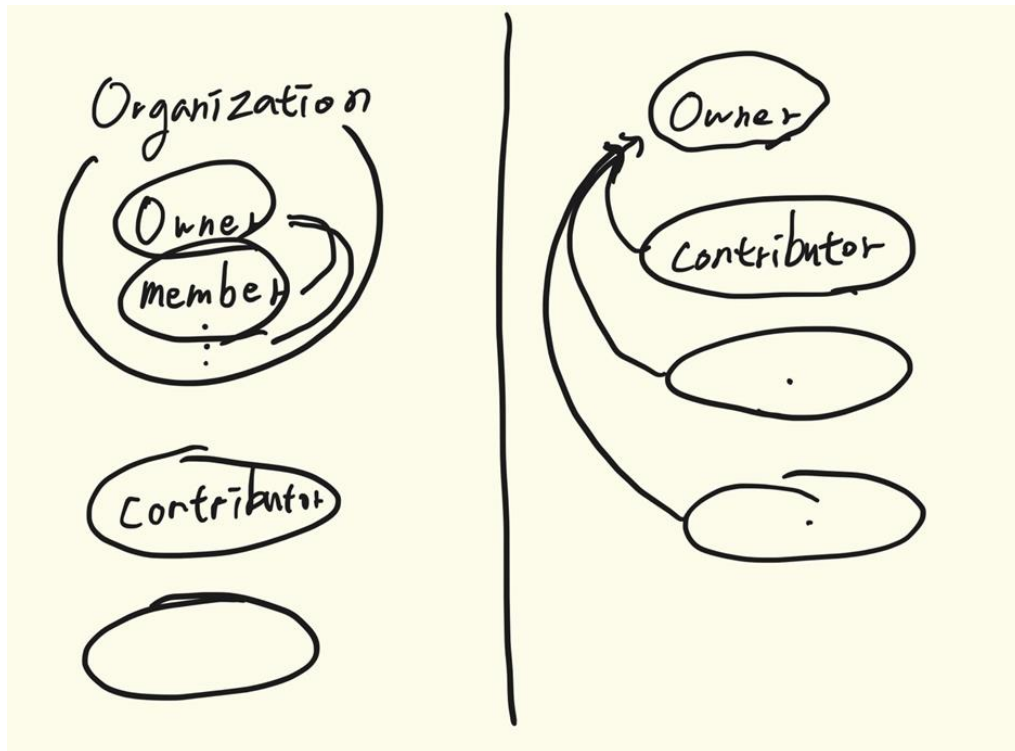


<git개념 정리>



1. Organization Account (좌측)

- 그룹을 만들고 프로젝트를 총괄하는 Owner를 제외하고도 권한을 가진 Member들을 구성해서 원격 저장소를 관리하는 방식.

주로 대기업이나 큰 프로젝트의 경우, 혹은 누구나 수정 가능한 멤버들과 함께 팀 협업을 해야 하는 경우에 Organization을 만들어 관리하게 됨

2. Personal User Account (우측)

- 개인 프로젝트를 Owner가 관리하는 방식

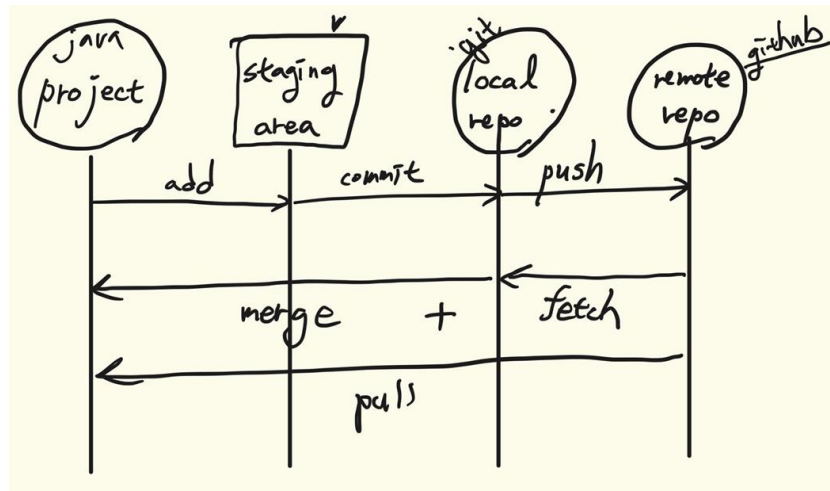
주로 소규모거나 개인이 원격 저장소를 운영하는 경우 쓰는 방식

3. Contributor

- 주로 원격 저장소의 다른 사람 프로젝트를 fork 해 와서 본인이 수정할 부분을 보고 프로젝트에 기여하고 싶어서 Contribute(기여)하는 계정들

* **clone**은 아예 프로젝트를 통째로 복사해 가서 본인의 컴퓨터에서 확인하려고 하는 거지만 그사이 원격 저장소와는 버전이 달라짐. 그래서 기여를 위해선 또 통째로 받아봐야 됨.

* **fork**는 그냥 clone하는 것과 다르게 또 하나의 branch를 가져간다고 보면 됨. 따라서 원격 저장소의 달라진 버전을 pull을 통해 다시 받아갈 수 있음.(변경 사항만 받아 올 수 있다는 얘기)



1. Java project

- Dynamic Web Project가 될 수도 있고 Spring Legacy Project가 될 수도 있고 어쨌든 Workspace 안에 있는 프로젝트

2. Staging area

- 일부만 커밋 하길 원할 때 커밋 전에 필요한 것들만 담아놓는 임시 공간

3. Local Repo(Local Repository)

- 프로젝트를 하는 프로그래머가 작업하는 컴퓨터의 저장소

4. Remote Repo(Remote Repository)

- 원격 저장소로 우리는 github가 이에 해당 됨

5. add

- 프로젝트 파일로부터 변경된 것을 Staging Area에 올리는 작업.
'+' 로 올리고 잘못 올렸으면 다시 '-' 로 Staging area에서 뺄 수 있음

6. commit

- 본인의 Local Repo로 Staging Area에 있던 것을 저장하는 작업

7. push

- Local Repo에만 있던 것을 Remote Repo로 올리는 작업

8. fetch

- 원격 저장소로부터 필요한 파일을 받음(변경 내역), 그러나 직접 로컬 branch에 반영하진 않음

9. merge

- Local Repo에 있는 변경 사항을 프로젝트에 직접 합치면서 반영하는 작업
(원격저장소에서 받은 것과 내 프로젝트 간에 경우에 따라 conflict(충돌)이 발생할 수 있음)

10. pull

- fetch와 merge를 한번에 처리할 때 두개를 붙여 pull이라고 표현함