("a");</code>	<code>DirectoryInfo dir = new DirectoryInfo("a"); FileInfo[] files =dir.GetFiles();</code>

파일 다루기

기능	File	FileInfo
생성	<pre>FileStream fs = File.Create("a.dat");</pre>	<pre>FileInfo file = new FileInfo("a.dat"); FileStream fs = file.Create();</pre>
복사	<pre>File.Copy("a.dat", "b.dat");</pre>	<pre>FileInfo src = new FileInfo("a.dat"); FileInfo dst = src.CopyTo("b.dat");</pre>
삭제	<pre>File.Delete("a.dat");</pre>	<pre>FileInfo file = new FileInfo("a.dat"); file.Delete();</pre>
이동	<pre>File.Move("a.dat", "b.dat");</pre>	<pre>FileInfo file = new FileInfo("a.dat"); file.MoveTo("b.dat");</pre>
존재여부 확인	<pre>if (File.Exists("a.dat")) // ...</pre>	<pre>FileInfo file = new FileInfo("a.dat"); if (file.Exists) // ...</pre>
속성 조회	<pre>Console.WriteLine(File.GetAttributes("a.dat"));</pre>	<pre>FileInfo file = new FileInfo("a.dat"); Console.WriteLine(file.Attributes);</pre>

FileStream

byte, byte[] 단위로 처리,

```
long someValue = 0x123456789ABCDEF0;

// 1) 파일 스트림 생성
Stream outputStream =
    new FileStream("a.dat", FileMode.Create);

// 2) someValue(long 형식)을 byte 배열로 변환
byte[] wBytes =
    BitConverter.GetBytes(someValue);

// 3) 변환한 byte 배열을 파일 스트림을
    통해 파일에 기록
outputStream.Write(wBytes, 0, wBytes.Length);

// 4) 파일 스트림 닫기
outputStream.Close();
```

```
byte[] rBytes = new byte[8];

// 1) 파일 스트림 생성
Stream inputStream =
    new FileStream("a.dat", FileMode.Open);

// 2) rBytes의 길이만큼(8바이트) 데이터를
    읽어 rBytes에 저장
inputStream.Read(rBytes, 0, rBytes.Length);

// 3) BitConverter를 이용하여 rBytes에
    담겨있는 값을 long 형식으로 변환
long readValue =
    BitConverter.ToInt64(rbytes, 0);

// 4) 파일 스트림 닫기
inputStream.Close();
```

File의 종류

- FileStream은 반드시 데이터 RW할 때, byte, byte[] 단위로 처리해야 함.
- 이진 파일 - BinaryWriter , BinaryReader
- 텍스트 파일 - StreamWriter, StreamReader

Helper Class

- StreamWriter : 텍스트 데이터 쓰기

```
StreamWriter sw = new StreamWriter( new FileStream("a.dat", FileMode.Create) );  
sw.WriteLine(int.MaxValue);  
sw.WriteLine("Good Morning!");  
sw.Close();
```

- StreamReader : 텍스트 데이터 읽기

```
StreamReader sr = new StreamReader( new FileStream("a.dat", FileMode.Open) );  
while ( sr.EndOfStream == false )  
{  
    Console.WriteLine(sr.ReadLine());  
}  
sr.Close();
```