

# 컴퓨터정보과 C# 프로그래밍

---

6주차 접근제한자, 프로퍼티, ...

# 강의 순서

1. C# 환경설치 / C# 기본 구조
2. 클래스 기본(필드) + 변수, 자료형
3. 클래스 기본(메소드) + 연산자, 수식
4. 클래스 기본(메소드) + 제어문
5. 배열/리스트/딕셔너리
6. **클래스 기본: 접근제한자 (한정자) + 프로퍼티(속성)**
7. 클래스 심화: 상속
8. 클래스 심화: 인터페이스/추상 클래스
9. 예외처리/일반화
10. 파일처리
11. UI (Winform or WPF)
12. LINQ/Delegate/Lambda/...

# 복습 및 5주차 추가요소

- **Array**
  - 고정길이, 인덱스(0~, 순서번호)를 통한 접근
  - 1차원 배열 []
    - Length 프로퍼티
  - n차원 배열 [, ]
    - GetLength() 메소드
  - 가변배열 [][]
- **Generic Collection**
  - **List<T>**
    - T형식의 데이터를 저장한다.
    - 일종의 가변길이 Array
      - 인덱스를 통한 접근
    - Property : Count
    - Method : Add(), Insert(), RemoveAt(), Remove()
  - **Dictionary<Tkey, Tvalue>**
    - Tkey형식의 Key와 Tvalue 형식의 Value를 한 쌍으로 갖음
    - Key를 통한 접근
    - Property : Count
    - Method: Add(), Remove()

# 객체지향 특징

- 추상화 (Abstraction)
  - 어떤 집단의 공통성과 본질을 모아서 추출
  - class/struct
- 상속 (Inheritance)
  - 기존의 클래스를 재사용하여 새로운 클래스를 작성하는 문법 요소
  - 상속
- 다형성 (Polymorphism)
  - 어떤 객체의 속성이나 기능이 상황에 따라 여러 가지 형태를 가짐.
  - method overloading, method overriding,
- 캡슐화 (Encapsulation)
  - 연관있는 속성과 기능을 하나의 캡슐로 만들어 데이터를 외부로부터 보호
  - 접근제한자

# 접근제한자

---

감출 것은 감추고, 보여줄 것만 보여주자

# 접근제한자 (Access Modifier)

- 캡슐화를 구현하기 위한 문법
- 필드, 메소드 등 멤버의 외부 노출여부를 결정
  - 되도록이면 모두 감추고 일부만 노출하는 것이 좋음
  - 클래스, 필드, 메소드, 프로퍼티 등의 요소에 적용할 수 있음.
- 주요 접근제한자

접근제한자	클래스 내부	클래스 외부	파생 클래스	동일 어셈블리	다른 어셈블리
<b>public</b>	O	O	O	O	O
protected	O	X	O	X	X
<b>private</b>	O	X	X	X	X
internal	O	O	O	O	X

- 기본 접근 제한자
  - class : internal
  - 그 외: private

# 프로퍼티

---

C#의 getter, setter

# 속성 (Property)

- 캡슐화와 편의성이 모두 필요해!
  - private으로 설정한 class의 field는 접근을 위해 getter/setter(메소드)를 사용해야 한다. → 불편해...
  - public으로 설정해서 편하게 한다. → 데이터 유효성을 보장하지 못해...
- 구성
  - 접근자(accessor)
    - get : 필드의 값을 읽을 때
      - 반드시 값을 반환해야 한다.
    - set : 필드의 값을 할당할 때
      - value 키워드 : 외부에서 들어오는 데이터를 갖고 있는 매개변수로 간주
  - 예제

```
class Test {  
    private int _age;  
    public int Age {  
        get { return _age; }  
        set { _age = value; }  
    }  
}
```

```
Test t = new Test();  
//t._age = 1;  
t.Age = 1;  
Console.WriteLine(t.Age);
```



# 자동 구현 프로퍼티

## ■ 예제

기본	자동구현프로퍼티	
	C# 3.0 ~	C# 7.0~
<pre>class Test {     private int _age;     public int Age {         get { return _age; }         set { _age = value; }     } }</pre>	<pre>class Test {     public int Age {         get;         set;     } }</pre>	<pre>class Test {     public int Age {         get;         set;     } = 1; }</pre>
<pre>class Test {     private int _age;     public int Age {         get =&gt; _group;         set =&gt; _group = value;     } }</pre>	<pre>class Test {     public int Age {         get;         private set;     } }</pre>	

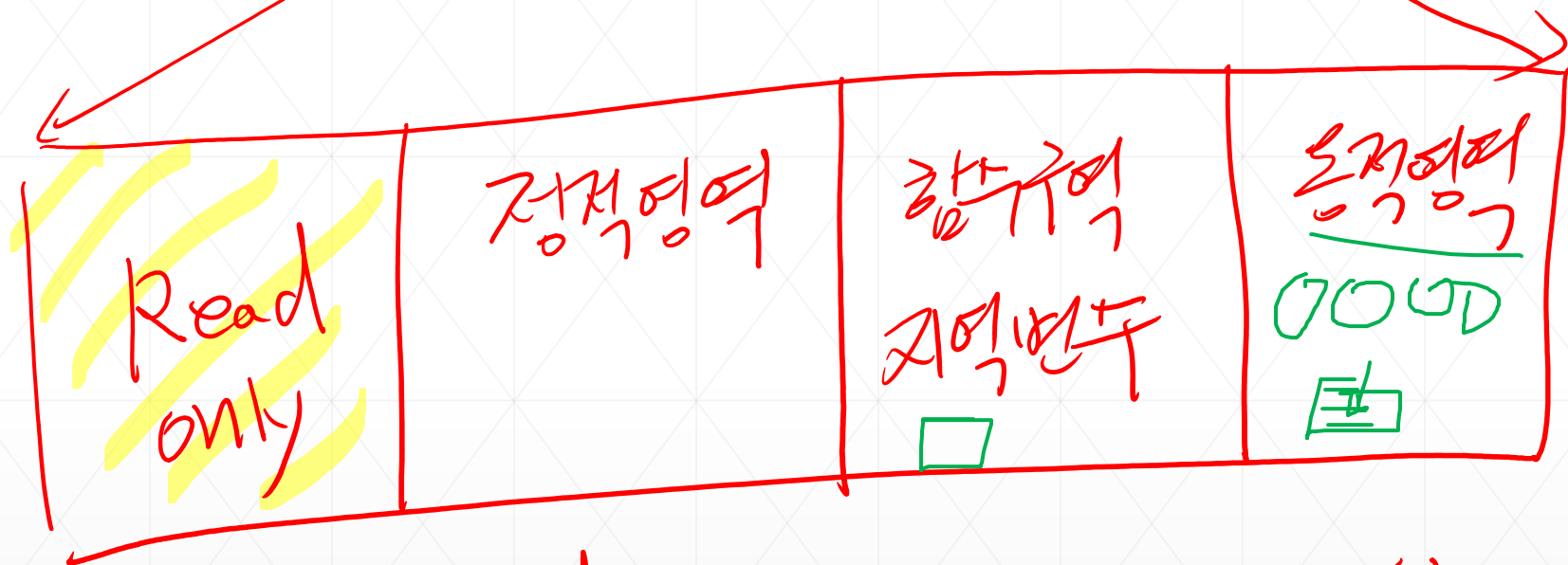
# 메모리~메모리~

---

메모리 구역

RAM

A



Code

data

Stack

Heap

[Static field]

[지역변수]

data

VT. S-max

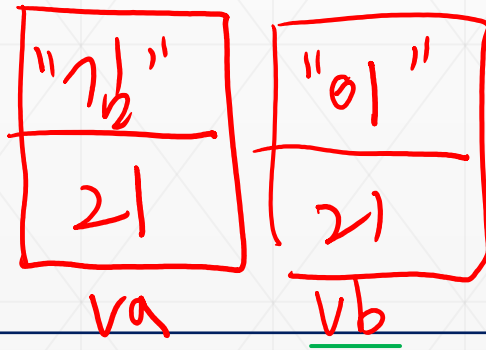


RT. S-max



Stack

types  
V (value type)



Heap

