

- 버전관리

git을 사용해봤습니다. (git은 프로젝트의 버전관리 도구입니다.)

워킹 디렉토리, 스테이지, 레포지토리.

워킹 디렉토리는 현재 작업하고 있는 프로젝트의 경로입니다. 이 프로젝트를 git에 올리기 위해 add 명령어로 스테이지에 올려줍니다. 스테이지는 git에 올리기 전에 한 번 더 점검?

사실 3학년이 돼서야 버전관리의 개념에 대해 알게 되었습니다. 소프트웨어를 개발하다 보면

- 객체지향

객체지향 설계의 개념에 대해서 설명드리겠습니다. 현실 세계에 있는 것을 객체로 나누고, 각 객체의 고유한 속성과 행동을 정의한 설계입니다.

제가 객체지향의 개념을 이해할 때 사용했던 방법으로 설명해보겠습니다.

먼저, 객체지향의 구성요소에는 크게 클래스 객체 속성 메소드가 있습니다.

비유를 하자면, 클래스는 동물, 객체는 각각의 동물이라고 볼 수 있습니다. 뽀빠라는 객체, 토토라는 객체, 초코, 미소.. 등이 있겠습니다. 그리고 모든 동물은 고유한 속성을 가집니다. 속성에는 그 동물의 종이나 다리의 개수, 성별, 이름, 나이 등이 있겠습니다. 또한 각 동물은 동일한 행동을 할 수 있습니다. 이를 메소드라고 표현하는데, 예를 들어 달리기 메소드, 밥 먹기 메소드, 말하기 메소드 등이 있습니다. 이제 우리는 원할 때마다 뽀빠라는 객체를 만들고 강아지이고 여자이고 다리는 네 개다. 라고 속성을 부여할 수 있습니다. 또 뽀빠 달리기, 뽀빠 말하기 등과 같이 메소드를 실행할 수 있습니다. 객체는 각각의 동물일 수도 있고 사람, 음식 등 현실에 있는 무엇이든 될 수 있습니다. 이러한 객체를 지향하는, 현실에 있는 것을 목표로 하는 프로그래밍을 객체 지향 프로그래밍의 정의라고 이해하고 있습니다.

추상화

여러 클래스들의 공통적인 속성과 메소드를 추출하여 정의하는 것입니다. 코드의 중복을 줄이고 필요없는 코드를 제거하여 유지보수를 용이하게 할 수 있는 장점이 있습니다. 추상 클래스나 인터페이스로 추상화를 구현하고 이를 하위 클래스에서 상속받을 수 있습니다.

둘의 차이는.. 사실 조금 모호하여 조금 더 공부할 필요가 있지만 아는 선에서 설명을..

추상 클래스는 abstract이라는 키워드를 붙여 이것을 상속받은 클래스가 반드시 구현하게 만드는 클래스입니다. 인터페이스도 일종의 추상 클래스인데 이제 조금 더 추상화가 되었다고 봅니다. 비유를 하자면 추상 클래스는 만들다 만 건축물이고 인터페이스는 정말 기본적인 틀만 잡은 설계도라고 생각합니다. 다중 상속...

상속

상위 클래스의 속성과 메소드를 하위 클래스가 물려받아서 사용하는 것입니다. 예를 들어 동물이라는 클래스가 있습니다. 하위 클래스 객체를 생성하면 상위 클래스의 생성자 함수 먼저 실행하게 되는 특징이 있습니다.

다형성

오버로딩 - 매개변수를 달리하여 이름만 같고 완전히 다른 함수

오버라이딩 - 상속받은 클래스의 메소드를 하위 클래스에서 재정의하는 것. 이름도 같고 매개

변수도 같지만 내부 코드가 다름.

캡슐화

클래스 내의 속성과 메소드를 캡슐로 싸는 것입니다. 정보를 은닉하고 필요한 부분만 노출하여 객체의 독립성을 보장하고 외부에서 접근할 수 없게 합니다. 즉, 정보를 외부로부터 보호할 수 있습니다. 객체지향 프로그래밍 언어의 대표라고 할 수 있는 Java는 접근제한자를 두어 외부 클래스에게 접근 범위를 부여합니다.

SOLID

단일책임원칙

1. 접근제어자

private : 자기 자신만

default : 같은 패키지

protected : 같은 패키지, 하위 클래스도

public : 접근 제한 없음

2. JVM

가비지 콜렉터는 사용하지 않는 객체 메모리를 주기적으로 검사하고 청소

사용하지 않는 메모리가 쌓이면 결국 메모리 누수로 이어짐

가비지 콜렉터에 의해 수행되는 프로세스를 가비지 컬렉션이라고 함

C나 C++은 free()와 같은 함수로 직접 메모리 해제

JAVA는 직접 안 해도 되지만 System.gc()로 가비지 콜렉터 호출 가능

자바로 작성한 프로그램이 어느 기기, 어느 운영체제 상에서도 실행될 수 있게 해줌

윈도우 리눅스 등등..

3. 멀티쓰레드

쓰레드는 하나의 프로그램이 실행되는 최소 단위. 멀티쓰레드는 여러 스레드를 동시에 실행시키는 응용 프로그램을 작성하는 기법. 즉 동시에 여러 작업을 할 수 있습니다. 예를 들어 하나의 데이터베이스를 가지고 여러 명의 사용자가 동시에 조회를 할 수 있습니다. 그러나 자원을 공유하는 것이기 때문에 여러 명이 동시에 동일한 검색어를 입력할 경우 충돌이 일어날 수 있습니다.

4. 정적 vs 동적

5. 리액티브

디버거

디버거는 소프트웨어 코드에서 오류를 찾아 수정하여 소프트웨어 프로그램 내부에 실시간으로 액세스할 수 있도록 합니다. 표준 디버깅 기능에는 중단점을 설정하여 프로그램을 일시 중지하고, 변수 값을 검사하고, 코드를 한 줄씩 확인하는 기능이 포함됩니다.

컴파일러 및 인터프리터

컴파일러와 인터프리터는 프로그래밍 언어로 작성된 코드를 기계 판독형 코드로 변환합니다. 컴파일러는 실행 가능한 프로그램을 생성하고 인터프리터는 프로그램을 직접 실행합니다.

트랜잭션 : 데이터베이스에서 하나의 논리적 기능을 수행하는 작업 단위입니다. 원자성, 일관성, 독립성, 영속성.

모델링

무결성

list vs map

리액트 : 페이스북이 개발한 오픈 소스 자바스크립트 라이브러리입니다. 실전에서 사용해 본 경험은 없습니다.

Node.js : 서버 측 애플리케이션 개발을 위한 자바스크립트 플랫폼입니다. vscode로 주식 데이터를 받아오고 노드제이에스 환경에서 표 형태로 주식 데이터를 확인하는 수업을 들었습니다. 실전 경험은 없습니다..

REST : 자원을 http URI로 구분하여 상태를 주고받는 방식입니다. get, post, put, delete 등의 메서드를 활용하여 메시지를 주고 받습니다.

REST API : REST 원리를 따르는 웹 서비스 통신 인터페이스입니다.

RESTful : REST의 원리를 따르는 시스템입니다.

- C#과 Java

두 언어 모두 객체 지향 프로그래밍을 지원. C#은 .NET 플랫폼에 의존적이고 Java는 플랫폼에 독립적.

- 의존성 : 한 객체가 다른 객체의 기능을 사용하기 위해 필요로 하는 관계

- DI : 애노테이션을 사용하여 의존성을 주입해주는 것. 코드의 결합도를 낮추고 테스트와 유지보수를 쉽게 함.

- AOP : 여러 관점에서 프로그램을 나누어 보는 것. 즉, 핵심 로직 외의 공통된 기능을 관점이라고 하고, 이들을 묶는 것입니다.

- PSA : 특정 기수레 종속되지 않도록 서비스 접근을 추상화. 다른 DB로 옮겨도 하나의 공통 인터페이스를 제공해서 기술 변경에 따른 코드 수정 최소화

- 사용해본 디자인패턴

- get, post의 역할과 사용 경험

get은 서버에서 정보를 조회할 때 사용하고. post는 서버에 데이터를 전달하거나 생성할 때 사용합니다. 최근에 게시판 구현할 때

- post, put의 차이

post는 서버에 데이터를 생성하거나 전달할 때 사용하고 put은 데이터를 수정할 때 사용합니다. SIDU

- 리액티브 프로그래밍

데이터의 변화에 실시간으로 반응하는 프로그래밍 패러다임입니다. 이벤트 기반의 비동기 처리를 통해 효율성을 높일 수 있습니다.

- RDBMS

관계형 데이터베이스 관리 시스템으로, 테이블 구조로 데이터를 저장하고 SQL을 사용해 관리합니다. 무결성과 일관성을 보장하여 금융, 의료 분야에서 자주 사용됩니다. 제가 사용해 본 시스템은..

- 클린 아키텍처

독립적인 계층을 통해 코드의 의존성을 최소화하고 유지 보수를 쉽게 하는 깔끔한 구조입니다. 비즈니스 로직과 UI를 분리해 가독성과 확장성을 높입니다.

- mvc 패턴

데이터와 UI, 로직을 분리하여 코드를 쉽게 관리할 수 있습니다. 웹에서 자주 사용하고

- DB 암호화

- 클라우드 아는 게 있다면 설명

인터넷을 통해 서버, 데이터베이스, 스토리지 등을 원격으로 제공하는 기술입니다. 비용 절감, 확장성, 유연성이 높아 다양한 환경에서 활용됩니다. 단점으로는 외부 서버에 저장하므로 보안 강화에 신경을 써야합니다.

IaaS : 서버, 스토리지 같은 인프라 제공. AWS EC2

PaaS : 플랫폼, 애플리케이션 개발 환경을 제공. google app engine.

SaaS : 완성된 소프트웨어를 제공. google workspace

- 트리거에 대해 아는지?

A라는 작업을 했을 때, B라는 작업이 연쇄적으로 실행되는 것입니다. 마치 방아쇠를 당기면 총알이 나가는 것처럼 말입니다. DB에서 트리거의 예로는,

- 오라클의 단점

초기 설치와 관리가 어렵고 비용이 높아서, 소규모 프로젝트에서는 과한 경우가 많다고 알고 있습니다.

- TCP/UDP 차이

TCP: 연결 지향적으로 데이터의 신뢰성을 보장함. 주로 파일 전송

UDP: 비연결성으로 속도가 빠르지만 신뢰성이 낮음. 주로 스트리밍

CSR (Client-Side Rendering): 웹 페이지의 렌더링을 클라이언트(브라우저) 측에서 수행합니다. 초기 로딩이 느릴 수 있지만, 이후에는 빠르고 동적인 사용자 경험을 제공합니다. 주로 React, Vue.js 같은 프론트엔드 프레임워크에서 많이 사용됩니다.

SSR (Server-Side Rendering): 서버에서 HTML을 미리 렌더링해 클라이언트에 전달하는 방식으로, 초기 로딩 속도가 빠르고 검색 엔진 최적화(SEO)에 유리합니다. Next.js 같은 프레임워크에서 지원합니다.

C#, 자바의 차이점

씨샵과 자바는 객체 지향 언어로 유사한 구조를 가지고 있지만 씨샵은 닷넷에서 주로 쓰이고 비동기 처리와 링크를 지원합니다. 자바는 JVM에서 실행되어 플랫폼 독립성을 강조하며 최근 람다와 같은 함수형 기능이 도입되었지만 씨샵 보다 제한적입니다.

JVM: 자바 프로그램을 실행하는 가상 머신으로 자바언어로 작성된 코드를 컴퓨터에서 실제로 실행할 수 있도록 중간에서 변환하는 역할을 합니다.

링크: 데이터 소스를 쉽게 조회하고 조작할 수 있도록 도와주는 기능입니다. SQL처럼 데이터를 필터링, 정렬, 그룹화하는 쿼리 문법을 씨샵 언어 자체에서 사용가능하도록 합니다.

자바에서 비동기 작업: 퓨처, 이그제큐터 서비스와 같은 도구 사용 (비동기 작업을 쉽게 도와주는 키워드는 X)

* 퓨처: 미래에 완료될 작업의 결과를 담는 객체

* 이그제큐터 서비스: 백그라운드에서 작업을 실행하는 서비스

씨샵에서 비동기 작업: 어싱크와 어웨이트라는 키워드를 제공해서 비동기 작업을 더 간편하게 쓸 수 있도록 언어에서 기본 지원

* 어싱크: 비동기 작업을 실행할 수 있는 함수 (함수가 끝나지 않아도 다음 코드로 넘어갈 수 O)

* 어웨이트: 어싱크 안에서 기다려야하는 작업에 붙임 (오래걸리는 작업에 사용하면 그 작업이 끝날때까지 기다린 후에 다음 코드 실행)

SaaS, IaaS, PaaS

사스: 완성된 소프트웨어를 인터넷으로 제공해주는 서비스 ex) 마이크로소프트 365

아이아스: 서버, 스토리지 같은 IT 인프라를 빌려주는 서비스 ex) AWS

파스: 애플리케이션 개발에 필요한 플랫폼과 환경을 제공하는 서비스 ex) 히어로쿠

OOP

현실의 사물을 코드로 표현하는 방식으로 데이터와 기능을 하나의 객체로 묶어 관리합니다. 특징으로는 캡슐화, 상속, 다형성, 추상화 등이 있고, 모듈 재사용으로 확장 및 유지보수가 용이합니다.

- * 캡슐화: 데이터와 기능을 하나의 객체로 묶고, 외부에서 접근을 제한하는 것 -> 데이터 보호, 무결성 유지가 쉬워짐
- * 상속: 기존 클래스의 속성과 기능을 물려받아 새로운 클래스를 만드는 것 -> 코드 재사용이 쉬워짐
- * 다형성: 같은 이름의 기능이 다양한 형태로 동작할 수 있게 하는 것
 - 오버로딩: 같은 클래스 내에서 메서드 이름은 같지만 매개변수를 다르게 정의하는 것
 - 오버라이딩: 상속받은 클래스에서 부모 클래스의 메서드를 재정의 하는 것
- * 추상화: 핵심 기능만 정의하고 세부 사항은 숨기는 것
 - 추상클래스 vs 인터페이스
 - 공통점: 두 개념 모두 추상적인 개념으로 직접적으로 객체를 생성할 수 X, 둘 다 다른 클래스에서 상속 받아 사용
 - 차이점: 인터페이스는 다중상속을 지원하고 모든 메서드가 추상 메서드이며 구현을 강제, 추상 클래스는 일부 구현을 가질 수 있고 단일 상속만 지원

SOLID

- * 단일 책임 원칙 (SRP): 클래스는 하나의 책임만 가져야한다. 하나의 목적을 위해 설계되어야 하며 한가지 역할만 수행해야 한다
- * 개방-폐쇄 원칙 (OCP): 확장에는 열려 있고, 수정에는 닫혀 있어야 한다. 새로운 기능을 추가할 때 기존 코드를 수정하지 않고 새로운 코드를 추가할 수 있어야한다
- * 리스코프 치환 원칙 (LSP): 자식 클래스가 부모 클래스의 기능을 완전히 대체할 수 있어야 한다.
- * 인터페이스 분리 원칙 (ISP): 여러 작은 인터페이스가 하나의 큰 인터페이스보다 낫다.
- * 의존성 역전 원칙 (DIP): 상위 모듈은 하위 모듈에 의존해서는 안되고, 둘 다 추상화에 의존해야한다. 구체적인 클래스에 의존하지 말고, 추상화된 인터페이스나 상위클래스에 의존해야한다

AOP

관계 지향 프로그래밍으로 핵심 비즈니스 로직에 있는 공통 관심사항을 분리하여 각각을 모듈화 하는 것을 의미합니다. 공통 모듈인 인증, 로깅, 트랜잭션 처리에 용이합니다.

get, post 역할

HTTP 요청 방식으로 웹 서버와 클라이언트 간에 데이터를 주고받을 때 사용됩니다.

GET 요청의 경우 서버에서 데이터를 요청할 때 사용됩니다. 요청 데이터는 URL에 쿼리 문자열 형태로 포함되어 전송되어 URL에 데이터가 보이기 때문에 민감한 데이터 전송에는 적합하지 않습니다. POST 요청은 서버에 데이터를 전송할 때 사용됩니다. HTTP 본문에 포함되어 전송되어 GET 보다 보안성이 높고 큰 데이터 전송이 가능합니다.

POST와 PUT의 차이

둘 다 서버에 데이터를 전송하여 리소스를 생성하거나 수정하는 HTTP 메서드지만 POST는 요청을 보낼때마다 새로운 리소스를 생성하여 중복 생성될 수 있습니다. PUT은 ID나 고유 식별자를 사용해 특정 리소스의 위치를 지정하여 요청을 보냅니다. 요청한 리소스가 존재하면 수정하고, 없으면 새로 생성할 수 있습니다. 같은 요청을 보내도 항상 같은 결과가 보장됩니다.

리액티브 프로그래밍

데이터 변화와 이벤트에 반응하여 비동기적으로 처리하는 프로그래밍 방식

RESTful API

HTTP 통신을 REST 설계 규칙을 잘 지켜서 개발한 API를 말합니다. REST 설계 규칙은 URI는 정보의 자원만 표현해야하며 자원의 상태와 행위는 HTTP 메서드에 명시하는걸 말합니다. GET 요청으로 데이터를 조회하고, POST로 새로운 데이터를 생성하며, PUT과 DELETE로 각각 데이터를 수정하고 삭제할 수 있습니다.

무상태성을 특징으로 해 각 요청이 독립적으로 처리되어 확장성과 성능이 높아지는 장점이 있습니다.

무상태성: 서버가 클라이언트의 이전 요청 상태를 기억하지 않는 것을 의미합니다. 즉, 각 요청은 독립적이며, 요청이 올 때마다 필요한 모든 정보를 포함하고 있어야 합니다.

운영체제란?

시스템의 자원과 동작을 관리하는 소프트웨어로 프로세스, 저장장치, 네트워킹, 사용자, 하드웨어 등을 관리하게 됩니다

메모리 구조

메모리는 크게 4가지 종류가 있습니다. 코드, 데이터, 힙, 스택이 있고 각각 코드는 소스코드가 들어가는 부분이고 데이터는 전역변수, 정적변수가 할당되는 부분입니다. 힙은 사용자가 직접 관리하는 영역으로 데이터가 동적으로 할당되는 공간입니다. 스택은 함수의 호출정보, 지역변수, 매개변수들이 저장되게 됩니다.

프로세스, 스레드

프로세스는 실행중인 프로그램입니다. 스레드는 프로세스 안에서 실행되는 흐름 단위입니다. 프로세스는 메모리와 CPU를 프로세스마다 할당받아서 사용하는데 스레드는 프로세스 안에서 다른 스레드와 메모리와 CPU를 공유해서 사용합니다.

CPU 스케줄링

준비큐에 있는 프로세스에 대해서 CPU를 할당하는 방법입니다. 크게 FCFS, SJF, SRT, 라운드 로빈(타임슬라이스), 우선순위 스케줄링이 있습니다.

가상메모리

모든 프로세스에 메모리를 할당하기에는 메모리 크기가 한계가 있어 사용하는 방법입니다. 프로세스에서 사용하는 부분만 메모리에 올리고 나머지는 디스크에 보관하는 기법을 가상메모리라고 합니다.

데드락

프로세스가 자원을 얻지 못해 다음 작업을 못하는 상태입니다. 예를 들어 프로세스1과 2가 각각 자원 B와 A를 얻어야되는데 프로세스1이 A를, 2가 B를 가지고 있어 서로 무한적 기다리는 상태를 데드락이라고 합니다. 데드락은 4가지 조건이 동시에 발생해야 성립되는데 네가지 조건은 상호배제, 점유대기, 비선점, 순환대기 입니다.

네트워크

OSI 7계층

인터넷 환경에서 통신하기 위해 네트워킹에 대한 표준을 7계층으로 나눈것입니다. 물리계층, 데이터링크계층, 네트워크계층, 전송계층, 세션계층, 표현계층, 응용계층이 있습니다

TCP, UDP

TCP는 신뢰성 있는 통신을 위해 사용하는 프로토콜로 높은 신뢰성을 보장하지만, UDP보다는 속도가 느립니다. 쓰리웨이, 포웨이 핸드셰이크로 서버와 클라이언트가 1대1로 통신을 하고 흐름제어와 혼잡제어가 이루어지게 됩니다. UDP는 비연결형 프로토콜로 손상된 데이터에 대해 재전송하지 않습니다. 신뢰성이 낮지만 대신 TCP보다 속도가 빨라서 스트리밍 같은 서비스에 주로 사용 됩니다. 마지막으로 일대일 일대다 다대다로 연결이 가능합니다.

IP

인터넷 프로토콜의 약자로 인터넷망을 통해 패킷을 전달하는 프로토콜 입니다. 비연결성과 비신뢰성의 특징을 가지는데 비연결성은 패킷을 보내는 길을 정하지 않는 것이고, 비신뢰성은 패킷의 완전한 전달을 보장하지 않는 것입니다.

DNS

도메인 주소를 IP주소로 변환해주는 시스템입니다. 다음과 같은 순서로 작동하게 되는데 첫째 URL을 입력하면 ISP가 관리하는 DNS해석기에 요청을 라우팅 시킵니다. 그 다음 DNS 해석기가 루트서버에 탑레벨의 서버주소를 요청하고 그 다음 탑레벨에서 세컨드 레벨, 세컨드 레벨에서 서브 DNS 서버로 작동하여 최종적으로 IP 주소를 얻게 됩니다.

로드밸런서

서버의 부하를 분산시켜주는 시스템 입니다. 크게 L4로드밸런서와 L7로드밸런서가 있는데 L4

로드밸런서는 4계층 이하의 정보를 가지고 로드를 분산해줍니다. 특히 맥주소, IP주소, 포트정보를 가지고 트래픽을 분산해주고 L7로드밸런서는 응용계층의 정보를 가지고 로드를 분산해줍니다. 패킷 내용을 확인하고 분산해서 디도스 같은 비정상적인 트래픽도 필터링할 수 있습니다.

데이터베이스

DBMS

데이터베이스 내 데이터에 접근하도록 도와주는 시스템입니다. 크게 질의처리기와 저장시스템으로 이루어져 있습니다.

트랜잭션 특징

원자성, 일관성, 격리성, 지속성 입니다. 원자성은 트랜잭션 내 실행한 작업은 모두 성공하거나 실패해야한다는 것입니다. 일관성은 데이터베이스 상태를 일관성있게 유지시키는것입니다. 격리성은 동시에 실행되는 트랜잭션은 서로 영향을 미치지 않아야한다는 것이고 지속성은 트랜잭션 완료시 결과가 영구적으로 반영되어야 한다는 것입니다.

NoSQL

Not Only SQL의 약자로 SQL을 보완한다는 의미

sql vs noSQL

개발한 프로젝트

개발환경, 사용언어

1)개체무결성(Entity Integrity)

• NULL불가, 중복불가

2)참조무결성(Referential Integrity)

• 외래키는NULL이거나참조릴레이션의기본키와일치해야함

• Restrict, Cascade

3)도메인무결성(Domain Integrity)

• 속성값은정의된도메인에속해야한다.- 67

[정보처리기사실기]마무리특강-데이터베이스구축

with.홍달쌤

4)고유무결성(Unique Integrity)

• 릴레이션의특정속성값은서로달라야한다.

5)키무결성(Key Integrity)

• 각릴레이션은적어도하나의키를가져야한다.

6)릴레이션무결성(Relation Integrity)

• 삽입,삭제,갱신등의연산은릴레이션의무결성을해치지않도록수행되어야한다

스택 : lifo 푸시 팝 픽 -> 웹 브라우저 방문기록 뒤로가기

큐 : fifo 엔큐 데큐 픽 -> 대기번호표

힙 : 우선순위. 최대힙 최소힙. 배열

선택 n제곱

버블 n제곱

삽입 n제곱

퀵 n로그n 피벗이 이동 / n제곱 (값이 한 쪽에 치우치면)

병합 분할정복 재귀 n로그n

기수 자릿수 n

HL7

다이콤에 대해

데이터 모델링 + 무결성 how

레스트폴

트랜잭션

데이터 정규화란

인덱스

회사에 대해 궁금한 점

본인이 오늘 면접을 통해 느낀 점 스스로 피드백

배열, 현재시간, k(10)

스크럼

데드락

쿼리 오류 - > 데이터 무결성을 위해 rollback을 호출