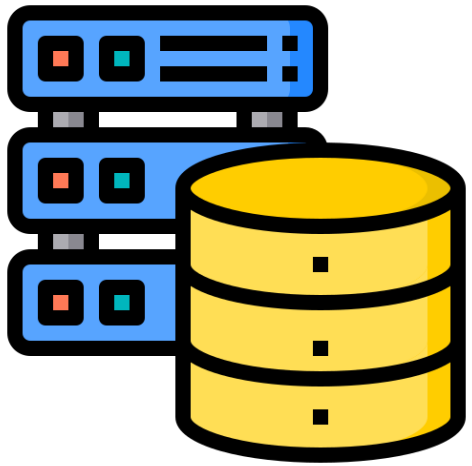
관련기술

Searution 팀

DataBase

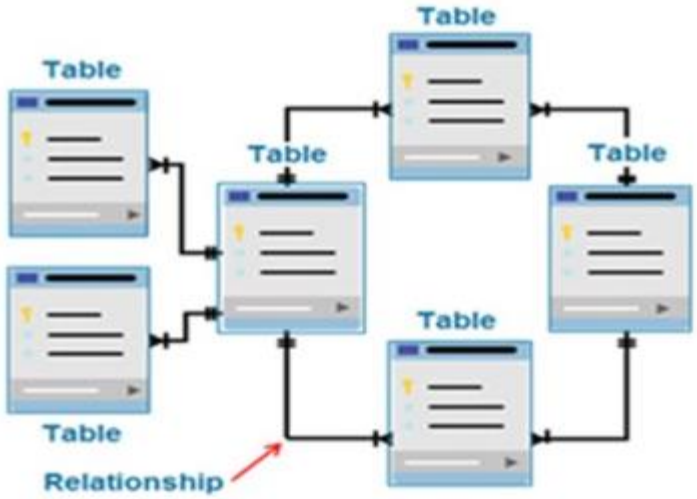
DataBase



여러 사람이 공유하여 사용할 목적으로 체계화해 통합, 관리하는 데이터의 집합

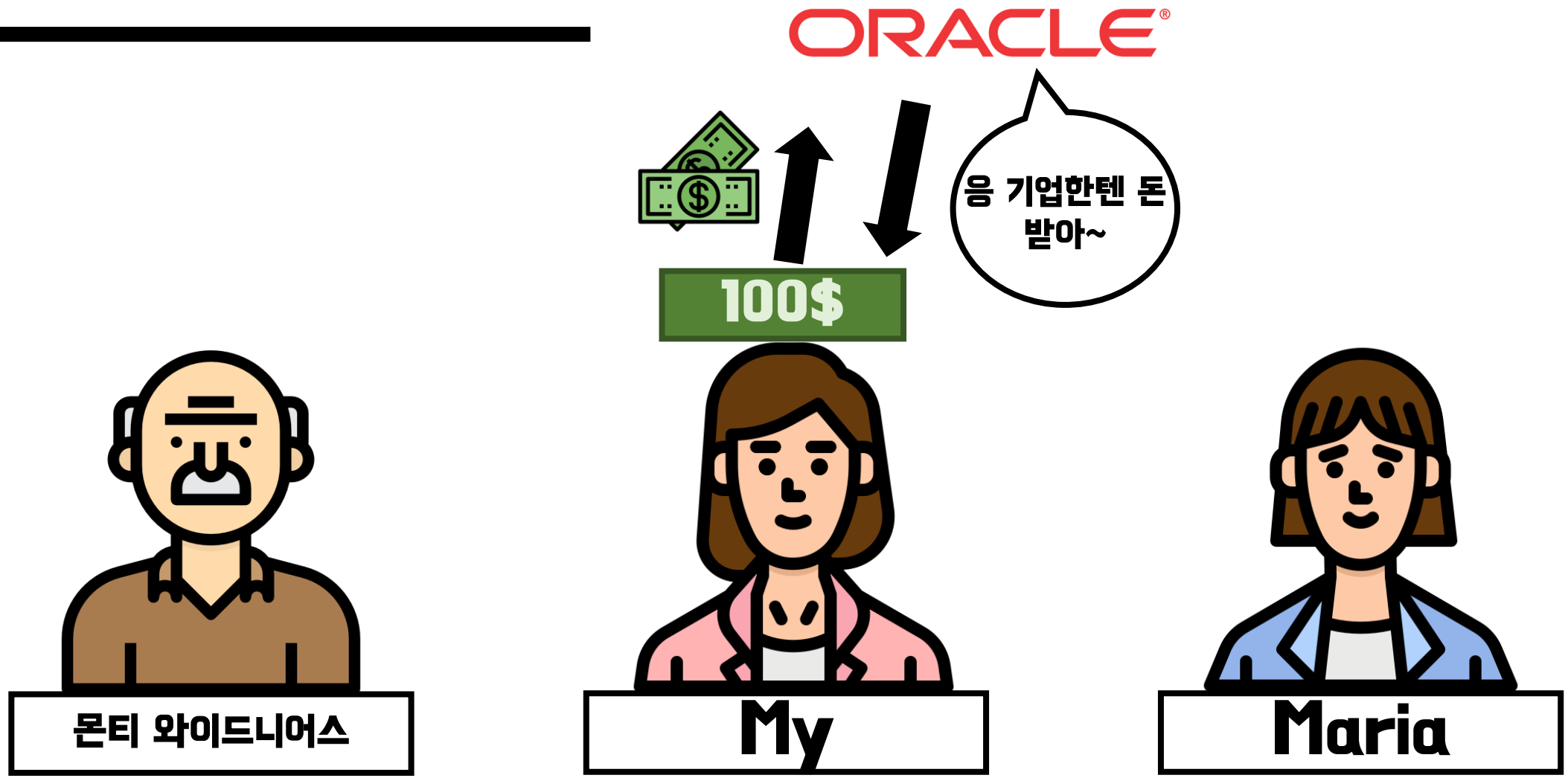
- RDB
- NoSQL
- Etc. 네트워크형, 계층형

RDBMS



관계형 데이터베이스 관리 시스템,
RDBMS는 정해져 있는 데이터 스키마에 따라 데이터베이스 테이블에 저장하며,
테이블간 연결(= 관계)을 통해 데이터를 관리함
따라서 RDBMS는 데이터 관리를 효율적으로 하기 위해 구조화가 중요함

MariaDB



MariaDB

DML	select, insert, delete, update 등	**데이터 조작어**: 데이터를 조회하거나 변형(삽입, 수정, 삭제)을 하는 명령어
DDL	create, drop, alter, rename 등	**데이터 정의어**: 테이블과 같은 데이터구조를 정의하는데 사용되는 명령어
DCL	grant, revoke	**데이터 제어어**: 데이터베이스에 접근하고 객체들을 사용하도록 권한을 주고 회수하는 명령어

DB 처리기술

Hadoop

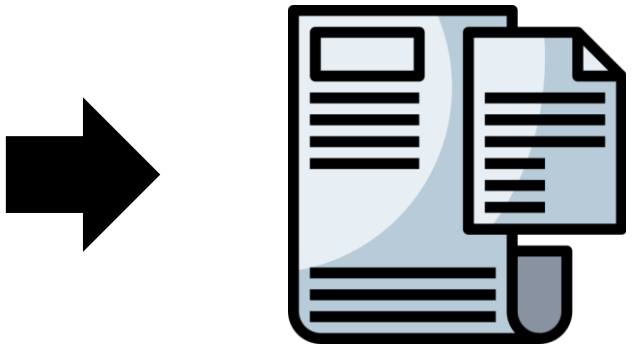
빅데이터 인프라 기술 중에 하나로 분산처리를 통해 수많은 데이터를 저장하고 처리하는 기술

- 맵-리듀스
- 분산형 파일 시스템

Map-Reduce

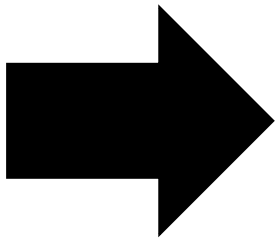
100

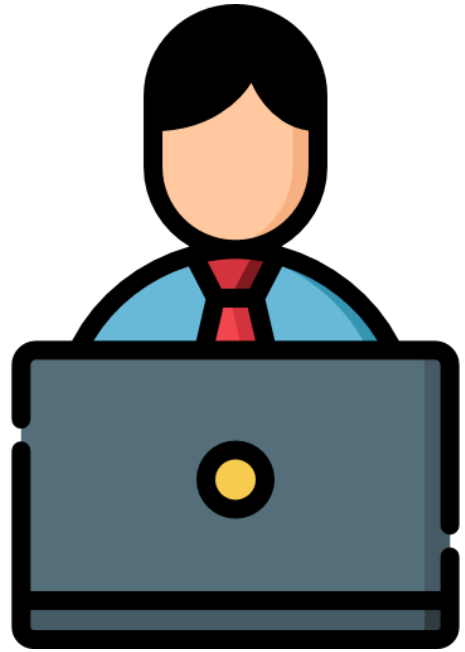
100



100

Map-Reduce



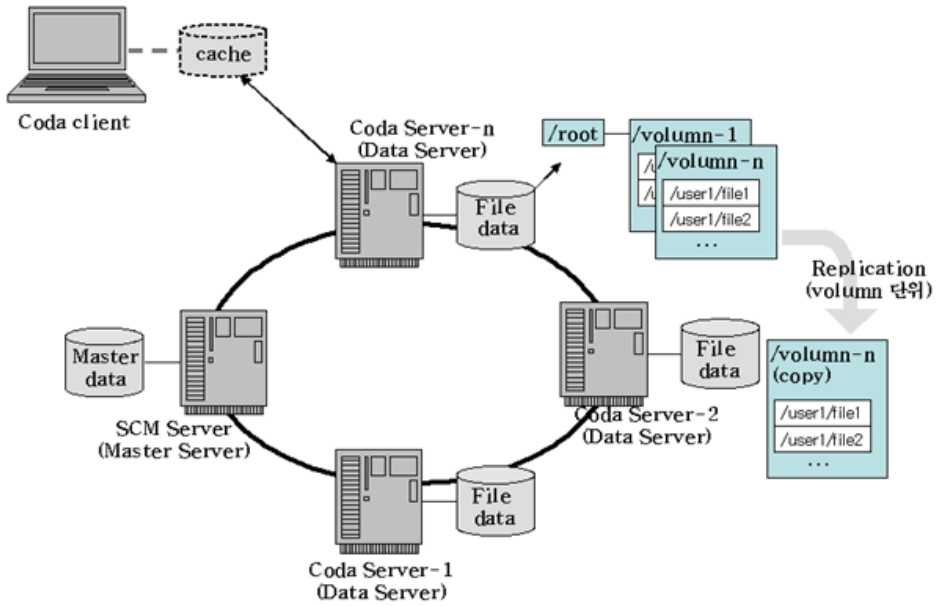


Map-Reduce

Map	흩어져 있는 데이터를 key, value로 데이터를 묶는 단계
Reduce	Map단계의 key를 중심으로 필터링 및 정렬하는 단계

map, reduce 함수를 통해서 구현할 수 있음.

분산형 파일 시스템



여러 대의 컴퓨터가 데이터를 처리하는 시스템.

하지만 데이터를 분산 처리하되 3개의 카피를 나누어 저장해두기 때문에 **한 컴퓨터에서 고장이 발생하거나 느려지더라도 데이터에 접근**할 수 있다.

Web

Framework

Frame + work

“틀” *“일”*

➡ “틀 위에서 일한다.”

소프트웨어 **어플리케이션이나 솔루션의 개발을 수월**하게 하기 위해 **소프트웨어의 구체적 기능들에 해당하는 부분의 설계와 구현을 재사용 가능**하도록 협업화된 형태로 제공하는 소프트웨어 환경

Example

언어

프레임워크

Spring

자바 플랫폼을 위한 오픈소스 애플리케이션 프레임워크로서 엔터프라이즈급 애플리케이션을 개발하기 위한 모든 기능을 종합적으로 제공하는 경량화된 솔루션

“Spirng Framework는 경량 컨테이너로 자바 객체를 담고 직접 관리함.
객체의 생성 및 소멸 그리고 라이프 사이클을 관리하며
언제든 Spring 컨테이너로 부터 필요한 객체를 가져와 사용할 수 있음.
이는 Spirng이 IOC(?) 기반의 Framework임을 의미함.“
출처: 구글링 된 어느 블로그

주요개념: 의존성

1. 의존성

```java
public class StartGame {
    public static void main(String[] args)
    {
        new MainFrame();
    }
}
```

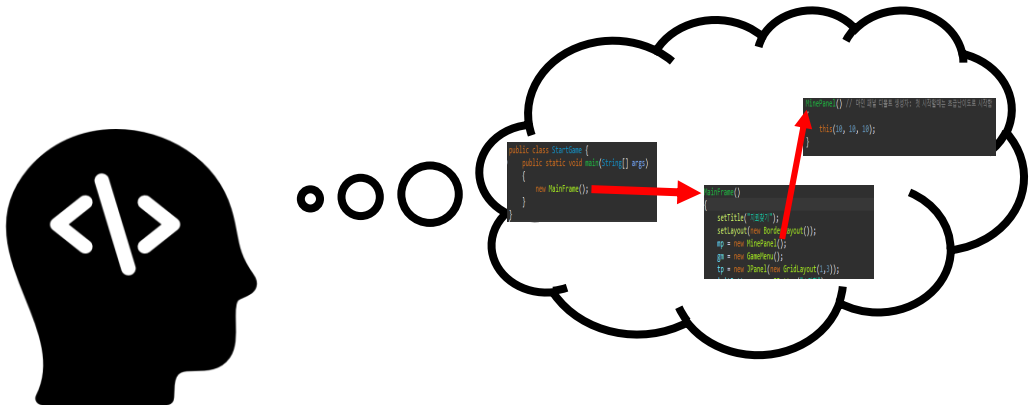
```java
MainFrame()
{
    setTitle("지뢰찾기");
    setLayout(new BorderLayout());
    mp = new MinePanel();
    gm = new GameMenu();
    tp = new JPanel(new GridLayout(1,3));
```

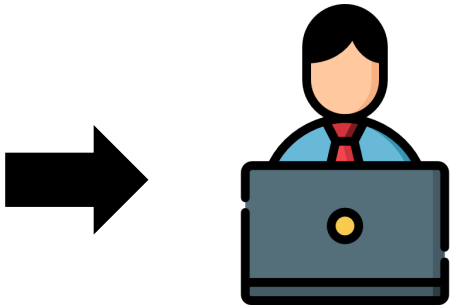
```java
MinePanel() // 마인 패널 디폴트 생성자: 첫 시작할때는 초급난이도로 시작함
{
    this(10, 10, 10);
}
```

클래스 내에서 다른 클래스의 객체를 생성하고 해당 객체내의 메서드를 호출하는 것
예) *StartGame은 MainFrame에 의존성이 있다*

주요개념: IOC(제어의 역전)

기존의 프로그래밍 방식



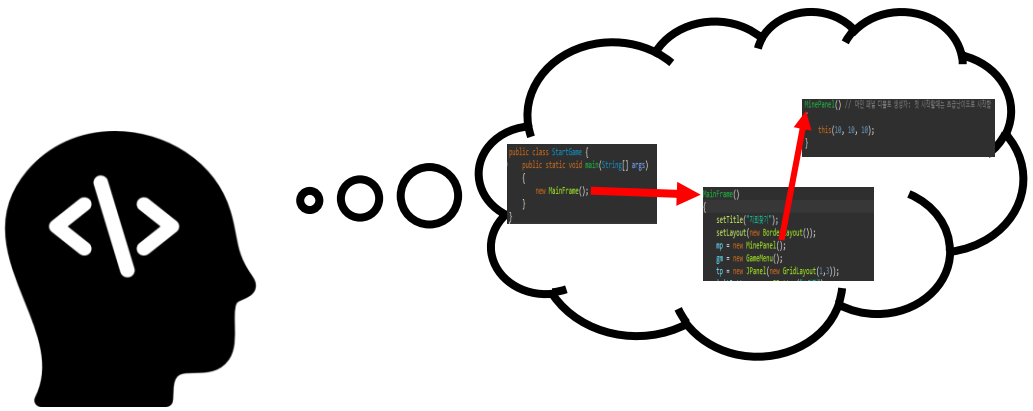


보자.. StartGame의 메인에서 MainFrame을 호출하고..

MainFrame에서는 지뢰 패널을 호출하고 게임을 시작해야겠네..

지뢰패널은 버튼객체를 갖고 있으면 될 것 같구...

주요개념: IOC(제어의 역전)



의존성을 알아서 주입해주는 짱짱 객체

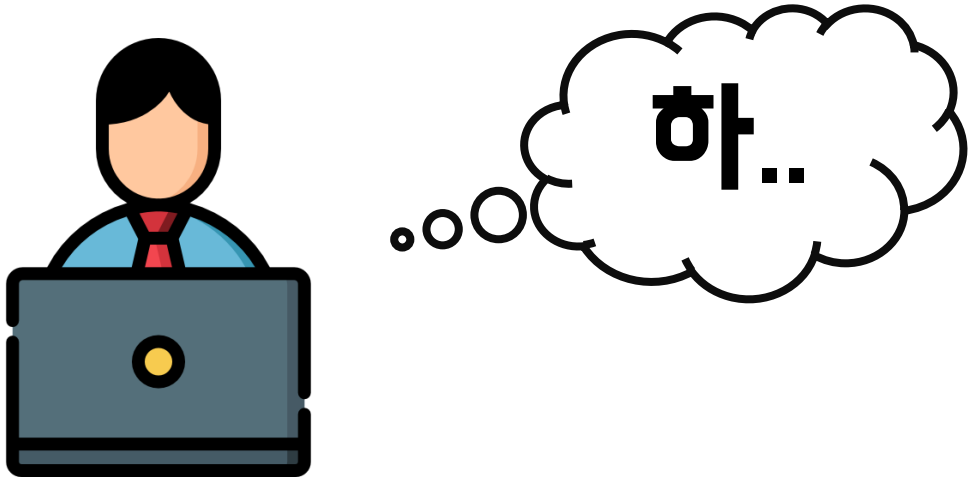
넌 StartGame 만들고

StartGame은 Main Frame 불러

MainFrame은 지뢰패널 호출해

IOC란 기존 사용자가 모든 작업을 제어하던 것을 특별한 객체에 모든 것을 위임하여
객체의 생성부터 생명주기 등 모든 객체에 대한 제어권이 넘어 간 것을 IOC. 즉, 제어의 역전 이라고 함

Spring의 단점



- **버전 관리**가 힘들다.(버전 별로 따로 해줘야 함).
- 개발자들이 개발할 때 과거에 성공적으로 사용했던 버전 및 기본 설정만을 가져다 쓰다 보니 **프로그램 모델이 경직**되어 있다. 따라서 **새롭고 효율적인 변화에 대한 도전이 부족**하다.
- **프로젝트 환경 설정에 긴 과정과 시간이 할애**된다. 따라서 관련 분야의 전문가가 필요하다.
- **전문가 의존도가 높다.**

SpringBoot

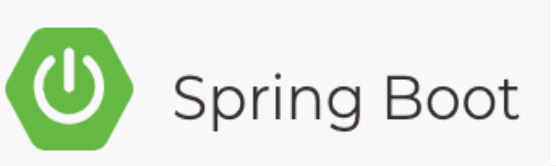
기존 스프링의 단점을 보완하고자 만든 프로젝트,
스프링 부트는 스프링 프레임 워크에 속하는 서브 프로젝트이다.

장점

자동설정

Example

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>4.2.2.RELEASE</version>
</dependency>
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.5.3</version>
</dependency>
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-validator</artifactId>
    <version>5.0.2.Final</version>
</dependency>
<dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.17</version>
</dependency>
```




```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

끝

보충 설명이 필요하시면 말로 하겠습니다.