1. While문

* if문과 비슷하지만, 조건이 있다면 반복함
* 나뉘는 조건, 끝나는 조건이 꼭 있어야 함 (무한루프 주의)

1. do while문

* 꼭 실행해야 하는 do를 실행 후, while문 실행 (잘 사용하지 않음)

텍스트이(가) 표시된 사진

자동 생성된 설명

1. for 문

* for(초기식; 조건식; 증가식)

텍스트이(가) 표시된 사진

자동 생성된 설명

1. break, continue

* 조건이 만족해도 for 또는 while 문에서는 계속해서 반복을 할 수 있기 때문에, break를 사용해서 임의로 빠져나오게 함

1. 함수 // 한정자 반환형식 이름(매개변수목록) {}

* Class 안에 정의되야 함
* 참조(ref)
* 다음과 같이 Add\_One에서는 num만이 변경되었지만, 매개변수에 ref를 달면 main함수의 실제 int a의 값이 변함 (a라는 진짜 메모리를 넘기는 의미)

텍스트이(가) 표시된 사진

자동 생성된 설명

1. ref, out

* 하지만 디자인 측면에서 ref를 붙이는 것보다 원본 a에 값을 저장하는 것이 좀 더 유리함

텍스트이(가) 표시된 사진

자동 생성된 설명

* Ref 사용 예시 (int a, int b 값을 swap 할 때,)

텍스트이(가) 표시된 사진

자동 생성된 설명

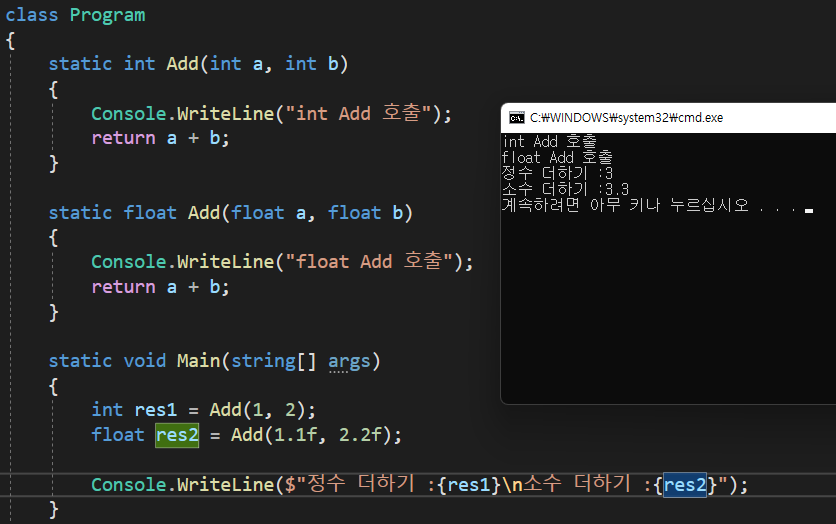
* 값을 여려개를 반환해야 하는 경우에는 out을 사용하여 넘길 수 있다 (ref와 같이 원본을 작업)

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 오버로딩 (과적화)

* 함수 이름의 재사용
* 함수의 이름이 같더라도(Add로 같음) 매개변수의 형식이 다르다면(int, float) 함수 이름을 같게 할 수 있다. 반환형식(int, float, void)는 영향을 주지 않는다.



* 선택형 매개변수
* 매개변수에 초기값을 넣어 “생략”을 할 수 있다.
* 매개변수의 형식이 (int. int, int, float, double)이기 때문에 함수를 사용할 때는 순서대로 사용해야 한다.
* C#에서는 따로 매개변수이름에 : 를 붙여, 값을 지정할 수 있다.

텍스트, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

1. 연습문제 99단 만들기(2단~9단)

* 내가 만든 것

텍스트이(가) 표시된 사진

자동 생성된 설명

* 강사님 코드

텍스트이(가) 표시된 사진

자동 생성된 설명

* 별을 찍는 코드

텍스트이(가) 표시된 사진

자동 생성된 설명

* 팩토리얼을 만드는 코딩

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

1. 디버깅

* 호출스텍 : 인셉션 같이 함수에 함수가 꼬리를 무는 상황에서 어디에서 호출되었는지 확인 가능
* Break point : 조건식을 걸어서 확인 할 수 있다.