

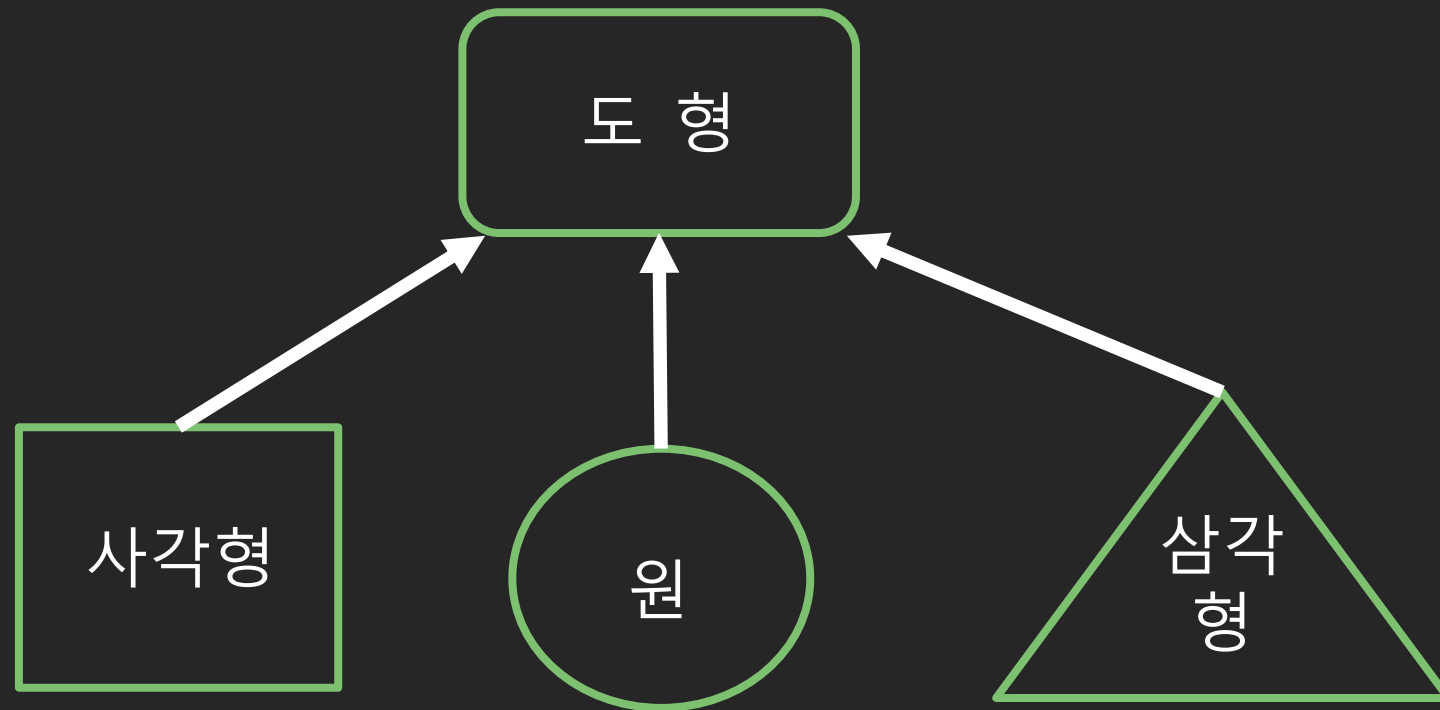
초보자도 할 수 있는 C# Programming

Fast Campus
Online

한창민 강사.

7-2강 상속과 다형성

- 클래스 상속?(IS-A 관계)
 - 부모클래스: base, parent, 상위, super
 - 자식클래스: derived, child, 파생, sub
 - 클래스의 재사용



- 클래스 상속? → 소스코드 (_081_01_classInheritance)

```
class Super
{
    protected int a;

    public void Print() {
        Console.WriteLine("Super Print()");
    }
}

class Sub:Super
{
    int b;

    public void Print() {
        Console.WriteLine("Sub Print() a: {0}, b: {1}", a, b);
    }
}
```

- 클래스 상속 - 생성자, 소멸자
 - 상속에서 생성자 함수와 소멸자 함수

➡ 소스코드 (_081_02_classInheritance)

- base 키워드

➡ 소스코드 (_081_03_class_base)

```
public Sub(int num, string name):base(num)
```

```
base.PrintSuper();
```

- is 키워드

- 객체의 형식 검사
- bool 리턴

```
if(aa is BB)
```

```
else if(aa is AA)
```

- as 키워드

- 형식 변환
- null 리턴

```
BB copyBB = bb as BB;
```

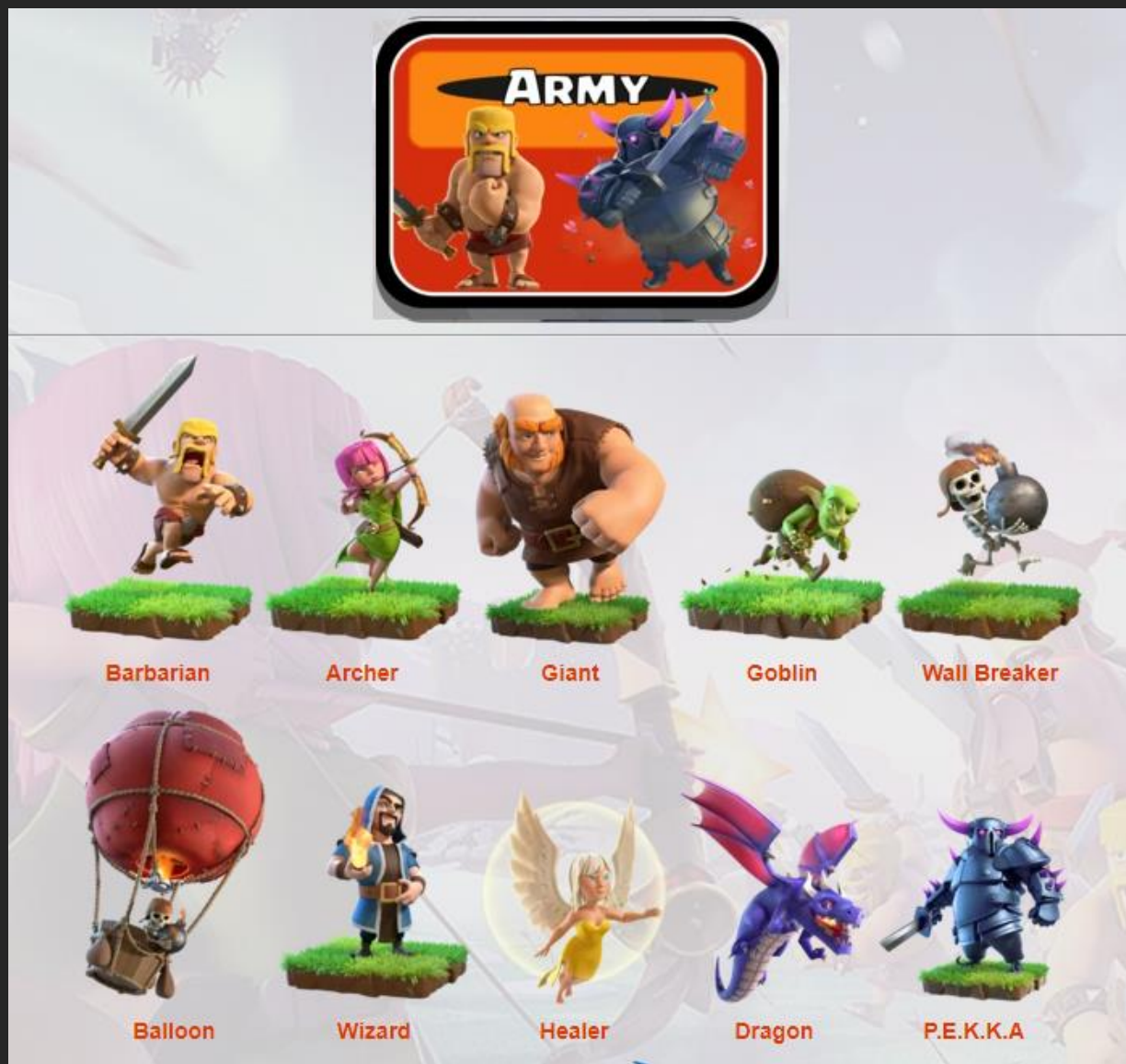
- 다형성 – **virtual, override**

- 객체 지향의 핵심
- 함수의 오버라이딩(재정의)
- 반복문으로 객체 관리

```
public virtual void Print()
```

```
public override void Print()
```

• 다형성



- sealed 키워드 → 상속, 재정의 불가
 - 클래스 sealed

```
sealed class CC
```

```
class DD : CC //CC클래스가 sealed라 오류
```

- 함수 sealed

```
class AAA  
    public virtual void PrintParent()
```

```
class BBB : AAA  
  
    public sealed override void PrintParent()
```

```
class CCC : BBB  
    public override void PrintParent()
```

- 클래스 HAS - A관계
 - 클래스가 다른 클래스를 가지는 구조
 - 두 개의 클래스가 매우 강한 연관성을 가짐

➡ 소스코드 (_085_class_has_a)

```
class AA
{
}

class BB
{
    AA[] aa;
}

class Program
{
    static void Main(string[] args) {
        BB bb = new BB();
        bb.SetNum(3, 200);

        bb.Print();
    }
}
```

- partial 키워드
 - 클래스를 나누어서 구현
 - 콘텐츠별 구분해서 코딩 가능

➡ 소스코드 (_086_partial)

```
partial class AA
```

```
▶ C# AA.ADD.cs
```

```
▶ C# AA.cs
```

```
▶ C# AA.MUL.cs
```

- 확장 메소드
- this 키워드

➡ 소스코드 (_086_partial)

```
static class Util
{
    public static void Print(this AA aa, string str) {
        aa.View(str);
    }

    public static void Sum(this int a) {
        Console.WriteLine("{0} + {1} = {0}", a, a, a + a);
    }
}
```

- 클래스 vs. 구조체

	class	struct
특징	참조 타입(Reference) - 힙에 생성	값 타입(Value) - 스택에 생성
사용	<code>new</code> 연산자	<code>new</code> 연산자 없는 생성 가능
생성자	파라미터 없는 생성자 가능	반드시 파라미터가 있어야 함

- 자료의 크기가 작을때
- 구조가 단순할때

➡ 소스코드 (`_088_struct`)