

23.03.11 6강

복습

Thread :

Thread (Java Platform SE 7)

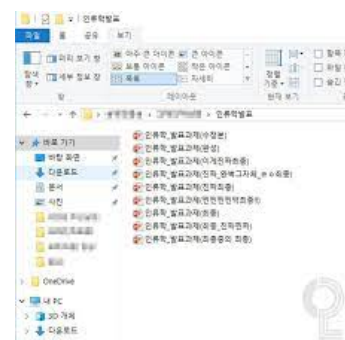
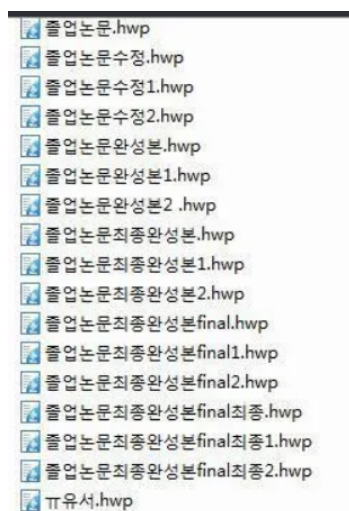
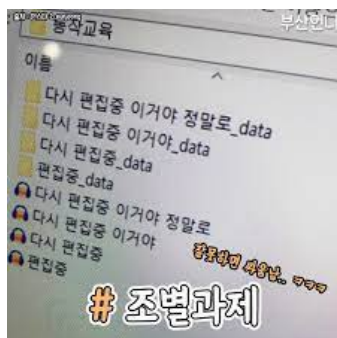
<https://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html>

버전관리

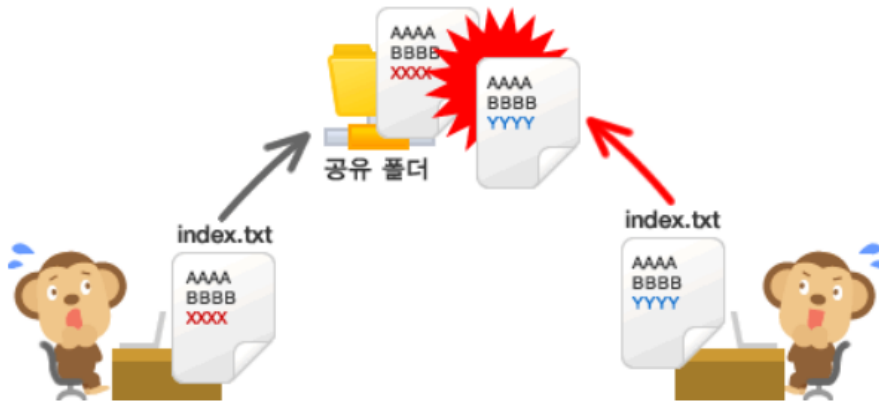
버전이란

“ 정리는 개발의 역사 ”

- 정리를 잘하려고 만든겁니다. 모든게.
- 개발자는 정리하는 방법만으로도 평생 싸우는 직업입니다.



혼동하기 쉽고 관리가 불편한 버전관리를 하지 않은 파일 관리 예시 - 찌찌찌찌종본



협업 시 서로 같은 문서를 수정하면 충돌할 수도 있음

코파일럿

<https://docs.github.com/ko/copilot>

github

rain2473 - Overview

rain2473 has 4 repositories available. Follow their code on GitHub.

<https://github.com/rain2473>

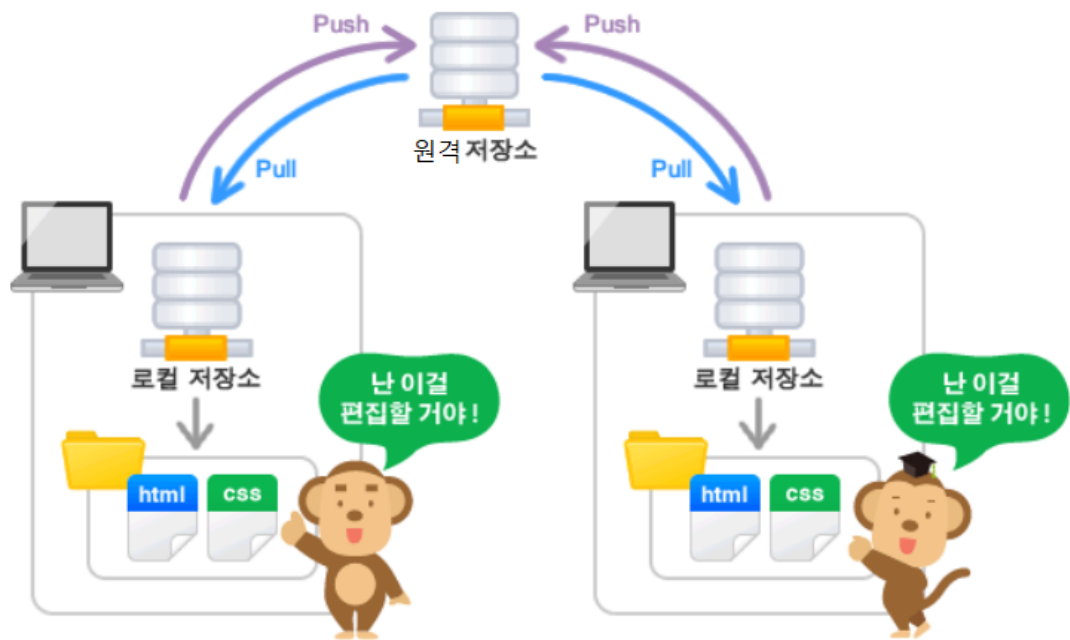
용어정리

저장소

저장소(Git repository)란 말그대로 파일이나 폴더를 저장해 두는 곳입니다.

Git 저장소가 제공하는 좋은 점 중 하나는 파일이 변경 이력 별로 구분되어 저장된다는 점입니다.

비슷한 파일이라도 실제 내용 일부 문구가 서로 다르다면 다른 파일로 인식하기 때문에 파일을 변경 사항 별로 구분해 저장할 수 있습니다.



원격저장소

물리적인 위치와 상관없이 네트워크 상에서 존재하는 저장소
저장용도로만 사용하기에 서버와는 조금 다른 개념.

github 에서 제공하는 원격저장소 : 온라인 상태라면 언제든지 접근

FTP 같은 고전적인 방식의 파일 저장소도 있음

로컬

원격저장소에 대응되는 현지의 저장소

사용자의 개발하는 pc 라고 생각하면 매우 적당.

레포지토리

git 에서 지정하는 저장소의 기본 단위.

하드디스크 드라이브와 유사, 레포지토리 안에 폴더도 있고 파일도 있음.

형상관리는 레포지트리 기반으로 관리.

레포지토리 (Repository)

git 에서 지정하는 저장소의 기본 단위 입니다.

쓰다보면 이해 되겠지만

쉽게 비슷한 개념을 알려드리자면
하드디스크 드라이브라고 생각하면 쉽습니다.
안에 폴더도 있고 파일도 있고.. 레포지토리
안에도 폴더도 있고 파일도 있습니다.

형상관리는 이 레포지트리 기반으로 관리가
됩니다.

git bash

초기설정

```
$ git config --global user.name "<사용자명>"  
$ git config --global user.email "<메일 주소>"  
$ git config --global color.ui auto  
$ git config --global alias.co checkout
```

한국어 설정

Note

Windows에서 콘솔(Git Bash)을 사용하는 경우, 한국어를 포함하는 파일명이 "\346\226\260\350\246..." 라는 형식으로 표시됩니다. 아래의 설정을 추가하면 한국어를 포함한 파일명도 올바르게 표시됩니다.

```
$ git config --global core.quotePath off
```

또한 Windows에서 콘솔을 이용하는 경우, 한국어를 입력할 수 없기 때문에 커밋 메시지에서 한국어를 사용하는 경우 '-m' 옵션 대신 외부 에디터를 이용해야 합니다.

사용하는 외부 에디터는 문자 코드 'UTF-8', 라인 피드 코드 'LF'와 호환되어야 합니다.

```
git config --global core.editor "\"[사용할 에디터의 경로]\""
```



```
MINGW64:/c/Users/admin/git_exam/tutorial
admin@DESKTOP-VAJSTK9 MINGW64 ~/git_exam
$ pwd
/c/Users/admin/git_exam

admin@DESKTOP-VAJSTK9 MINGW64 ~/git_exam
$ mkdir tutorial

admin@DESKTOP-VAJSTK9 MINGW64 ~/git_exam
$ cd tutorial

admin@DESKTOP-VAJSTK9 MINGW64 ~/git_exam/tutorial
$ pwd
/c/Users/admin/git_exam/tutorial

admin@DESKTOP-VAJSTK9 MINGW64 ~/git_exam/tutorial
$ git init
Initialized empty Git repository in C:/Users/admin/git_exam/tutorial/.git/

admin@DESKTOP-VAJSTK9 MINGW64 ~/git_exam/tutorial (master)
$ |
```

커밋

파일 및 폴더의 추가/변경 사항을 저장소에 기록하려면 '커밋'이란 버튼을 눌러줘야 합니다.

커밋 버튼을 누르면 이전 커밋 상태부터 현재 상태까지의 변경 이력이 기록된 커밋(혹은 리비전)이 만들어집니다.

커밋은 아래 그림처럼 시간순으로 저장됩니다. 최근 커밋부터 거슬러 올라가면 과거 변경 이력과 내용을 알 수 있겠죠.



Tips

버그 수정, 기능 추가 등 특별한 의미가 있는 업데이트를 작업 별로 구분해서 각각 커밋하면, 나중에 이력을 보고 특정 변경 내용을 찾기 쉽습니다.

커밋은 이렇게 이력을 남기는 중요한 작업이기 때문에 커밋 버튼을 누를땐 커밋 메시지를 필수로 입력해야 합니다. 메시지가 없으면 커밋이 실행되지 않습니다.

Tips

메시지는 명료하고 이해하기 쉽게 남겨야 본인 뿐만 아니라 다른 사람이 커밋 이력을 확인하기 쉽습니다. **Git** 에서 권장하는 메시지 형식을 따르는 것도 좋습니다.


1번째 줄 : 커밋 내의 변경 내용을 요약
2번째 줄 : 빈 칸
3번째 줄 : 변경한 이유

주로 위 형식으로 메시지를 작성합니다.



파일 커밋(Commit) 하기 【튜토리얼1 Git의 기본】 | 누구나 쉽게 이해할 수 있는 Git 입문~버전 관리를 완벽하게 이용해보자~ | Backlog

파일 커밋(Commit) 하기 【튜토리얼1 Git의 기본】 | 누구나 쉽게 알 수 있는 Git에 입문하신 것을 환영합니다. Git을 사용해 버전 관리를 할 수 있도록 함께 공부해봅시다!

 https://backlog.com/git-tutorial/kr/intro/intro2_4.html

객체지향 프로그래밍

소스코드가 너무 길어져서 개발자 자신도 파악하기 힘들어 짐
대규모 프로그램도 즐겁게 개발하기 위해
인간이 파악하기 어려운 복잡함을 극복하기 위해

생각 하는 방법 (사고)의 전환

암기가 아닌 이해와 전체상 그리는 것

절차지향 → 객체지향

객체지향 프로그래밍이란?



객체지향(Object-Oriented)은 소프트웨어 개발에서 사용되는 방법론 중 하나입니다. 이 방법론은 프로그램을 구성하는 다양한 개체(object)를 중심으로 개발을 진행하는 방식을 말합니다.



객체는 데이터와 해당 데이터를 처리하는 메서드(method)를 포함하는 소프트웨어 개체입니다. 이러한 개체들은 서로 관련된 데이터와 메서드를 묶어서 클래스(class)라는 논리적인 단위로 정의됩니다. 클래스는 객체를 생성하는 템플릿(template) 역할을 하며, 객체를 생성하면 해당 객체는 클래스에서 정의된 데이터와 메서드를 가지게 됩니다.

객체지향의 핵심 개념은 캡슐화(encapsulation), 상속성(inheritance), 다형성(polymorphism)입니다. 캡슐화는 객체의 데이터와 메서드를 외부에서 직접 접근하지 못하도록 보호하는 것을 말하며, 상속성은 부모 클래스에서 정의된 특성을 자식 클래스에서 상속받는 것을 말합니다. 다형성은 하나의 메서드나 클래스가 다양한 형태로 동작할 수 있는 능력을 말합니다.

객체지향은 소프트웨어의 유지보수성, 재사용성, 확장성을 향상시키는 등의 장점을 가지고 있어서 현재까지 많은 소프트웨어 개발에 사용되고 있습니다.

객체지향의 장점

프로그램의 수정이 용이

프로그램의 일부를 간단히 재사용 가능

현실과 객체지향

객체 (object) : 현실 세계에 존재하는 모든 것

클래스 (class) : 컴퓨터 세계에서 객체를 구체화 한 틀

객체의 예시



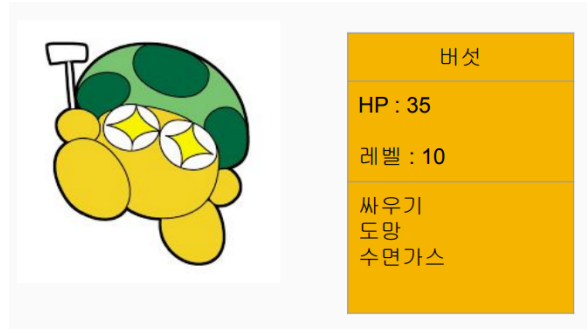
용사
이름 : 나이트
HP : 100
싸우기 도망 잠자기 앉기



통장
계좌번호 : 111-222-3333
잔액 : 10,000원
입금 출금 조회



축구선수
이름 : 메시
드리블 : 100
슛 패스 달리기



객체간의 상호작용

객체끼리의 상호작용 → 현실의 구현

이해가 용이하다 → 버그 발생 적다

객체 지향 프로그래밍의 3대 특징

캡슐화 (Encapsulation)

상속 (Inheritance)

다형성 (Polymorphism)

연습문제

1

연습문제 7-1

ATM, 발권기, 블로그 등 “현실세계의 인간의 활동을 프로그램으로 기기화, 자동화 한 것”의 예를 나열해 보라.

(힌트1) 현대에는 “기기가 하는 것이 당연한 것” 이 것 중, 옛날에는 사람이 수작업으로 했던 것들

(힌트2) ATM이나 발권기 같은 “겉 모양이 컴퓨터가 아닌 것” 중에서도 프로그램이 동작하고 있다

키오스크, 이메일, 깃허브, 엑셀, 파워포인트, 워드프로세스, 파이썬 크롤링 툴 등

2

연습문제 7-2

다음 프로그램을 작성 할 때 등장하는 객체 (현실세계의 등장인물) 에는 어떤 것이 있는지 나열하시오.

1. 현재 비행중인 모든 비행기와 공항을 관리하는 항공관제 시스템
2. 국내의 영화관을 선택하면, 그 영화관에서 상영중인 영화와 그 주연배우 목록을 표시 해 주는 프로그램
3. 남아 있는 식재료를 입력하면, 레시피 중에서 그 식재료를 사용한 요리를 검색 해 주는 프로그램

- 1) 모든 공항, 모든 비행기, 근무중인 직원, 승객, 활주로, 등등...
- 2) 모든 국내 영화관(지역, 이름, 고유번호, 상영관 수), 현재 상영중인 영화(배우, 주연, 상영시간)
- 3) 남은 식재료, 모든 음식(카테고리), 레시피(재료, 소요시간)

3

연습문제 7-3

어떤 도시의 관광안내소에는 터치 패널식의 "관광안내단말" 이 설치되어 있다. 이용자가 화면에서 희망조건을 입력하면, 추천 가게나 명소 유적의 이름, 주소, 전화번호, 해설을 제시 해 준다. 이 관광안내단말에 동작하는 프로그램의 내부에는 여러가지 오브젝트가 동작하고 있다. 아래 2가지 오브젝트가 가지고 있는 "동작", "속성"을 나열 해 보시오.

1. 현실 세계의 안내판을 재현한 "안내판" 오브젝트
2. 현실 세계의 가게나 명소 유적을 재현한 "관광지" 오브젝트

(힌트) 오브젝트에는 동작이나 속성만을 단독으로 가지고 있는 것도 있다.

안내판 - 위치, 지도, 명소, 설명, 사진, 이름 등등

관광지 - 위치, 이미지, 관광지 유형, 표시 기능 등등



1. "안내판" 오브젝트



- 동작: 안내 정보를 표시하는 기능, 사용자 입력에 따라 정보를 업데이트 하는 기능
- 속성: 안내 정보 (가게나 명소 유적의 이름, 주소, 전화번호, 해설), 위치 정보, 디자인(폰트, 색상 등)

1. "관광지" 오브젝트

- 동작: 사용자 입력에 따라 해당 관광지 정보를 표시하는 기능
- 속성: 관광지 정보 (이름, 주소, 전화번호, 해설), 이미지, 위치 정보, 관광지 유형(문화재, 자연관광지 등)

인스턴스와 클래스

각 오브젝트가 가지고 있을만한 "속성" 이나 "동작" 을 생각 해 종류와 내용을 정의

각 오브젝트를 가상세계에 도출, 동작시켜 보기

용어정리

오브젝트 (object)

현실 세계의 모든 객체

클래스 (class)

오브젝트를 가상세계 용으로 구체화 한 것 (붕어빵 틀)

인스턴스 (instance)

클래스를 활용 해 메모리 상에 만들어 낸 것 (붕어빵)

ex) 학생(object) - 학생 클래스(class) - 나(instance)

클래스명과 멤버변수명의 명명 규칙

클래스명	명사	단어의 맨 처음은 대문자	Hero, MonsterInfo
필드 명	명사	최초 이외의 단어의 맨 처음은 대문자	mLevel, mList, level, items, itemList
메소드 명	동사	최초 이외의 단어의 맨 처음은 대문자	attack, findWeakPoint

클래스 정의에 따른 효과

정의한 클래스로 인스턴스를 생성 할 수 있다

클래스로 생성한 인스턴스를 넣을 수 있는 새로운 변수의 타입이 이용 가능 해 진다

Hero 클래스를 정의하면 Hero 타입의 변수가 이용 가능
클래스를 정의하면 Java 에서 이용가능한 타입의 종류가 점점 늘어 남

연습문제

```
package game;

import java.util.Random;

public class Cleric {
    String name;

    int hp = 50;
    int mp = 10;
    int hpMax = 50;
    int mpMax = 10;

    void selfAid() {
        if (mp < 5) {
            System.out.println("mp가 부족합니다!");
        } else {
            System.out.println("selfaid를 실행합니다!\nhp가 최대로 충전 됩니다!");
            this.hp = hpMax;
            this.mp -= 5;
        }
    }

    int pray(int sec) {
        Random random = new Random();
        int mpRecovery = random.nextInt(3) + sec;
        if (mpRecovery > mpMax - this.mp) {
            mpRecovery = mpMax - this.mp;
        }
        this.mp += mpRecovery;
        System.out.println("pray를 실행합니다!\n mp가 " + mpRecovery + "만큼 회복됩니다!");
        return mpRecovery;
    }
}
```