

2주 복습

+ 추가

1. 산술이항자 (+ - * / %) 하나씩 해보기
2. 할당연산자 (= += -= ..) 하나씩 해보기
3. 증감 연산자(++ --) 앞뒤로 하나씩 해보기
4. 2개 숫자 입력후 A가 짝수 이고 B가 5보다 큰가를 if문으로
5. 2개 숫자 입력후 A가 홀수 이거나 B가 3보다 작은가를 else if문 으로
6. else문으로 없음 출력하기
7. 숫자 입력받은거 만큼 문장 출력하기(while)
8. 숫자 입력받은거 만큼 문장 출력하기(for)
9. 위에 4,5,6,7,8을 메소드로 뽑아서 쓰기
10. for문 안에서 숫자를 계속 입력받는다.
 - 1 숫자가 짝수인경우 continue로 넘김.
 - 2 숫자가 10인경우 break로 for문 종료.
 - 3 그 외의 숫자인 경우 “다음숫자 입력”을 출력

추가 내용

키입력

멤버 변수

Github - 이슈

```

static void Main(string[] args)

    //키입력값받는 변수
    ConsoleKeyInfo cki;

    //#의 시작위치
    int x = 10, y = 10;

    while (true)//무한루프
    {
        //화면 초기화
        Console.Clear();

        //커서를 x,y의 위치로 이동
        Console.SetCursorPosition(x, y);

        //#을 입력
        Console.Write('#');

        //누르는 키를 입력받아 true값이면 넣음
        cki = Console.ReadKey(true);

        switch (cki.Key)
        {
            case ConsoleKey.LeftArrow: x--; break;

            case ConsoleKey.RightArrow: x++; break;

            case ConsoleKey.UpArrow: y--; break;

            case ConsoleKey.DownArrow: y++; break;

            case ConsoleKey.Q: return;
        }
    }

```

키입력

ConsoleKeyInfo 구조체가 키 입력을 받는다.

Console.Clear() 가 화면 모두 지움.

Console.SetCursorPosition(x,y)로 커서 위치 이동

Console.ReadKey(true) 로 값을 받음.

switch문에서 키입력에 맞는 값 적용됨.

참조 0개

```
class Student
```

```
{  
    int numberOne = 1;
```

참조 0개

```
public void YourName()
```

```
{  
    int numberTwo = 2;  
  
    System.Console.WriteLine(numberOne);  
    System.Console.WriteLine(numberTwo);  
    System.Console.WriteLine(numberThree);  
}
```

참조 0개

```
public void MyName()
```

```
{  
    int numberThree = 3;  
  
    System.Console.WriteLine(numberOne);  
    System.Console.WriteLine(numberTwo);  
    System.Console.WriteLine(numberThree);  
}
```

멤버 변수

중괄호를 기준으로 값을 사용할 범위를 정의한다.

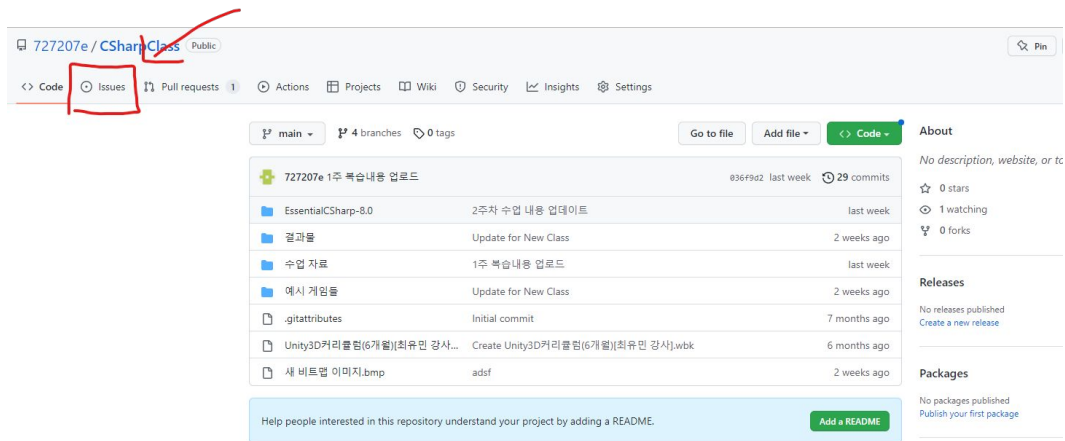
`numberOne`은 내부에서 모두 사용 가능하다.

`YourName()` 메소드안에서 `numberTwo`는 사용 가능하다.

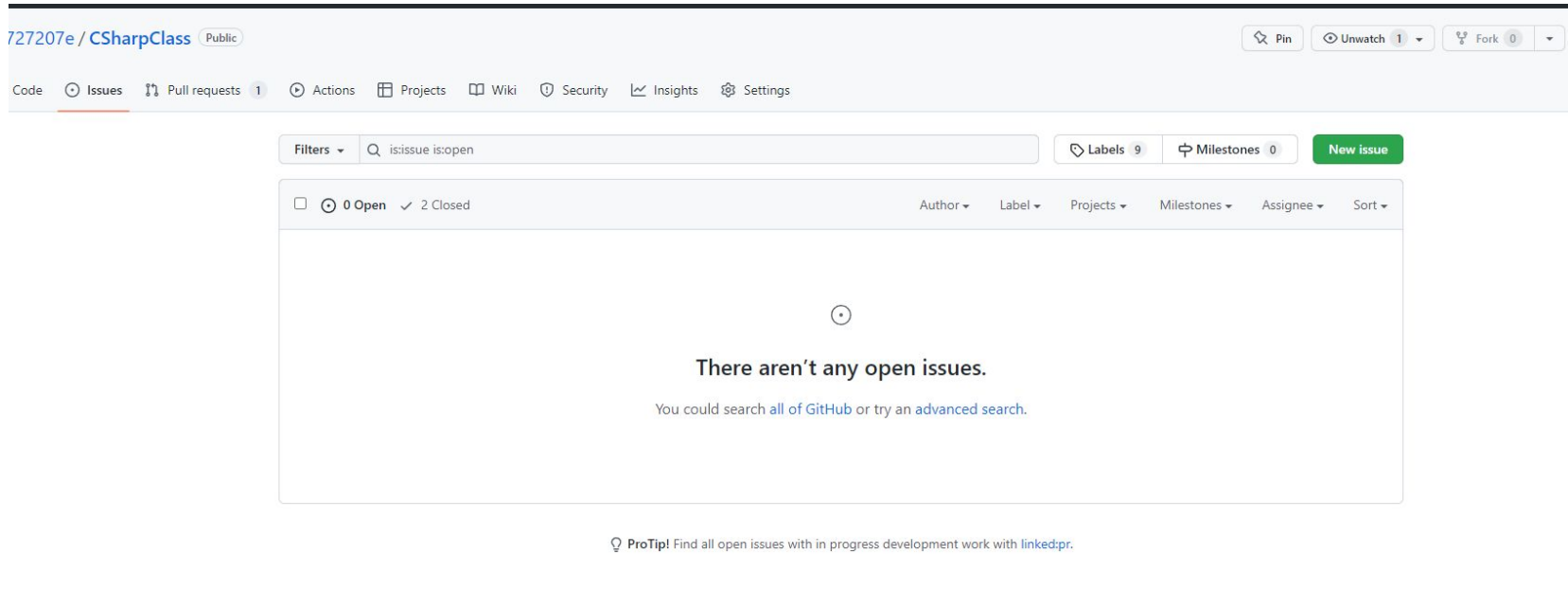
하지만 `numberThree`는 `MyName()`에서 정의되었으므로 사용 불가능하다.

이슈

해당 폴더에서 해야할 작업 / 생긴 버그 / 등등을 기록하는 공간.



우측상단에 초록상자 NewIssue를 눌러 이슈를 제작한다.



제목 입력 / 본문 내용 작성 / 우측에 작업하는 사람 할당(Assigness)

완료되면 우측하단에 초록버튼으로 이슈 제작

Issues

Pull requests 1

Actions

Projects

Wiki

Security

Insights

Settings

[Bug] 공격기 입력 안됨.

Write

Preview

H B I

문제 상황
알수없는 이유로 "칼을 찬 경우" 공격기 입력이 안됨.

Due date
2월 17일까지 진행할것.

Attach files by dragging & dropping, selecting or pasting them.

Styling with Markdown is supported

Submit new issue

Assignees

727207e

Labels

Projects

None yet

Milestone

No milestone

Development

Shows branches and pull requests linked to this issue.

Helpful resources

[GitHub Community Guidelines](#)

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

[Bug] 공격기 입력 안됨. #6

Edit New issue

Open 727207e opened this issue now · 0 comments

727207e commented now Owner

문제 상황

알수없는 이유로 "칼을 찬 경우" 공격기 입력이 안됨.

Due date

2월 17일까지 진행할것.

727207e added the bug label now

727207e self-assigned this now

Assignees

727207e

Labels

bug

Projects

None yet

Milestone

No milestone

Development

Create a branch for this issue or link a pull request.

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close issue Comment

Remember, contributions to this repository should follow our GitHub Community Guidelines.

Notifications

Unsubscribe

You're receiving notifications because you're watching this repository.

1 participant

Lock conversation

Pin issue

아래에 댓글 달 수 있음.
해당 이슈를 종료(Close)할 수 있음.
우측에 해당 이슈를 작업할 Branch를 제작할 수 있음.
작업이 끝나면 해당 이슈를 닫으면 된다.

6장

클래스



클래스

클래스는 일종의 빵틀.

어떤값을 넣는지, 누가 만드는지
어디서 만드는지에 따라 다른 녀석이
된다.

참조 0개

```
internal class Program
{
    참조 0개
    static void Main(string[] args)
    {
        Student AStudent = new Student();
        AStudent.name = "최유민";
        AStudent.age = 28;

        Student BStudent = new Student();
        BStudent.name = "uuuuu";
        BStudent.age = 21;

        System.Console.WriteLine($"A학생 이름 : {AStudent.name}");
        System.Console.WriteLine($"B학생 이름 : {BStudent.name}");
    }
}
```

참조 4개

```
class Student
{
    public string name;
    public int age;
}
```

C:\WINDOWS\system32\cmd

A학생 이름 : 최유민

B학생 이름 : uuuuu

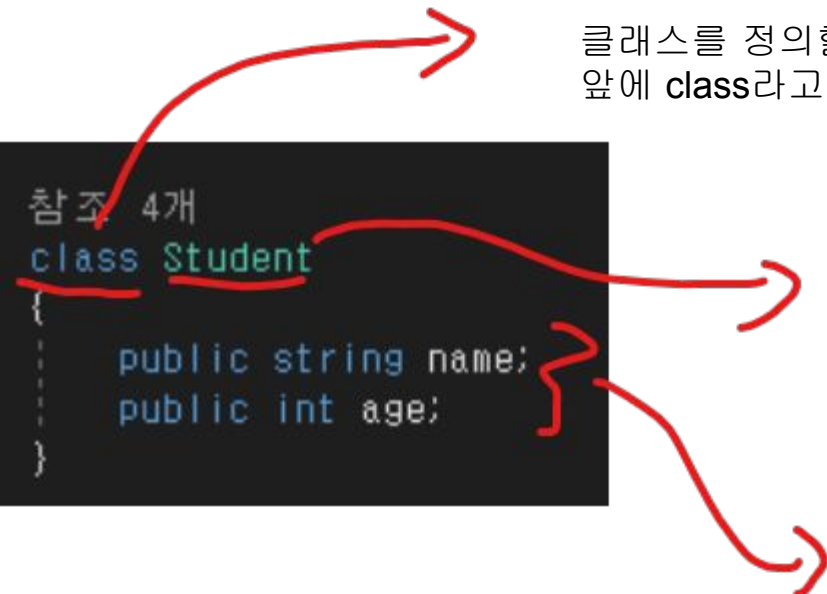
계속하려면 아무 키나 누르

Class

클래스를 정의할때
앞에 **class**라고 작성한다.

참조 4개

```
class Student
{
    public string name;
    public int age;
}
```



Class 이름


클래스이름

Field


필드라고 불린다.
클래스가 가지고 있는 정보이다.

클래스 인스턴스

클래스를 사용할 거라고 변수처럼 선언한다.
해당 클래스를 사용할 공간을 할당받는다.
이를 인스턴스화라고 한다.



```
Student AStudent = new Student();  
AStudent.name = "최유민";  
AStudent.age = 28;
```



클래스가 가지고 있는 데이터를 쓰기위해선
클래스뒤에 .(점) 을 붙여서 가져와서 수정하거나 쓸 수
있다.

```
Student AStudent = new Student();  
AStudent.name = "최유민";  
AStudent.age = 28;  
  
Student BStudent = new Student();  
BStudent.name = "uuuuu";  
BStudent.age = 21;
```

앞서말한 빵틀처럼
같은 클래스를 가져다 썼지만 다른 결과가 된다.

Astudent는 최유민이고
Bstudent는 uuuu 이다.

참조 4개

```
class Student
{
    public string name;
    public int age;

    참조 2개
    public void WriteDownMyName()
    {
        System.Console.WriteLine($"제 이름은 {name}입니다.");
    }
}
```

클래스 메소드

클래스안에 메소드를 작성할 수 있다.
물론 외부에서 사용할 수 있다.

바로 값을 정의할 수 있다.

참조 2개

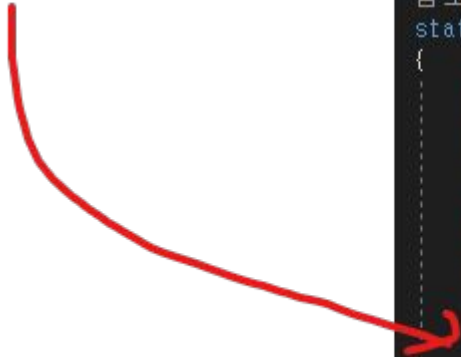
```
class Student
{
    public string MyCase1 = "WhatIsThis";
}
```

참조 8개

```
static void Main(string[] args)
{
    Student AStudent = new Student();
    AStudent.name = "최유민";
    AStudent.age = 28;

    Student BStudent = new Student();
    BStudent.name = "uuuuu";
    BStudent.age = 21;

    AStudent.WriteDownMyName();
    BStudent.WriteDownMyName();
}
```



1. 학생 클래스를 만들어서 학생 하나의 정보를 입력받는다(ReadLine) -> 출력
2. 학생 클래스를 만들어서 메소드로 학생 하나의 정보를 입력받는다. -> 출력
3. 학생 클래스를 만들어서 for문으로 학생 여럿을 입력받는다 (배열) -> 출력
4. 먹을 것 클래스를 만들어서 음식을 여러개 입력받고, 특정 음식을 입력하면 제외된다. -> 출력
5. 나이에 해당하는 띠를 출력해주는 클래스를 만든다. 만 나이도 계산되어 출력한다.
6. 게임들을 보유하고 있는 게임 클래스를 제작(필드에 미리 배열로 정의되어 있음)
게임을 입력하면 배열에 값이 있는지 비교해 있는지 없는지를 출력해준다.
7. 계산기 클래스를 만들어서 계산기에 필요한 모든 내용을 넣는다
숫자 + 기호 + 숫자로 결과값을 출력한다. (예, $3 * 2$)
결과를 저장하고, 계속 진행된다.
8. 계산기를 수정해서, 숫자 + 기호 + 숫자 인데, 첫번째 숫자는 그 전 결과값을 이어받는다.
(예, $3*2 = 6$. 다음 계산 6 으로 시작해서 $+2 = 8$)

9. 서점 클래스에 제목, 지은이, 출판사를 입력받아 책 데이터를 쌓는다.
특정 키를 입력받을때 까지 지속되며, 특정 키 입력을 받으면
제목을 입력받고 제목에 맞는 지은이, 출판사를 순서대로 출력해준다.

10. 메뉴 클래스에 햄버거, 가격, 세트 종류 를 입력받아 데이터를 쌓는다.
특정 키를 입력받을때 까지 지속되며, 특정 키 입력을 받으면
햄버거를 입력받고, 특정키를 입력받을때 까지 지속되며,
지금까지 나온 햄버거 종류들과 햄버거 가격 합을 출력한다.

11. 통장 클래스를 만든다. 입금 또는 출금 중 하나를 선택한다(**readline**으로 1이나 2를 입력받음)
입금으로 현재 잔고를 출력하고, 출금으로 현재 잔고를 출력한다.

12. 학생 클래스를 만들어서 학생들의 이름과 반을 입력받아 배열로 저장한다.
이후, 선생 클래스를 만들고, 선생의 데이터를 배열로 가지고 있다.
번째 선생의 데이터를 고르면, 선생이 담당하는 반의 학생들을 모두 출력한다.

참조 4개

```
class Student
{
    public string MyCase1;
    private string MyCase2;
    protected string MyCase3;
}
```

캡슐화

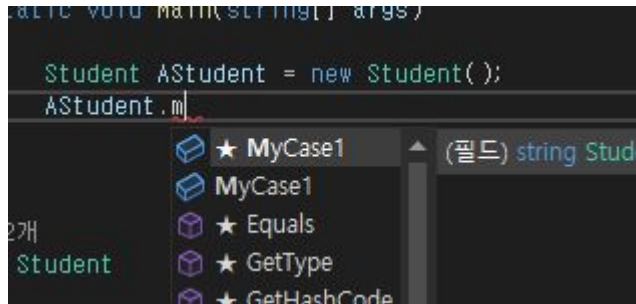
원하는 값만 보여주고 숨길 수 있다(유지 관리가 편함)

public - 외부에서 사용 가능

private - 외부에서 사용 불가

protected - 상속받은 자식만 사용 가능

아래 그림과 같이
MyCase1 밖에 사용 못함



위에 만들었던 클래스들중에 안쓰는 애들은 모두 **private**로 바꿔본다.

7장

상속

```

namespace ConsoleApp1
{
    참조 0개
    internal class Program
    {
        참조 0개
        static void Main(string[] args)
        {
            Classes cccc = new Classes();
            cccc.WriteSchoolInfo("한수초등학교");
        }
    }

    참조 1개
    class School
    {
        protected string SchoolName;

        참조 1개
        protected void WriteSchoolName()
        {
            System.Console.WriteLine($"학교 이름 : {SchoolName}");
        }
    }

    참조 2개
    class Classes : School
    {
        private string ClassName = "3반";

        참조 1개
        public void WriteSchoolInfo(string _schoolName)
        {
            SchoolName = _schoolName;

            WriteSchoolName();
            System.Console.WriteLine($"내 반은 {ClassName}");
        }
    }
}

```

C:\WINDOWS\system32\cmd.exe

학교 이름 : 한수초등학교
내 반은 3반
계속하려면 아무 키나 누르십시오

상속

한 클래스를
상속받는다고
말한다.

상속해주는 클래스를
부모 클래스.

상속 받는 클래스를
자식 클래스.

참조 1개

```
class School
```

```
{
```

```
    protected string SchoolName;
```

참조 1개

```
    protected void WriteSchoolName()
```

```
{
```

```
        System.Console.WriteLine($"학교 이름 : {SchoolName}");
```

```
}
```

School이란 클래스가 있다.

학교이름의 값을 가지고 있다.

protected는 이제 상속받을 자식이 사용할 수 있다.

참조 2개

```
class Classes : School
```

```
{
```

```
    private string ClassName = "3반";
```

참조 1개

```
    public void WriteSchoolInfo(string _schoolName)
```

```
    {
```

```
        SchoolName = _schoolName;
```

```
        WriteSchoolName();
```

```
        System.Console.WriteLine($"내 반은 {ClassName}");
```

```
    }
```

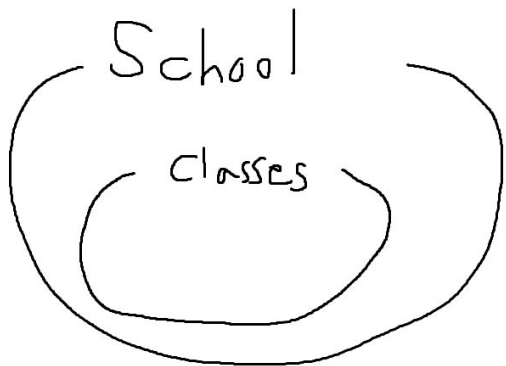
```
}
```

Classes : School

Classes란 클래스는 School 클래스를 상속 받는다.

Classes는

상속 받는 School의 데이터도 모두 사용할 수 있다.



1. 은행 클래스를 상속받는 기업은행 / 하나은행 / 신한은행

은행 클래스가 전체적으로 가질 정보들을 갖는다 (회원 아이디 / 회원 이름 / 적금 이자)

각각의 은행들은 적금 이자가 모두 다르다.

기업은행에 100만원 넣으면 1년뒤 나올 이자 금액,

하나은행 / 신한 은행 모두 각각 출력한다.

2. 식당 클래스를 상속받는 한식 / 양식 / 중식

식당 클래스가 전체적으로 가질 정보들을 갖는다 (돈 받기 / 주문한 음식 주기)

각각 식당들은 메뉴가 모두 다르다.

한식은 삼계탕 1만원 등등 다르다.

가고싶은 식당을 입력받고, 메뉴를 모두 띄워준다.

유저는 메뉴를 입력하고, 내야할 가격, 고른 메뉴를 띄워준다.

3. 예능 클래스를 상속받는 SBS / KBS / MBC

예능 클래스가 전체적으로 가질 정보들을 갖는다 (몇번 채널인가)

각각 채널들은 몇번인지 모두 다르다.

채널을 입력받으면 각 방송사가 틀어줄 수 있는 예능들을 띄워준다.

특정 예능을 입력하면 몇번채널 어느 방송사 어느 예능을 보는 중 이라고 띄워준다.

은행 클래스 만들기

은행 클래스 상속받는 기업 / 하나 / 카카오 / 신한 만들기.

각 은행을 배열로 넣기 (예, 은행[0] = 기업 , 은행[1] = 하나 ...)
은행 클래스는 송금하기 / 출금하기 / 입금하기 / 등이 있다.

—————

유저 클래스 만들기

내 데이터를 가지고 있다. 이름 / 보유중인 현금

—————

내가 입금하고 , 출금하고 , 다른 은행으로 송금하는 로직 구현하기