5. 객체 지향 프로그램의 본질

목표 : 클래스와 객체의 개념을 알고 구분지어 보는 것

1. 메소드 프로그램의 한계

- 메소드를 활용하면 로직의 재사용이 가능하여 개발을 효율적으로 할 수는 있음 하지만 메소드만으로는 많은 양의 로직을 처리하는게 힘들고, 메소드가 너무 많아질 경우 추후 유지보수 하는게 어려워짐

2. 객체의 개념과 클래스

- 객체는 같은 종류의 데이터와 메소드가 함께 있는 구성체

- 클래스는 객체를 만들기 위한 설계도

- 객체는 클래스로부터 메모리에 생성이 됨

6. 클래스의 기초적인 코딩방법

클래스 만드는 방법

```
Public class ExClass{

        Private 자료형 인스턴스변수(속성,필드)이름;

        Public ExClass(){}

        Public method(){}

}
```

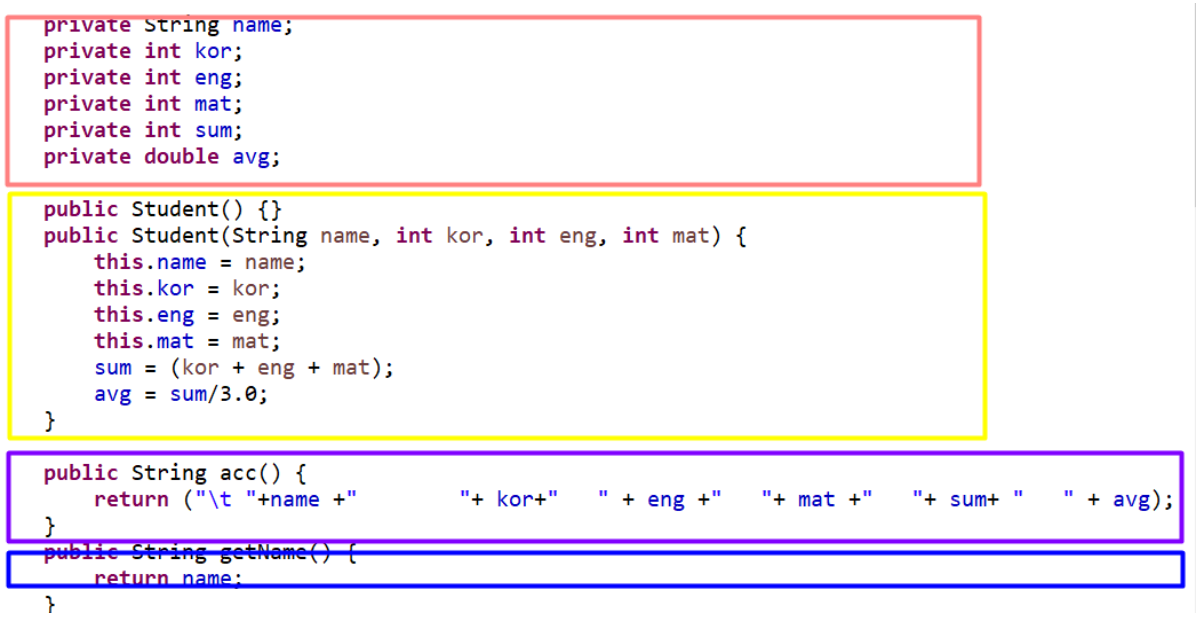
데이터(인스턴스변수,멤버변수,필드) : 이 데이터는 생성자나 setter를 이용해서 초기화하지 않으면 객체는 null, 숫자는 0 ,boolean은 false로 초기화 되어 들어간다.

생성자함수 : 클래스명과 똑같이 리턴타입이 없는 메소를 생성자라 하며 처음 클래스형 객체를 만들 때 호출된다. 모든 클래스는 반드시 하나 이상의 생성자가 있어야 한다. 만약 하나도 없으면 디폴트로 디폴트 생성자를 만들어 준다

*

메소드

Getter,setter



빨간 박스 : 클래스에 선언해놓은 변수 값

노란 박스 : 생성자 함수. 위에 선언해놓은 변수값으로 생성자 함수를 만든 후 메인에서 객체를 만들고 객체생성 시 변수를 입력한 값이 들어감

보라 박스 : 함수를 만들어 놓은 곳. 메인에서 입력한 변수값을 어떻게 출력할지 만들어 놓은 메소드함수

파란 박스 : 게터