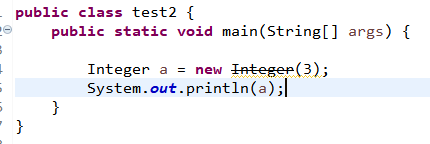
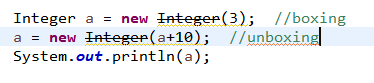
**Wrapper class**

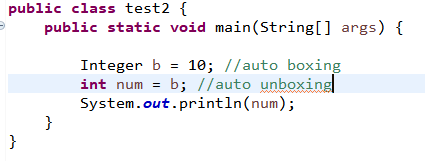
- 기본 자료형 데이트들을 인스턴스로 표현할 때



Byte, Integer, Long, Double 등 특정 기본형 wrapper class가 존재

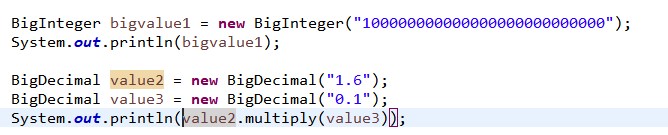


Boxing은 인스턴스 생성, unboxing은 데이터의 호출



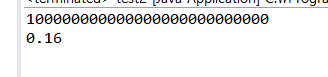
자동으로 처리해줌

**BigInteger, BigDecimal class와 난수 생성, Token**

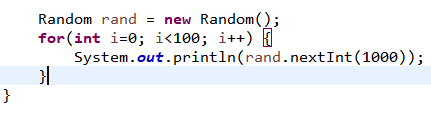


BigInteger는 표현 못하는 큰 수를 표현하기 위함

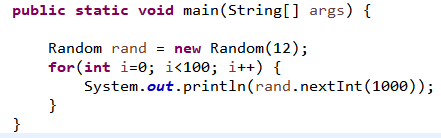
BigDecimal은 정확한 오차를 위해 확인 // 결과 값 책이랑 다르게 나옴



===========================Random================================



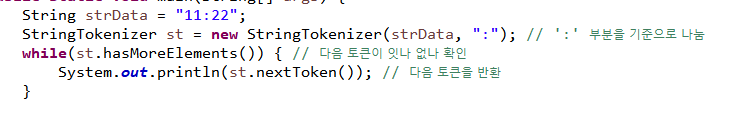
기본 Random()을 사용하면 가우시안 분포를 이용해 난수 생성



Seed 랜덤을 이용하면 12를 기준으로 컴파일을 해도 똑같은 난수가 생성

rand.setSeed(System.*currentTimeMillis*()); 사용하여 씨앗을 현재 시간을 이용하여 반환

=========================Token===================================



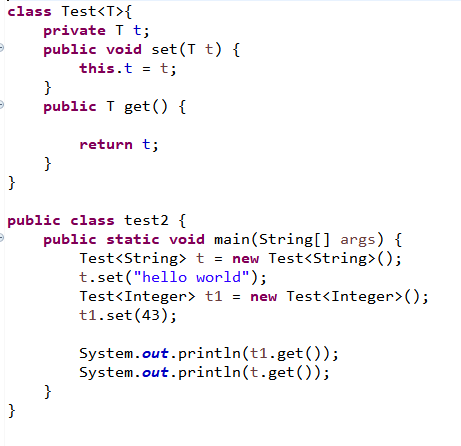
StringTokenizer을 사용하여 토큰들을 나누어서 정의 되어 있는 함수들을 사용하면 된다.

**제네릭클래스**

-자료형의 일반화

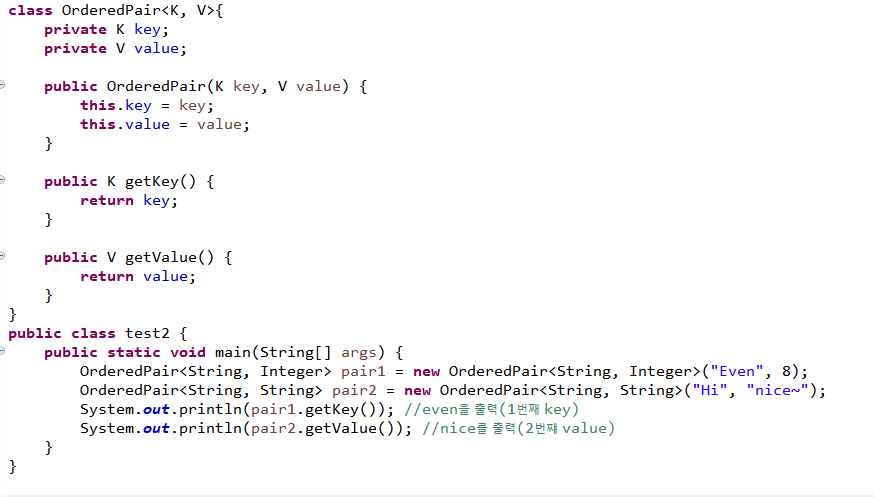
-클래스 내부에서 사용할 데이터타입을 인스턴스를 생성할 때 확정하기 위함

기본의 방법 : Object 타입으로 객체를 받아서 다형성을 이용, 객체 반환 시 원하는 타입으로 캐스팅 해야하는 불편함 존재



객체를 만들 때 자료형을 자신이 선언하면서 할 수 있다.

=========================다중타입매소드=============================



타입에 상관 없이 K or V냐에 따라 값이 달라진다

제네릭클래스는 여러 개의 매개타입을 가진다

E – element

K – key

N – number

T – type

V – value

S, U, V 등 – 2,3,4번째 타입