리터럴(Literal)은 자바에서 기본적으로 부여하는 타입(정수는 int, 실수는 double…..)

그래서 float은 실수임에도 불구하고 그냥 저장할려면 에러가 발생

Float f = 12.34 (error!)

해결할려면 타입캐스팅을 하거나 끝에다가 F를 붙여준다.(Long은 L)

정수 10진수 0x\*\*\*\* 16진수

객체는 특성과 메서드로 이루어진 객체지향프로그래밍(OOP)의 핵심요소로서 아래와 같이 구성됨

|  |
| --- |
| 클래스 이름(객체이름)  (ex) 자동차) |
| 속성들  (ex) 이름, 색상,속도…) |
| 기능들  (ex) 속도 업, 다운) |

실제 코드화

Public class Car{

String name;

String color;

Int speed;

Public Car(){}

Public void speedup();

Public void speeddown();

객체 생성 방법 Car c1 = new Car(); 🡨 메모리의 주소시작값이 들어감

이렇게 생성된 객체변수는 인스턴스라 한다.(정의한 것은 클래스)

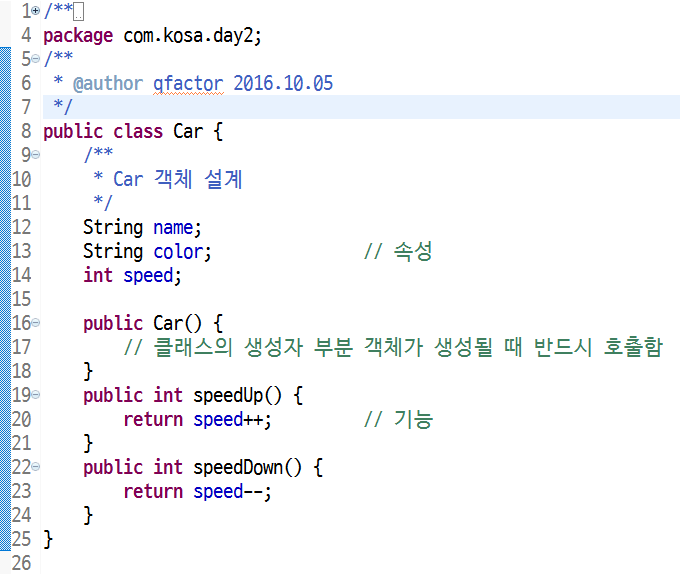
**\*기본 데이터 타입 8개 외의 단어를 생성할 시에는 모두 메모리의 주소값이 들어감\***

` 프로젝트 수주 -> 객체 모델링 -> 모델링 도식화 -> 구현 -> 완성

(설 계) (코딩)

객체 작업

1. 설계 (클래스)
2. 생성 (인스턴스)
3. 사용

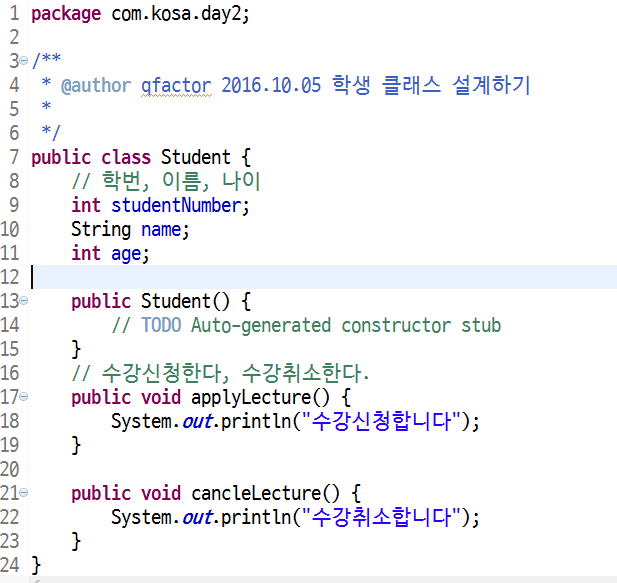


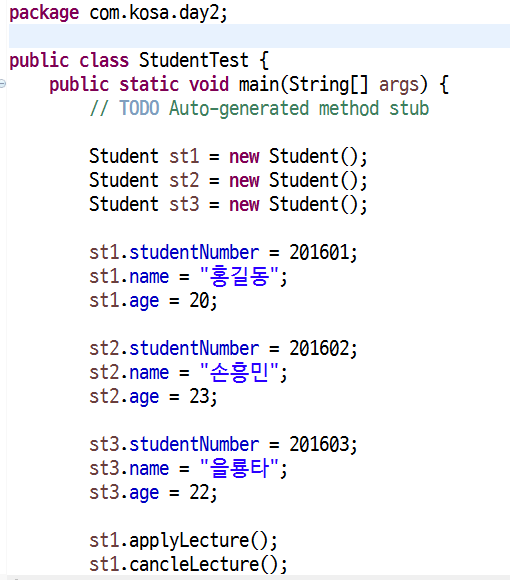
클래스 설계 연습

객체 이름 : 학생

객체 속성 : 학번, 이름 나이

객체 기능 : 수강신청한다, 수강취소한다.





변수의 범위

1. 멤버변수

* 클래스 멤버로 선언(heap 메모리에 생성)(이 메모리는 임의접근 불가)
* Heap 메모리
* 객체 생성 시 = new
* 참조가 끊긴 후에 가비지 컬렉터에 의해 삭제
* 자동 초기화됨 (기본데이터 0, false 참조데이터 null)

1. 지역번수

* 메서드 내에서 선언(stack 메모리에 생성)(개발자가 유일하게 접근 가능한 메모리)
* Stack 메모리
* 메서드 실행 시 생성
* 메서드 종료 시 소멸

Ex) GoodsStack obj = new GoodsStack();

|  |  |
| --- | --- |
| 지역변수 | 멤버변수 |
| Stack | Heap |
| Obj | GoodsStack 클래스와 멤버변수, 메서드들 |

메소드(Method) : 실행문을 가지고 있는 단위

메서드의 형식

**Public void methodType(int num)**

**접근제어자 리턴형식 이름 매개변수(인자)**

일반적으로 메서드의 이름은 소문자로 시작하고

단어시작글자는 대문자로 하는 카멜(낙타)표기법을 쓴다.

메서드 호출 = 메서드가 가지고 있는 명령문들을 실행시키는 것

Ex) c1.speedUp();

\*생성자 : 객체가 생성될 때 호출되는 메서드

1. 메서드 이름 = 클래스이름

2. 리턴타입이 존재하지 않음

3. 초기화 기능 담당

Ex) Car class의 메모리 동작

|  |  |
| --- | --- |
| Stack | Heap |
| New Car(“s60”, “white”);  ->This.name = name;  This.color = color; | Name = “s60”  Color = “white” |
| Car c3, Car() | Car name, color, speed…. |
| Car c2, Car() | Car name, color, speed…. |
| Car c1, Car() | Car name, color speed… |
| Main args |  |

This : 자신의 주소값(보통 객체 자신)

그래서 매개변수와 멤버변수의 이름이 같아도 This를 이용하면 같이 쓸 수 있다.

Ex) this.name = name;

디폴트(기본)생성자 : 생성자가 없어도 알아서 생성함

접근자 – 클래스, 변수, 메서드에 붙여서 접근권한을 지정할 수 있다.

* Public, Protected, (default), Private -> 순으로 폐쇄적이다.

위치는 항상 맨 왼쪽(앞) 이다.

Private – 내부에서만 사용 가능함(같은 클래스)

(default) - 같은 패키지

Protected - 같은 패키지 or 상속받은 클래스들

Public - 모두 접근가능