Git

스테이징 영역에서 파일 수정하면 trcaked, modified로 commit 불가

work directory => staging area => local repository

git add -> tracked, unmodified O

tracked, modified<-(수정) X

커밋 로그 조회

git log

git log -p

patch

- 파일단위 변경내용 보여줌

git log -(숫자)

- 최근 몇 개의 커밋을 보여줄지 지정함

git log --status

status

- 각 커밋의 통계 정보를 볼 수 있음

- 파일의 수전, 변경, 삭제등을 볼 수 있음

git log --pretty=oneline

- 각 커밋을 한 줄로 보여줌

--graph : 여러 브랜치를 생성하고 병합하는 작업을 가시적으로 보여줌

git add 명령으로 스테이징 영역에 파일 추가됨

stage : commit으로 기록할 수 있는 최종 상태

- tracked, unmodified 상태

unstage : 스테이징 영역에 있지만, 파일에 변화가 있는 상태

- commit 으로 기록할 수 있는 상태가 아님

새로운 브랜치 생성 및 이동

- 장소에 브랜치 추가 : git branch (이름)

- 작업중인 브랜치 변경 : git checkout (브랜치이름)

객체지향 프로그래밍

객체(object)

우리가 표현할 수 있는 실세계의 모든 사물  
 객체지향 프로그래밍의 구현

객체를 정의 => 각 객체가 제공하는 기능들을 구현

=> 객체의 기능들간의 소통(메세지 전달)을 통해 객체간 협력 구현

클래스(Class)

객체와 관련된 데이터와 처리 동작을 한데 모은 코드 뭉치

클래스는 객체를 프로그래밍에서 사용하려고 추상화해서 기술해놓은 설계도임.

설계도대로 실체화 된 것: 객체(object)

객체를 생성한다, 인스턴스화 한다고 표현

추상화(Abstraction)

객체에서 특징적인 속성과 행위를 추출하는 과정

명사적인 특징을 뽑아냄 => 멤버변수(필드), member variable 추출

동사적인 특징을 뽑아냄 => 멤버함수(메서드) member method 추출

클래스타입 객체생성

**FieldInit fi = new FieldInit();**

참조변수 생성자

fi라는 변수르 만드는 데 변수의 자료형을 FieldInit이라는 클래스형으로 함

FieldInit 클래스의 FieldInit()생성자를 이용해서 new해서(새로 만들어서)

생성된 객체가 메모리에 적재되고 변수 fi라는 메모리주소에 대입됨

**@Override**

**public String toString() {**

**return super.toString();**

object 클래스의 toString() 오버라이딩(재정의)

this.(인스턴스) : 현재 클래스의 객체를 지정할 때 사용

super.(인스턴스): 자식 클래스가 자신을 생성할 때 부모 클래스의 생성자를 불러 초기화 할때 사용

toString() : 클래스의 멤버변수(필드)의 값들을 수시로 확인하고자 할때나,

원하는 포맷으로 멤버변수들을 출력하고자 할 때 많이 사용함