

객체(Object)란?

- “의사나 행위가 미치는 대상”
 - 구체적, 추상적 데이터 단위
- ex) 학생 버스 학교 등...

클래스란?

- 객체에 대한 속성과 기능을 코드로 구현한 것

클래스 정의 방법

```
(접근제어자) class 클래스 이름 {  
    멤버 변수;    -> 속성  
    메서드;       -> 기능, 행위  
}
```

String은 문자열 클래스이다.

하나의 자바 파일에는 하나의 클래스를 두는 것이 원칙이나, 여러개의 클래스가 같이 있는 경우 public 클래스는 단 하나이며, public 클래스와 자바 파일의 이름은 동일해야함

자바의 모든 코드는 class 내부에 위치

이름 짓는 방법

- 패키지의 이름은 소문자
- 클래스의 이름은 대문자로 시작해서 단어가 바뀔 때 마다 소문자
- 변수이름은 소문자로 시작해서 단어가 바뀔 때 마다 대문자
- 메서드의 이름은 소문자로 시작해서 단어가 바뀔 때 마다 대문자

클래스의 진짜 이름은 => 패키지 이름 .클래스이름

패키지를 만들면 컴퓨터 로컬 디스크에 디렉토리 형태로 만들어진단

패키지는 소스코드를 어떤 계층 구조로 관리할지 구성하는 것이다.

메서드

- 함수의 일종 -> 객체의 기능을 제공하기 위해 (클래스안에 있는 멤버변수를 이용) **클래스 내부에 구현되는 함수**
- 하나의 기능을 수행**하는 일련의 코드
- 중복되는 기능은 함수**로 구현하여 함수를 호출하여 사용함

함수(메서드) 정의하기

```
int addNum (int num1, int num2) {  
    int result;  
    result = num1 + num2;  
    return result;  
}
```

add-> 함수이름

int num1, int num2 -> 매개변수: 함수의 수행을 위해 필요한 변수

return-> return 함수: 함수 수행 결과를 반환하기 위한 예약어

함수 반환형(add앞에있는 int): 반환값의 자료형을 나타냄 / 반환값이 없는 경우 void 라고 씀

함수가 호출될 때 함수가 사용하는 메모리 = Stack 메모리

main() -> addNum() 호출

생성(인스턴스화)

클래스 (Static 코드) -----> 인스턴스 dynamic memory) => 메모리에 올라간 객체

클래스 생성하기

클래스를 사용하기 위해서는 클래스를 생성하여야 함

new 예약어를 이용하여 클래스 생성

클래스형 변수이름 = new 생성자;

Student stdentA = new Student(); ==> 하나의 객체(클래스)를 생성한 것임

int I = 10;

vs

Student studentLee = new Student(); ==> 인스턴스화 하기

여기서 studentLee는 힙메모리에 생성된 Student객체의 주소값을 가진다

studentLee ==> 참조변수

참조변수를 이용해서 Student 클래스에 있는 멤버변수나 매서드를 사용할 수 있다.

앞에 Student는 객체 타입이다. 이 안에는 매서드나 속성을 정의 할 수 있고 개를 생성해서 쓰는 것이다. 객체는 반드시 생성해서 쓴다.

Student라는 자료형(객체타입)에 의해서 studentLee라는 변수를 선언

studentLee는 Student(객체)의 주소 값을 가진다

studentLee와 studentKim은 각각의 독립된 메모리 값을 가진다

Heap 메모리는 new 라는 키워드에 의해서 생성됨

하나의 클래스 코드로부터 여러 개의 **인스턴스**를 생성
Heap 메모리에 생성됨
각각의 인스턴스는 다른 메모리에 다른 값을 가짐

Stack 메모리는 지역변수들이 있는 곳

```
public static void main(String[] args) { // 메인함수지만 메서드는 아니다!!
```

```
    Student studentLee = new Student();  
    studentLee.studentName = "김진호";  
    studentLee.studentId = 100;  
    studentLee.address = "서울시 동작구 상도동";
```

```
    Student studentKim = new Student(); //  
    studentKim.studentName = "김유신";  
    studentKim.studentId = 101;  
    studentKim.address = "서울시 용산구 이태원";
```

```
    studentLee.showStudentInfo();  
    studentKim.showStudentInfo();
```

```
}
```

객체 -> 객체 지향 프로그래밍의 대상, 생성된 인스턴스

클래스 -> 객체를 프로그래밍하기 위해 코드로 만든 상태

인스턴스 -> 클래스가 메모리에 생성된 상태

멤버 변수 -> 클래스의 속성, 특성

메서드 -> 멤버 변수를 이용하여 클래스의 기능을 구현

참조 변수 -> 메모리에 생성된 인스턴스를 가리키는 변수

참조 값 -> 생성된 인스턴스의 메모리 주소값

생성자(constructor)

인스턴스 생성시 new 키워드와 함께 사용했던 생성자

생성자는 반환 타입이 없다

생성자는 인스턴스를 초기화 할 때의 명령어 집합

생성자의 이름은 그 클래스의 이름과 같음

생성자는 메소드가 아님. 상속되지 않으며, 리턴 값은 없음

디폴트 생성자

하나의 클래스에는 반드시 적어도 하나 이상의 생성자가 존재

프로그래머가 생성자를 기술하지 않으면 Default생성자가 자동으로 생김(컴파일러가 코드에 넣어줌)

디폴트 생성자는 매개변수가 없다

디폴트 생성자는 구현부가없다

만약 클래스에 매개변수가 있는 생성자를 추가하면 디폴트 생성자는 제공되지 않음

필요에 의해 생성자를 추가 하는 경우 여러 개의 생성자가 하나의 클래스에 있음(오버로드)