Git

branch

프로젝트 기준 코드 : main 브랜치

여러 개발자가 서로 다른 버전의 코드를 만들 때 서로의 영향을

주고 받지 않기 위해 필요

충돌 발생

각 브랜치가 동시에 하나의 파일을 수정하는 경우 발생

Conflict

병합하면서 생기는 충돌

병합 실패한 코드에서 내용 확인

<<<<HEAD : 충돌 발생 부분 시작 표시

>>>>zezn : 충돌 발생 끝 표시  
 ==== : 경계 표시

java

생성자(constructor)

class\_ (name)(argument){}

생성자는 반환값이 없고(void 안 씀), 클래스 이름과 동일함

객체를 생성할 때 new라는 키워드와 함께 사용

역할

객체가 생성될 때 변수나 상수를 고기화

다른 초기화 기능을 수행하는 메서드를 호출

인스턴스가 생성될 때 마다, 호출되는 ‘인스턴스 초기화 메서드’

대부분의 생성자는 외부에서 접근 가능하지만, 필요에 의해 Private선언

하기도 함

기본 생성자(default constructor)

클래스 생성자가 하나도 없는 경우 컴파일러 생성자 코드를 넣어줌

클래스에 따로 생성자를 구현할 경우 제공되지 않음

ex) public Student(){}

매개변수와 구현부가 없음

생성자 오버로딩

매개변수가 있음

컴파일러가 제공해 주는 기본 생성자 외에 필요에 의해 생성자를 직접 구현할 수 있음

하나의 클래스에 여러 개의 생성자 사용할 수 있음(오버로딩 개념)

this()

생성자에서 다른 생성자를 호출

같은 클래스 내에서 다른 생성자 호출할 때 사용해야 함

다른 생성자 호출은 생성자의 첫 문장에 사용해야 함

this.와 다른 개념

this. 는 인스턴스 자신을 가리키는 참조변수

인스턴스의 주소가 저장되어 있음

ex)

public Ex(){

this(매개변수값);

인스턴스의 복제

public Copy(Copy copy){ 매개변수가 참조형 변수인 오버로딩

this.a = copy.a  
 this.b = copy.b ….

}

변수의 자료형

기본 자료형

int, float, double …

참조 자료형

String, Student, Date …

참조 자료형은 클래스에 따라 다름

매개변수에 참조형변수을 받는 경우

public int add(A a) :

A클래스의 변수값을 a라고 이름 붙이고 그 값을 참조

인스턴스(객체) : new 키워드를 사용하여 클래스를 메모리에 생성한 상태

참조 변수 : 메모리에 생성된 인스턴스를 가리키는 변수

참조 값 : 생성된 인스턴스의 메모리 주소 값