

메소드는 기능 또는 동작이었는데요.

먼저 형태는 아래와 같습니다.

리턴타입 메소드명(매개변수){

메소드 몸체 }

메소드는 리턴을 갖는데요. 리턴타입이 없을 경우 void를 사용합니다.

저희가 자바를 할 때마다 사용하는 public static void main(String[] args) { }

이것 또한 메소드입니다. 우리는 보통 "메인메소드다" 라고 하죠.

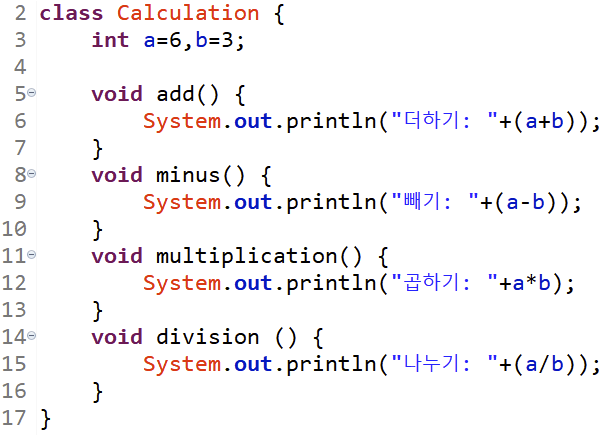
public은 접근제한지정자 이고 static은 정적메소드라 하고

void는 리턴타입인데 리턴타입이 없다는 뜻입니다.

main은 이 메소드의 이름이죠.

그리고 String[] args는 매개변수가 될 것입니다.

예시코드



Calculation이라는 클래스를 먼저 만들어줬습니다.

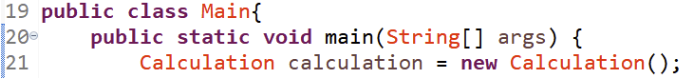
그리고 필드로 a, b를 만들어서 값을 초기화 시켜줬습니다.

그 후에 메소드를 만들었습니다.

저는 리턴은 없는 void 형식으로 했고요 메소드이름은 add로 했습니다.

그리고 메소드가 동작할 기능을 { }안에 넣어줬습니다.

이렇게 아래도 같은 방식으로 만들었습니다.



그러고 나서 메인 클래스와 메인메소드를 만들어서 실행을 해봐야겠죠??

그리고 만든 클래스를 사용하려면 객체를 생성해 줘야 합니다.

객체 생성은 많이 해보셨을 겁니다. 21번 줄이 객체 생성이고요.

검은색, 걸린, 빨간색, 하얀색이(가) 표시된 사진

자동 생성된 설명

객체를 생성하셨다면 클래스의 메소드를 사용할 수 있는 것입니다.

사용법은 "."을 찍고 메소드명을 적어주시면 됩니다.

목재의, 도시, 하얀색, 거리이(가) 표시된 사진

자동 생성된 설명

아래와 같은 결과를 볼 수 있습니다.

제가 코드를 짤 때 메소드를 void add(){ } 이런 식으로 썼는데

만약 static void add(){ }라고 쓰면 어떨까요??

스크린샷이(가) 표시된 사진

자동 생성된 설명

그러면 객체를 생성하지 않고, 클래스 이름에 "."을 찍어 사용할 수 있습니다. static은 공유의 개념으로

이해하시면 됩니다.

**생성자란**

생성자란 멤버변수를 초기화해주는 용도로 쓰입니다.

특징으로는 클래스의 이름과 같아야 합니다. 메소드와 차이점은 메소드는 우리가 만들고 싶은 대로

변수를 선언해서 만들었는데요 생성자는 무조건!!! 클래스의 이름과 동일하게 사용해야 합니다.

다음으로 생성자는 리턴타입을 갖지 않습니다.

이렇게 생성자의 특징과 쓰임을 알아봤는데요. 이제 예제를 통해서 알아보겠습니다.

스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명

그리고 생성자를 만들지 않으면 자동으로 빈 생성자를 만드는데요.

그런데 빈 생성자를 만들고 실행문이 없기 때문에 우리 눈에 보이지 않는 거겠죠.

그런데 굳이 눈에 보이게 쓰자면 public information(){ } 이런 형태일 겁니다.

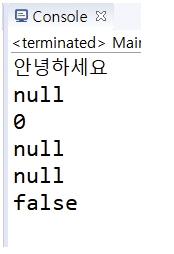
그리고 저는 그것을 또 출력문으로 보기 위해 그냥 아무 문자열을 넣어봤습니다.

그런데 생성자를 하나라도 생성해 주면 빈 생성자는 자동으로 만들어지지 않습니다.

사람이(가) 표시된 사진

자동 생성된 설명

자 이제는 생성자 없이 그냥 클래스의 객체를 만들어서 그것을 출력해봤습니다.



이런 결과가 나오네요. 그 이유는 제가 처음에 생성자에 대해 말할 때 쓰는 목적을

멤버변수들을 초기화해주기 위해서라고 했습니다.

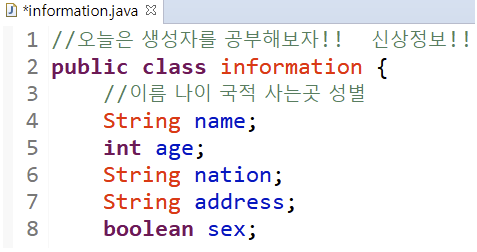
바로 빈 생성자가 값을 초기화해주기 때문입니다. 빈 생성자를 위에처럼 만들거나 만들지 않아도 자동으로 생성되니 사용자가 생성자를 만들어 값을 넣어주기 전에는 무조건 빈 생성자가 초기치로 초기화해준다는 것입니다.

위에 보시는 것과 같이 수치형 변수는(int, double, float...)은 0으로 참조형 변수는 null로 boolean형은 false로

초기화를 해주는 것입니다.

​

그럼 이제 생성자를 만들어 보겠습니다.



위에서 했던 것과 같이 필드를 선언해 줬습니다.

스크린샷, 나이프이(가) 표시된 사진

자동 생성된 설명바로 이게 생성자입니다. 클래스 이름과 동일하고 변수의 타입과 변수명을 적어주시면 됩니다.

제가 여기서 변수명을 같게 써서 헷갈리실 수도 있는데 위에 name과 아래 name은 다른 것입니다.

( ) 안에 있는 것은 지역변수로 다른 것입니다.

그런데 서로 다른 변수가 이름이 같으니 컴퓨터 입장에선 어떤 게 어떤 건지 모르겠죠??

그래서 this라는 녀석을 써준 것입니다. this를 써준 name은 필드의 name을 지칭하고 있는 겁니다.

헷갈리신다면 ( ) 안에 String name을 String nm 이런 식으로 바꾸셔도 상관은 없습니다.

**오버라이드**

부모의 함수를 그대로 갖다가 쓰기 때문에 자식 클래스에서 메소드를 다르게 쓰고 싶을 때는 메소드 오버라이딩(재정의)을 해야 합니다.

​

조류이(가) 표시된 사진

자동 생성된 설명

나이프, 테이블이(가) 표시된 사진

자동 생성된 설명

기본데이터와 참조타입

기본타입은 byte short char int long float double boolean

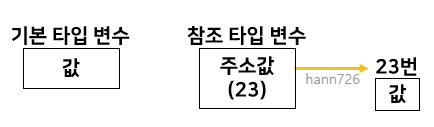
참조는 String 타입, 문자열 타입임 이거 외에도 배열, 열거, 클래스, 인터페이스 타입이 있음.

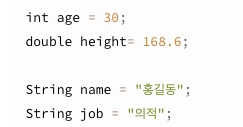
기본타입과 참조타입의 차이점은

기본타임은 변수에 값을 저장하지만

참조타입은 값이 아닌 값이 저장된 메모리의 주소를 값으로 받음

‘메모리의 주소를 참조하여 값을 불러온다.’ 라는 의미





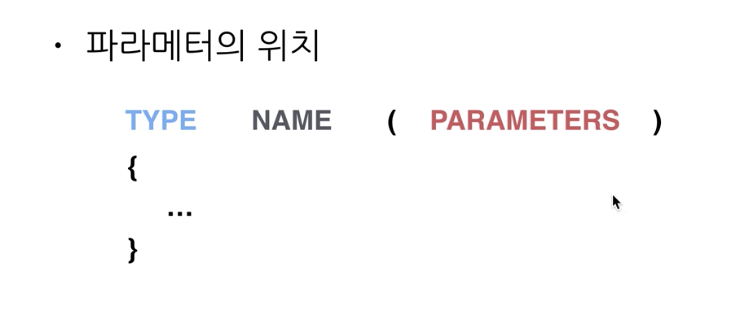
텍스트이(가) 표시된 사진

자동 생성된 설명

파라미터란

나이프이(가) 표시된 사진

자동 생성된 설명



나이프, 테이블이(가) 표시된 사진

자동 생성된 설명

나이프, 테이블이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명