[예상문제] 2022년 2회 정보처리기사 실기 [2022-07-19 22:34 [2회차 합격하자]님까지 댓글 정리완료 !!)

소프트웨어 생명주기 (SDLC) **폭프나반 / 요설구테유**
- 시스템의 요구분석부터 유지보수까지 전 공정을 체계화한 절차이다.

폭포수 모델
- 각 개발 단계를 마무리 지은 후 넘어가는 모델이다.

프로토타이핑 모델
- 주요 기능을 프로토타입으로 구현하고, 피드백을 반영해 만들어가는 모델

나선형 모델
- 위험을 최소화하기 위해 점진적으로 완벽한 시스템으로 개발해 나가는 모델이다.

반복적 모델
- 병렬적으로 개발 후 통합하거나, 반복적으로 개발해 점차 완성시켜나가는 모델이다.

애자일 방법론
- 절차보다는 사람이 중심이 되어 변화에 유연한 경량 개발 방법론이다.

XP
- 가치**용단의피존**

1. 용기
2. 단순성
3. 의사소통
4. 피드백
5. 존중

- 12가지 기본 원리

1. 짝 프로그래밍(Pair Programming)
   - 개발자 둘이서 짝으로 코딩하는 원리이다.

2. 공동 코드 소유(Collective Ownership)
   - 시스템에 있는 코드는 누구나 수정 가능하다.

3. 지속적인 통합(CI; Continuos Integration)
   - 여러 번 소프트웨어를 통합하고 빌드 해야 한다.

4. 계획 세우기(Planning Proceess)
   - 고객이 원하는 가치를 정의하고 개발에 필요한 것 무엇인지, 어떤 곳에서 지연이 될 수 있는지 알려주어야 한다.

5. 작은 릴리즈(Small Release)
   - 작은 시스템을 먼저 만들고 짧은 단위로 릴리즈 해야 한다.

6. 메타포어(Metaphor)
   - 공통 이름 체계와 시스템 서술서를 통해 고객과 개발자 간의 의사소통을 원활하게 한다는 원리이다.

7. 간단한 디자인(Simple Design)
   - 요구사항에 적합한 단순한 시스템을 설계한다.

8. 테스트 기반 개발(TDD; Test Drive Develop)
   - 테스트를 먼저 수행하고, 테스트 요구사항에 맞도록 프로그램 수정한다.

9. 리팩토링(Refactoring)
   - 기능을 바꾸지 않고 중복 제거, 단순화 등을 위해 코드를 재구성한다.

10. 40시간 작업(40-Hour Work)
    - 피곤으로 인한 실수가 없도록 주 40시간 이상을 일하지 말아야 한다는 원리이다.
11. 고객 상주(On Site Customer)
    - 개발자들의 질문에 즉각 대답해줄 수 있는 고객을 프로젝트에 풀타임으로 상주시켜야 한다는 원리이다.
12. 코드 표준(Coding Standard)
    - 코딩 표준을 두고 효과적으로 개발한다.

스크럼

- 매일 정해진 시간/ 장소에서 짧은 시간의 개발을 위한 애자일 방법론이다.

요구공학 **도분명확**

- 요구사항을 도출, 분석, 명세, 확인하는 구조화된 활동이다.

형상 관리 **식통감기**

1. 형상 식별
   - 형상 관리 대상을 정의 및 식별하는 활동
2. 형상 통제
   - 형상 항목 버전 관리를 위해서 변경 여부와 변경 활동등을 통제하는 활동
3. 형상 감사
   - 소프트웨어 베이스라인 무결성 평가
4. 형상 기록
   - 소프트웨어 현상 및 변경관리에 대한 각종 수행결과를 기록

CMMI **매니저님 CMMI 정의 부탁드리겠습니다.. !**

- ISO15504(SPICE)를 준수하는 소프트웨어 개발능력/ 성숙도 평가 및 프로세스 개선 활동의 지속적인 품질 개선 통합 모델

데이터베이스

1. 정규화 **원부이결다조**
   - 데이터 중복성을 제거하여 이상 현상을 방지하고, 데이터 일관성 유지를 하는 무손실 분해하는 과정이다.

1-1. 1NF

- 원자값으로 구성

1-2 2NF

- 부분 함수 종속 제거(완전 함수적 종속 관계)

1-3 3NF

- 이행함수 종속 제거

1-4 BCNF

- 결정자 후보 키가아닌 함수 종속 제거

1-5 4NF

- 다치(다중 값) 종속 제거

1-6 5NF

- 조인 종속 제거

2. 반정규화
   - 성능 향상 및 관리운영 단순화를 위해 중복, 통합, 분리 등을 수행하는 모델링 기법이다.

NoSQL

- 데이터 저장에 고정된 테이블 스키마가 필요 없으며, 조인 연산을 사용할 수 없으며, 수평적으로 확장이 가능한 DBMS이다.

보안

1. Secure Coding **입보시 에코캡아**

1-1 입력 데이터 및 검증 표현

- 프로그램 입력값에 대한 검증 누락·부적절한 검증, 잘못된 형식 지정

1-2 보안 기능

- 보안 기능(인증, 접근 제어, 기밀성, 암호화, 권한 관리 등)의 부적절한 표현

1-3 시간 및 상태

- 거의 동시에 수행 지원하는 병렬 시스템 또는 하나 이상의 프로세스가 동작하는 환경에서 시간 및 상태의 부적절한 관리

1-4 에러 처리

- 에러 미처리, 불충분한 처리 등으로 에러 메시지에 중요정보가 포함

1-5 코드 오류

- 개발자가 범할 수 있는 코딩 오류로 인해 유발

1-6 캡슐화

- 기능성이 불충분한 캡슐화로 인해 인가되지 않은 사용자에게 데이터 누출

1-7 API 오용

- 의도된 사용에 반하는 방법으로 API를 사용하거나, 보안에 취약한 API의 사용

2. 대칭키

- 암호화와 복호화에 같은 암호 키를 쓰는 알고리즘이다.

2-1 DES

- 미국의 연방 표준국(NIST)에서 발표한 대칭 기반 블록 암호화 알고리즘
- 블록 크기 64bit, 키 길이 56bit, 16라운드 암호화 알고리즘

2-2 AES

- 2001년 NIST에서 발표한 블록 암호화 알고리즘
- 3DES 성능 문제 해결을 위해 개발
- 블록 크기 128bit, 라운드 수 10, 12, 14 라운드 분류

2-3 SEED

- 한국인터넷진흥원(KISA) 개발한 블록 암호화 알고리즘
- 128bit 비밀키로부터 생성된 16개의 64bit 라운드 키를 사용
- 블록 크기 128bit, 키 길이에 따라 128bit, 256bit 분류

2-4 ARIA

- 2004년 국가정보원과 산학연구협회가 개발한 블록 암호화 알고리즘
- 연산은 XOR과 같은 단순한 바이트 단위 연산으로 구성
- 블록 크기 128bit, 키 길이에 따라 128bit, 192bit, 256bit로 분류

2-5 IDEA

- DES를 대체하기 위해 스위스 에서 개발한 블록 암호화 알고리즘

2-6 LFSR

- 선형 되먹임 시프트 레지스터(LFSR)는 시프트 레지스터의 일종

3. 비대칭 키

- 사전에 개인 키를 나눠 가지지 않은 사용자들이 안전하게 통신하는 방식이다.

3-1 디피-헬만

- 최초 공개키 알고리즘
- 이산대수 계산이 어려운 문제를 기본 원리로 하고 있음

3-2 RSA

- 3명의 MIT 수학 교수가 고안한 큰 인수의 곱을 소인수 분해하는 수학적 알고리즘 이용하는 공개키 암호화 알고리즘

3-3 ECC

- 유한체 위에서 정의된 타원 곡선 군에서 이산대수의 문제에 기초한 공개키 암호화 알고리즘

3-4 ElGamal

- 이산대수의 계산이 어려운 문제를 기본 원리로 하고 있음
- RSA와 유사하게 전자서명과 데이터 암 복호화에 함께 사용 가능

4 해시 암호 방식 (일방향 암호 방식)

- 임의 길이의 정보를 입력받아, 고정된 길이의 암호문을 출력하는 암호 방식이다.

5. 서버 접근 통제의 유형

5-1. 강제적 접근 통제(MAC)

- 객체에 포함된 정보의 허용등급과 접근 정보에 대하여 주체가 갖는 접근 허가 권한에 근거하여객체에 대한 접근을 제한하는 방법

5-2. 임의적 접근 통제(DAC)

- 주체나 그룹의 신분(=신원)에 근거하여 객체에 대한 접근을 제한하는 방법

5-3 역할 기반 접근 통제(RBAC)

- 중앙 관리자가 사용자와 시스템의 상호관계를 통제하며 조직 내 맡은 역할에 기초하여 자원에 대한 접근을 제한하는 방법

6. 3A

6-1 인증(Authentication)

- 접근을 시도하는 가입자 또는 단말에 대한 식별 및 신분을 검증

6-2 권한 부여(Authorization)

- 검증된 가입자나 단말에게 어떤 수준의 권한과 서비스를 허용

6-3 계정 관리(Accounting)

- 리소스 사용에 대한 정보를 수집하고 관리하는 서비스

7. 보안 공격 관련 용어

7-1 워터링홀(Watering Hole)

- 악성코드를 배포하는 URL로 자동으로 유인하여 감염시키는 공격기법

7-2 스피어피싱(Spearfishing)

- 고위 공직자, 유명인 등 특정 개인 및 회사를 대상으로 개인정보를 캐내거나 특정 정보 탈취 목적으로 하는 피싱 공격

7-3 랜섬웨어(RansomWare)

- 악성코드 한 종류로 감염된 시스템 파일들을 암호화하여 복호화할 수 없도록 하고, 피해자로 하여금 암호화된 파일을 인질처럼 잡고 몸값을 요구하는 악성 소프트웨어

7-4 스턱스넷(Stuxnet)

- 독일 지멘스사의 SCADA 시스템을 공격 목표로 제작된 악성코드
- 산업 기반 시설의 제어 시스템에 침투해서 오작동을 일으키는 악성코드 공격 기법

7-5 드라이브 바이 다운로드(Drive By Download)

- 해커가 불특정 웹 서버와 웹 페이지에 악성 스크립트를 설치하고, 불특정 사용자 접속 시 사용자 동의 없이 실행되어 의도된 서버로 연결하여 감염시키는 공격기법

7-6 킬 스위치(Kill Switch)

- 스마트폰 이용자가 도난당한 스마트폰의 작동을 웹사이트를 통해 정지할 수 있도록 하는 일종의 자폭기능이다.

7-7 APT

- 특정 타깃을 목표로 하여 다양한 수단을 통한 지속적이고 지능적인 맞춤형 공격기법

8. iptables

- 리눅스상에서 방화벽을 설정하는 도구이다.

신기술

1. 클라우드

1-1. Infra as a Service (IaaS)

- 서버, 스토리지 같은 시스템 자원을 클라우드로 제공하는 서비스

1-2. Software as a Service (SaaS)

- 소프트웨어 및 데이터는 중앙에 호스팅되고 사용자는 웹 브라우저 등의 클라이언트를 통해 접속하여 소프트웨어를 서비스 형태로 이용하는 서비스

1-3. Platform as a Service (PaaS)

- 애플리케이션을 개발, 실행, 관리할 수 있게 하는 플랫폼을 제공하는 서비스

2. 빅데이터 3V

- 주어진 비용, 시간내에 처리 가능한 데이터 범위를 넘어서는 수십 페타바이트 크기의 비정형 데이터이다.

2-1. Volume

2-2. Variety

2-3. Velocity

3. IoT

3-1. MQTT

- 사물통신, 사물인터넷과 같이 대역폭이 제한된 통신 환경에 최적화하여 개발된 푸시기술 기반의 경량 메시지 전송 프로토콜

3-2. 블루투스**블루투스 공격말고는 블루투스 low energy 밖에 없네요 ㅠㅠ 매니저님 부탁드려요 !**

- 근거리 무선 기술 표준

3-3. 피코넷

- 여러 개의 독립된 통신장치가 블루투스 기술이나 UWB 통신 기술을 사용하여 통신망을 형성하는 무선 네트워크 기술

응집도 **우논시절 통순기**

1. 우연적 응집도 (Coincidental Cohesion)
   - 모듈 내부의 각 구성 요소들이 서로 관련 없는 요소만 구성된 응집도
2. 논리적 응집도 (Logical Cohesion)
   - 유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소들이 한 모듈에서 처리되는 경우의 응집도
3. 시간적 응집도 (Temporal Cohesion)
   - 연관된 기능이라기보다는 특정 시간에 처리되어야 하는 활동들을 한모듈에서 처리할 경우의 응집도
4. 절차적 응집도 (Procedural Cohesion)
   - 모듈이 다수의 관련 기능을 가질 때 모듈 안의 구성요소들이 그 기능을 순차적으로 수행할 경우의 응집도
5. 통신적(교환적) 응집도 (Communication Cohesion)
   - 동일한 입 출력을 사용하여 다른 기능을 수행하는 활동들이 모여 있을 경우의 응집도
6. 순차적 응집도 (Sequential Cohesion)
   - 모듈 내에서 한 활동으로부터 나온 출력값을 다른 활동이 사용할 경우의 응집도
7. 기능적 응집도 (Functional Cohesion)
   - 모듈 내부의 모든 기능이 단일한 목적을 위해 수행되는 경우의 응집도

결합도**내공외제스자**

1. 내용 결합도 (Content Coupling)
   - 다른 모듈 내부에 있는 변수나 기능을 다른 모듈에서 사용하는 경우의 결합도
2. 공통 결합도 (Common Coupling)
   - 파라미터가 아닌 모듈 밖에 선언되어 있는 전역 변수를 참조하고 전역변수를 갱신하는 식으로 상호 작용하는 경우의 결합도
3. 외부 결합도 (External Coupling)
   - 두 개의 모듈이 외부에서 도입된 데이터 포맷, 통신 프로토콜, 또는 디바이스 인터페이스를 공유할 경우의 결합도
4. 제어 결합도 (Control Coupling)
   - 어떤 모듈이 다른 모듈의 내부 논리 조직을 제어하기 위한 목적으로 제어 신호를 이용하여 통신하는 경우의 결합도
5. 스탬프 결합도 (Stamp Coupling)
   - 모듈 간의 인터페이스로 배열이나 객체, 구조 등이 전달되는 경우의 결합도
6. 자료 결합도 (Data Coupling)
   - 모듈 간의 인터페이스로 전달되는 파라미터를 통해서만 모듈 간의 상호 작용이 일어나는 경우의 결합도

OSI 7 Layer**아파서 티내다, 피나다**

1. Physical Layer **단위 : bit**
   - 0과 1의 비트 정보를 회선에 보내기 위한 전기적 신호 변환
   - 프로토콜 : RS-232C
   - 장비 : 허브, 리피터
2. DataLink Layer**단위 : Frame**
   - 인접 시스템 간 데이터 전송, 전송 오류 제어
   - 동기화, 오류제어, 흐름제어, 회선 제어
   - 프로토콜 : HDLC, PPP
   - 장비 : 브리지, 스위치
3. Network Layer**단위 : Packet**
   - 단말기 간 데이터 전송을 위한 최적화된 경로 제공
   - 프로토콜 : IP, ICMP
   - 장비 : 라우터
4. Transport Layer**단위 : Segment**
   - 송수신 프로세스 간의 연결
   - 신뢰성 있는 통신 보장
   - 프로토콜 :TCP, UDP
   - 장비 : L4 스위치
5. Session Layer **단위 : Data**
   - 송수신 간의 논리적인 연결
   - 연결 접속, 동기 제어
   - 프로토콜 : RPC, NetBIOS
   - 장비 : 호스트
6. Presentation Layer **단위 : Data**
   - 데이터 형식 설정, 부호교환, 암 복호화
   - 프로토콜 :JPEG, MPEG
   - 장비 : 호스트
7. Application Layer**단위 : Data**
   - 사용자와 네트워크 간 응용서비스 연결, 데이터 생성
   - 프로토콜 : HTTP, FTP
   - 장비 : 호스트

스케줄링

1. 선점형 스케줄링 (Preemptive Scheduling)
   - 하나의 프로세스가 CPU를 차지하고 있을 때 우선순위가 높은 다른 프로세스가 현재 프로세스를 중단 시키고 CPU를 점유하는 스케줄링 방식

SRT(Shortest Remaining time first)
- SJF(비선점형) 기법을 **선점형**으로 바꾼 형태
- 가장 **짧은 시간이 소요되는 프로세스를 먼저 수행**하고, 남은 처리 시간이 더 짧다고 판단되는 프로세스가 준비 큐에 생기면 **언제라도 프로세스 점유**가 되는 방식

다단계 피드백 큐 (Multi Level Feedback Queue)
- 새로운 프로세스는 높은 우선순위를 가지고 프로세스의 실행시간이 길어질수록 점점 낮은 큐로 이동하고 마지막 단계는 라운드 로빈 방식을 적용하는 방식
- **FCFS + Round Robin**

다단계 큐 (Multi Level Queue)
- **작업들을 여러 종류의 그룹으로 분할,**여러 개의 큐를 이용하여 상위 단계 작업에 의한 하위 단계 작업이 선점당함

라운드 로빈(Round Robin)
- 프로세스는 **같은 크기의 CPU 시간을 할당**, 프로세스가 **할당된 시간 내에 처리완료를 못하면 준비 큐 리스트의 가장 뒤로 보내지고,**CPU는 대기중인 다음 프로세스로 넘어감

2. 비선점형 스케줄링 (Non Preemptive Scheduling)
   - 한 프로세스가 CPU를 할당 받으면 작업 종료 후 CPU 반환 시까지 다른 프로세스는 CPU 점유가 불가능한 스케줄링 방식

우선순위(Priority)
- 각 프로세스에게 **우선 순위를 부여**하여**순위가 높은 순서대로**처리하는 방법

기한부(Deadline)
- 작업들이 명시된 시간이나 **기한 내에 완료**되도록 하는 계획

HRN(Highest Response (Ratio) Next)
- SJF 스케줄링 기법을 보완한 스케줄링 방식으로, **시스템 응답이 커질수록 우선순위가 높아**진다는 의미

FIFO(First In First Out)

- 각 페이지가 주기억장치에 적재될 때마다 **가장 먼저 들어왔던 페이지**가 가장 오래 있었기 때문에 **해당 페이지를 교체**하는 기법

SJF(Shortest Job First)

- **비선점형**
- 프로세스가 **도착하는 시점에 따라** 그 당시 **가장 작은 서비스 시간을 갖는 프로세스가 종료 시 까지 자원을 점유**하는 방식

블랙 박스 테스트**동경결상 유분 폐원비**

- 소프트웨어가 수행할 특정 기능을 알기 위해, **각 기능이 완전히 작동되는 것을 입증**하는 테스트

1. 동등분할 테스트 = 동치분할 테스트(Equivalence Partitioning Testing)
   - 입력 데이터의 영역을 **유사한 도메인 별로 유효 값/무효 값을 그룹핑하여 대푯값 테스트 케이스를 도출**하여 테스트 하는 기법

2. 경곗값 분석 테스트 = 한곗값 테스트(Boundary Value Analysis Testing)
   - 등가 분할 후 경곗 값 부분에서 오류 발생 확률이 높기 때문에 **경곗 값을 포함하여 테스트 케이스를 설계**하여 테스트 하는 기법

3. 결정 테이블 테스트(Decision Table Testing)
   - **요구사항의 논리와 발생 조건을 테이블 형태로 나열**하여, 조건과 행위를 모두 조합하여 테스트 하는 기법

4. 상태 전이 테스트(State Transition testing)
   - 테스트 대상·시스템이나 객체의 상태를 구분하고, **이벤트에 의해 어느 한 상태에서 다른 상태로 전이**되는 경우의 수를 수행하는 테스트 기법

5. 유스케이스 테스트(Use Case Testing)
   - 시스템이 실제 사용되는 **유스케이스로 모델링 되어있을 때 프로세스 흐름을 기반으로 테스트 케이스를 명세화 하여 수행**하는 테스트 기법

6. 분류 트리 테스트(Classification Tree Method Testing)
   - SW의 일부 또는 전체를 **트리 구조로 분석 및 표현하여 테스트 케이스를 설계**하여 테스트 하는 기법

7. 페어와이즈 테스트 (Pairwise Testing)
   - 테스트 데이터값 간에 **최소한 한 번씩을 조합**하는 방식
   - 커버해야 할 범위를 상대적으로 적은 양의 테스트 세트로 구성

8. 원인-결과 그래프 테스트(Cause-Effect Graphing Testing)
   - **그래프를 활용**하여 **입력 데이터 간의 관계 및 출력에 미치는 영향을 분석**하여 **효용성이 높은**테스트 케이스 를 선정하여 테스트 하는 기법

9. 비교 테스트(Comparison Testing)
   - 여러 버전의 프로그램에 **같은 입력 값을 넣어서 동일한 결과 데이터가 나오는지 비교**해 보는 테스트 기법

화이트 박스 테스트 **구결조 조변다 기제데**

- 각 **응용 프로그램의 내부 구조와 동작을 검사**하는 소프트웨어 테스트

1. 구문 커버리지(Statement Coverage)
   - 프로그램 내의 **모든 명령문을 적어도 한 번 수행**하는 커버리지

2. 결정 커버리지=선택 커버리지 (Decision Coverage) = 분기 커버리지(Branch Coverage)
   - 결정 포인트 내의 **전체 조건식이 적어도 한 번은 참과 거짓의 결과를 수행**하는 테스트 커버리지

3. 조건 커버리지(Condition Coverage)
   - 결정 포인트 내의 **개별 조건식이 적어도 한 번은 참과 거짓의 결과가 되도록 수행**하는 테스트 커버리지

4. 조건/결정 커버리지(Condition/Decision Coverage)
   - **전체 조건식 뿐만 아니라 개별 조건식도**참 한 번, 거짓 한 번 결과가 되도록 수행하는 테스트 커버리지

5. 변경 조건/결정 커버리지(Modified Condition/Decision Coverage)
   - 개별 조건식이 다른 개별 조건식에 영향을 받지 않고 **전체 조건식에 독립적으로 영향을 주도록 함**으로써 조건/결정 커버리지를 향상시킨 커버리지

6. 다중 조건 커버리지(Multiple Condition Coverage)
   - 결정 조건 내 모든 개별 조건식의 **모든 가능한 조합을 100% 보장**하는 커버리지

7. 기본 경로 커버리지 = 경로 커버리지 (Base Path Coverage)
   - 수행 가능한 **모든 경로를 테스트**하는 기법

8. 제어 흐름 테스트(Control Flow Testing)
   - 프로그램 **제어 구조를 그래프 형태로 나타내어**내부 로직을 테스트하는 기법

9. 데이터 흐름 테스트(Data Flow Testing)
   - **제어 흐름 그래프에 데이터 사용현황을 추가한 그래프**를 통해 테스트하는 기법

10. 루프 검사[테스트](Loop Testing)
   - 프로그램의 **반복구조에 초점**을 맞추어 테스트

디자인 패턴 유형

생성(Creational) 패턴 **생 빌프로 팩앱싱**

- 빌더(Builder)
- **생성과 표기를 분리**해 복잡한 객체를 생성
- 프로토타입(Prototype)
- **기존 객체를 복제**함으로써 객체를 생성
- 팩토리메서드(Factory Method)
- **생성할 객체의 클래스를 국한하지 않고**객체를 생성
- 추상 팩토리(Abstract Factory)
- 동일한 주제의 **다른 팩토리를 묶음**
- 싱글톤(Singleton)
- **한 클래스에 한 객체만 존재**하도록 제한

구조(Structural) 패턴**구 브데 퍼플 프록 컴어**

- 브리지(Bridge)
- **구현뿐만 아니라, 추상화된 부분까지 변경**해야 하는 경우
- 데코레이터(Decorator)
- 객체의 결합을 통해 **기능을 동적**으로 유연하게 확장
- 퍼사드(facade)
- **통합된 인터페이스**제공
- 플라이웨이트(Fly-Weight)
- 여러 개의 '가상 인스턴스'를 제공하여 **메모리 절감**
- 프록시(Proxy)
-**특정 개체로의 접근을 제어**하기 위한 용도
- 컴포지트(Composite)
- **복합 객체와 단일 객체를 동일하게 취급**
- 어댑터(Adapter)
- **기존에 생성된 클래스를 재사용**할 수 있도록 중간에서 맞춰주는 역할

행위(Behavioral) 패턴 **행 미인이 옵데 스테비커 스트메체**

- 미디에이터(Mediator)
- 상호작용의 유연한 변경을 지원
- 인터프리터(Interpreter)
- 문법 자체를 캡슐화
- 이터레이터 (iterator)
- 컬렉션 구현 방법을 노출시키지 않으면서그 집합체 안에 들어있는 모든 항목에 접근할 수 있는 방법을 제공하는 디자인 패턴
- 템플릿 메소드(Template Method)
- 상위 작업의구조를 바꾸지 않으면서서브 클래스로 작업의 일부분을 수행
- 옵저버(Observer)
- 한 객체의 상태가 바뀌면그 객체에 의존하는 다른 객체들한테 연락이 가고 자동으로 내용이 갱신되는 방법을 제공하는 디자인 패턴
- 비지터(Visitor)
- 특정 구조를 이루는 복합 객체의 원소 특성에 따라 동작을 수행할 수 있도록 지원하는 행위
- 커맨드(Command)
- 요구사항을 객체로 캡슐화
- Strategy
- 행위 객체를 클래스로 캡슐화 해 동적으로 행위를 자유롭게 변환
- State
- 객체의 상태에 따라 행위 내용을 변경
- Memento
- 객체를 이전 상태로 복구시켜야하는 경우, '작업취소(undo)' 요청
- Chain Of Responsibility
- 한 요청을 2개 이상의 객체에서 처리
- 예) 마우스클릭/ 키보드클릭 -> 전달

UML(Unified Modeling Language)

- 객체지향 소프트웨어 개발 과정에서 **산출물을 명세화, 시각화, 문서화할 때 사용되는 모델링 기술과 방법론을 통합해서 만든**표준화된 범용 모델링 언어

UML 관계

1. 연관 관계(Association Relationship)
   - 2**개 이상의 사물이 서로 관련**되어 있는 관계
   - 방향성은 화살표로 표현 -----> 저는 "연실화"로 외웠어요 ! (연관 관계는 실선 화살표)
   - 양방향일 경우 화살표 생략하고 실선으로만 연결
2. 집합 관계(Aggregation Relationship)
   - 하나의 사물이 다른 사물에 포함되어 있는 관계
   - 포함하는 쪽과 포함되는 쪽은 서로 독립적
   - 속이 빈 마름모 화살표로 연결 --> 저는 "집속빈마"로 외웠어요 ! (집합 관계는 속이 빈 마름모)

3. 포함 관계(Composition Relationship)
   - **포함하는 사물의 변화가 포함되는 사물에게 영향**을 미치는 관계
   - 속이 채워진 마름모 -> 저는 "**포속마**"로 외웠어요 ! (포관 관계는 속이 채워진 마름모)

4. 일반화 관계(Generalization Relationship)
   - 하나의 사물이 다른 사물에 비해 더**일반적인지 구체적인지 표현**
   - 속이 빈 화살표로 연결 -> 저는 "일빈화"로 외웠어요 ! (일반화 관계는 속이 빈 화살표)

5. 의존관계(Dependency Relationship)
   - 사물 사이에 서로 연관은 있으나 필요에 의해 **서로에게 영향을 주는 짧은 시간 동안만 연관을 유지**하는 관계
   - 점선 화살표로 연결 ----> 저는 "**의점화**"로 외웠어요 ! (의존 관계는 점선 화살표)

6. 실체화 관계(Realization Relationship)
   - 할 수 있거나 해야 하는 기능, **서로를 그룹화 할 수 있는 관계**
   - 속이 빈 점선 화살표로 연결 ----------------------->
     저는 "**실속빈점화**"로 외웠어요 ! (실체화 관계는 속이 빈 점선 화살표)

UML 구성요소 **사관다**

1. 사물
2. 관계
3. 다이어그램

UML 다이어그램

1. 구조적 다이어 그램

1-1. 클래스 다이어그램

- 클래스 간 관계를 표현

1-2. 객체 다이어그램

- 객체 간 관계를 표현

1-3. 컴포넌트 다이어그램

- 컴포넌트 간 관계를 표현

1-4. 배치 다이어그램

- 물리적 요소들의 위치를 표현

1-5. 복합체 구조 다이어그램

- 복합 구조인 경우 그 내부 표현

1-6. 패키지 다이어그램

- 패키지 간 관계 표현

2. 동적 다이어그램

2-1 유스케이스 다이어그램

- 사용자 관점에서 표현

2-2 시퀀스 다이어그램

- "시간적 개념" 중심으로 메시지 표현

2-3 커뮤니케이션 다이어그램

- 객체들이 주고 받는 메시지와 상호작용 (객체 간 연관)까지 표현

2-4 상태 다이어그램

- 객체의 상태와 상태를 표현

2-5 활동 다이어그램

- 시스템이 수행하는 활동을 표현

2-6 타이밍 다이어그램

- 객체의 상태 변화와 시간 제약을 표현

3. UML 스테레오 타입

3-1. << >> : 길러멧 기호

3-2 <<include>> : 어떤 시점에 반드시 다른 유스케이스를 실행

3-3 <<extend>> : 어떤 시점에 다른 유스케이스를 실행 할 수도 있고 아닐 수도 있음

3-4 <<abstract>> 추상 클래스 (인스턴스 생성x, 공통 특징만 정의)

3-5 <<interface>> 모든 메서드와 상수가 추상인 클래스

3-6 <<entity>> 정보 또는 행위를 표현하는 클래스

3-7 <<boundary>> 상호작용을 담당하는 클래스

3-8 <<control>> 로직 및 제어를 담당하는 클래스

SOAP

- HTTP, HTTPS, SMTP 등을 통해서 XML 기반의 데이터를 주고 받는 프로토콜

WSDL

- 웹 서비스명, 프로토콜 정보 등 웹 서비스에 대한 상세 정보가 기술된 XML 형식으로 기술한 언어 및 파일

JSON

- AJAX를 위해 키-값(Key-Value)쌍과 속성-값(Attribute-Value)쌍으로 이루어진 데이터 오브젝트를 전달하기 위해 인간이 읽을 수 있는 텍스트를 사용 하는 개방형 표준 포맷이다.

AJAX

- 브라우저가 가지고 있는 XMLHttpRequest 객체를 이용해서 전체 페이지를 새로 고치지 않고도 페이지의 일부분만을 위한 데이터를 로드하는 기법

REST

- 웹과 같은 분산 하이퍼미디어 시스템을 위한 소프트웨어 아키텍처의 한 형식이다.

오버로딩

- 한 클래스 내에서 메서드를 중복해서 생성하는 것이다.

오버라이딩

- 부모 클래스로부터 상속받은 메서드를 재정의 하는 것이다.

추상클래스

- 유사 클래스들의 공통된 특징을 정의하고, 하나 이상의 추상 메서드와 일반 필드, 메서드를 포함하는 클래스이다.

인터페이스 (네이버 참고 태클 걸어주세요 ㅎㅎ)

- 서로 다른 두 시스템, 장치, 소프트웨어를 서로 이어주는 부분이다.
- 사용자인 인간과 컴퓨터를 연결하여 주는 장치이다.

티어드롭(Tear Drop)

- IP Fragment Offset 값을 서로 중첩되도록 조작하여 수신 측이 재조합하는 과정에서 오류 발생 및 시스템 기능을 마비시키는 공격

DRDos

- 공격자가 출발지 IP를 공격대상 IP로 위조하여 다수의 반사 서버로 정보 전송, 공격 대상은 반사 서버로부터 다량의 응답을 받아 서비스가 거부되는 공격

SYN Flooding

- TCP의 구조적인 문제를 이용한 공격으로 SYN 패킷만 보내 점유하여 서버를 사용 불가능하게 하는 공격이다.

스머핑(스머프 공격)

- 출발지 주소를 공격대상 IP 주소로 설정하여 직접 브로드 캐스팅하여 타깃 시스템을 마비시키는 공격

Slowloris

- HTTP 프로토콜 취약점을 이용한 공격으로, header와 body를 구분하기 위해 정상적으로 /r/n/r/n header 끝에 전송해야 하지만 /r/n을 1번만 보내어 나머지 /r/n이 올 때 까지 기다리게 하는 공격이다.

RUDY

- 요청 헤더의 Content – Length를 비정상적으로 크게 설정하여 메시지 바디부분을 매우 소량으로 보내 계속 연결 상태 유지시켜 자원을 소진 시키는 기법

성능 테스트부스스내

1. 부하 테스트 (Load Testing)
   - 시스템의 부하를 계속 증가시키면서 시스템의 임계점을 찾는 테스트
2. 강도 테스트 (Stress Testing)
   - 임계점 이상의 부하를 가하여 비정상적인 상황에서의 처리를 테스트
3. 스파이크 테스트 (Spike Testing)
   - 짧은 시간에 사용자가 몰릴 때 시스템의 반응 측정 테스트
4. 내구성 테스트 (Endurance Testing)
   - 오랜 시간 동안 시스템에 높은 부하를 가하여 시스템 반응 테스트

IPv6

- IPv4가 가지고 있는 주소 고갈, 보안성, 이동성 지원 등의 문제점을 해결하기 위해서 개발된 128Bit 주소체계를

갖는 차세대 인터넷 프로토콜이다.

- 특징
- IP 주소의 확장, 이동성, Ad-hoc 네트워크 지원, Plug&Play 지원 등

형상 관리 (Configuration Management)

- 소프트웨어 개발을 위한 전체 과정에서 발생하는 모든 항목의 변경 사항을 관리하기 위한 활동

형상 식별

- 형상 관리 대상을 정의 및 식별하는 활동

형상 통제

- 형상 항목의 버전 관리를 위한 형상통제위원회운영

  *형상통제위원회(CCB) : 형상 관리에 대한 주요방침을 정하고 산출물을 검토하며, 단계별 의사결정을 수행하는 조직이다.

형상 감사

- 소프트웨어 베이스라인의 무결성 평가

형상 기록

- 소프트웨어 형상 및 변경 관리에 대한 각종 수행결과를 기록

베이스라인 (Baseline)

- 개발 과정의 각 단계의 산출물을 검토, 평가, 조정, 처리 등 변화를 통제하는 시점의 기준

소프트웨어 버전관리 도구 **공클분**

1. 공유 폴더 방식
   - 버전 관리 자료가 로컬 컴퓨터의 공유 폴더에 저장되어 관리되는 방식

2. 클라이언트·서버 방식
   - 버전 관리 자료가 중앙 시스템(서버)에 저장되어 관리되는 방식

3. 분산저장소 방식
   - 버전 관리 자료가 하나의 원격 저장소와 분산된 개발자 PC의 로컬 저장소에 함께 저장되어 관리되는 방식

관계 대수
   - 원하는 정보와 그 정보를 어떻게 유도하는가를 기술하는 절차적 정형 언어

1. 일반 집합 연산자 **합교차카**

1.1 합집합
   - 합병 가능한 두 릴레이션 R과 S의 합집합

1.2 교집합
   - 릴레이션 R과 S에 속하는 모든 튜플로 결과 릴레이션 구성

1.3 차집합
   - R에 존재하고 S에 미 존재하는 튜플로 결과 릴레이션 구성

1.4 카티션 프로덕트
   - R과 S에 속한 모든 튜플을 연결해 만들어진 새로운 튜플로 릴레이션 구성

2. 순수 관계 연산자 **셀프조디**

2.1 셀렉트 -> 조건에 만족

2.2 프로젝트 -> R에 관련된 속성들만(주어진 속성들의 값으로만)

2.3 조인 -> 공통된 속성들

2.4 디비전 -> S에 관련된 R 속성 반환

관계 해석

- 튜플 관계 해석과 도메인 관계 해석을 하는 비절차적 언어

소프트웨어 개발 보안의 3대 요소 **기무가**

기밀성(Confidentiality)

- 인가되지 않은 개인 혹은 및 시스템 접근에 따른 정보 공개 및 노출을 차단하는 특성

무결성(Integrity)

- 정당한 방법을 따르지 않고서는 데이터가 변경될 수 없으며, 데이터의 정확성 및 완전성과 고의/악의로 변경 되거나 훼손되지 않음을 보장하는 특성

가용성(Availability)

- 권한을 가진 사용자나 애플리케이션이 원하는 서비스를 지속 사용할 수 있도록 보장하는 특성

SW 개발 보안 용어 **자위취위**

자산(Asset)

- – 조직의 데이터 또는 소유자가 가치를 부여한 대상

위협(Threat)

- – 조직이나 기업의 자산에 악영향을 끼칠 수 있는 사건이나 행위

취약점(Vulnerability)

- – 위협이 발생하기 위한 사전 조건으로 시스템의 정보 보증을 낮추는 데 사용되는 약점

위험(Risk)

- – 위협이 취약점을 이용하여 조직의 자산 손실 피해를 가져올 가능성

EAI(Enterprise Application Integration) **포허메하**

- 기업에서 운영되는 서로 다른 플랫폼 및 애플리케이션 간의 정보를 전달, 연계, 통합이 가능하도록 해주는 솔루션

1.1 포인트 투 포인트(Point to Point)

- 가장 기초적인 애플리케이션 통합 방법으로 1:1 단순 통합방법

1.2 허브 앤 스포크(Hub&Spoke)

- 단일한 접점의 허브 시스템을 통하여 데이터를 전송하는 중앙 집중식 방식

1.3 메시지 버스(Message Bus)

- 애플리케이션 사이 미들웨어를 두어 연계하는 미들웨어 통합 방식

1.4 하이브리드(Hybrid)

- 그룹 내부는 허브 앤 스포크 방식을 사용하고, 그룹 간에는 메시지 버스 방식을 사용하는 통합 방식

ESB(Enterprise Service Bus) 방식

- 기업에서 운영되는 서로 다른 플랫폼 및 애플리케이션 들 간을 하나의 시스템으로 관리 운영할 수 있도록 서비스 중심의 통합을 지향하는 아키텍처

병행 제어(Concurrency Control)

- 다수 사용자 환경에서 여러 트랜잭션을 수행할 때 데이터베이스 일관성 유지를 위해 상호 작용을 제어하는 기법

병행 제어 미보장 시 문제점 **갱현모연**

1. 갱신 손실(Lost Update)
   - 먼저 실행된 트랜잭션의 결과를 나중에 실행된 트랜잭션이 덮어쓸 때 발생하는 오류
2. 현황 파악오류(Dirty Read)
   - 트랜잭션의 중간 수행 결과를 다른 트랜잭션이 참조하여 발생하는 오류
3. 모순성(Inconsistency)
   - 두 트랜잭션이 동시에 실행되어 데이터베이스의 일관성이 결여되는 오류
4. 연쇄복귀(Cascading Rollback)
   - 복수의 트랜잭션이 데이터 공유 시 특정 트랜잭션이 처리를 취소할 경우 트랜잭션이 처리한 곳의 부분을 취소하지 못하는 오류

회복 기법

- 트랜잭션을 수행하는 도중 장애로 인해 손상된 데이터베이스를 손상되기 이전의 정상적인 상태로 복구시키는 작업

RIP(Routing Information Protocol)

- 거리벡터 알고리즘에 기초하여 개발된 내부 라우팅 프로토콜이다.

BGP(Border Gateway Protocol)

- 경로 정보를 교환하기 위한 라우팅 프로토콜이다.

OSPF(Open Shortest Path First)

- 규모가 크고 복잡한 TCP/IP 네트워크에서 RIP의 단점을 개선하기 위해 자신을 기준으로 링크 상태 알고리즘을 적용하여 최단 경로를 찾는 라우팅 프로토콜

비즈니스 연속성 계획(BCP)

- 각종 재해, 장애, 재난으로부터 위기관리를 기반으로 재해복구, 업무복구 및 재개 등을 통해 비즈니스 연속성을 보장하는 체계이다.

BIA

- 장애나 재해로 인해 운영상의 주요 손실을 볼 것을 가정하여 시간 흐름에 따른 영향도 및 손실평가를 조사하는 BCP를 구축하기 위한 비즈니스 영향 분석

RTO

- – 업무중단 시점부터 업무가 복구되어 다시 가동될 때까지의 시간

RPO

- – 업무중단 심점부터 데이터가 복구되어 다시 정상가동될 때 데이터의 손실 허용 시점

방화벽(Firewall)

- 내부의 네트워크와 인터넷 간에 전송되는 정보를 선별하여 수용, 거부, 수정하는 기능을 가진 침입 차단 시스템
- 외부로 나가는 패킷은 그대로 통과시키고 외부에서 들어오는 패킷은 엄밀히 체크해 인증된 패킷만 통과시킴

침입 차단 시스템 (IPS; Intrusion Prevention System)

- 네트워크에 대한 공격이다 침입을 실시간적으로 차단하고, 유해 트래픽에 대한 조치를 능동적으로 처리하는 시스템이다.

침입 탐지 시스템(IDS; Intrusion Detection System)

- 네트워크에서 발생하는 이벤트를 모니터링하고 비인가 사용자의 침입을 실시간으로 탐지하는 시스템이다.

TCP(Transmission Control Protocol)

- 연결 지향적이고, 신뢰성이 있으며, IP 프로토콜 위에서 연결형 서비스를 지원하는 전송계층 프로토콜이다.

UDP(User Datagram Protocol)

- 비연결성이고, 신뢰성이 없으며, 순서화되지 않은 데이터그램 서비스를 제공하는 전송 계층 프로토콜이다.

트랜잭션(Transaction)

- 인가받지 않은 사용자로부터 데이터를 보장하기 위해 DBMS가 가져야하는 특성
- DB에서 논리적 기능 수행을 위한 작업의 최소 단위

1. 특성

1.1 원자성(Atomcitiy)

- 트랜잭션을 구성하는 연산 전체가 모두 정상적으로 실행되거나 모두 취소되어야 하는 성질

1.2 일관성(Consistency)

- 시스템이 가지고 있는 고정 요소는 트랜잭션 수행 전과 수행 완료 후의 상태가 같아야 하는 성질

1.3 고립성(Isolation)

- 동시에 실행되는 트랜잭션들이 서로 영향을 미치지 않아야 한다는 성질

1.4 영속성(Durability)

- 성공이 완료된 트랜잭션의 결과는 영속적으로 데이터베이스에 저장되어야 하는 성질

XSS(Cross Site Script)

- 검증되지 않은 외부 입력 데이터가 포함된 웹페이지가 전송되는 경우, 사용자가 해당 웹페이지를 열람함으로써 웹페이지에 포함된 부적절한 스크립트가 실행 되는 공격

CSRF(Cross-Site Request Forgery)

- 사용자가 자신의 의지와는 무관하게 공격자가 의도한 행위를 특정 웹 사이트에 요청하게 하는 공격

IP Spoofing

- IP 프로토콜의 취약점을 이용하여 IP 주소를 임의로 변조해서 접속하는 행위를 의미한다.

ARP Spoofing

- 특정 호스트의 MAC 정보를 공격자의 MAC 정보로 변경하여 스니핑 하는 공격 기법

트로이 목마(Trojan Horses)

- 악성 루틴이 숨어 있는 프로그램으로 겉보기에는 정상적인 프로그램으로 보이지만 실행하면 악성 코드를 실행 하는 프로그램

ICMP Redirect

- 3계층에서 스니핑 시스템을 네트워크에 존재하는 또 다른 라우터라고 알림으로써 패킷의 흐름을 바꾸는 공격기법

패스워드 크래킹 유형

1. 패스워드 탐지 공격 (Password Detection Attack)
시스템 또는 서비스의 ID와 패스워드를 크랙하기 위해서 ID와 패스워드가 될 가능성이 있는 단어를 파일로 만들어 놓고 이 파일의 단어를 대입하여 크랙하는 공격기법

[패스워드 탐지 공격의 유형]

가. 사전(Dictionary) 크래킹(=사전 대입 공격)
- 시스템 또는 서비스의 ID와 패스워드를 크랙하기 위해서 ID와 패스워드가 될 가능성이 있는 단어를 파일로 만들어 놓고 이 파일의 단어를 대입하여 크랙하는 공격기법

나. 무차별(Brute Force) 크래킹 (=무차별 대입 공격)
- 패스워드로 사용될 수 있는 영문자(대소문자), 숫자, 특수문자 등을 무작위로 패스워드 자리에 대입하여 패스워드를 알아내는 공격기법

다. 레인보우 테이블을 이용한 공격
- 패스워드별로 해시값을 미리 생성해서 크래킹하고자 하는 해시값을 테이블에서 검색해서 역으로 패스워드를 찾는 방법

테스트 레벨 종류 **단통시인**
1. 단위테스트
   - 사용자의 요구사항에 대한 단위 모듈, 서브 루틴 등을 테스트 하는 단계
2. 통합 테스트
   - 단위 테스트를 통과한 모듈 사이의 인터페이스 통합된 컴포넌트 간의 상호 작용을 검증하는 테스트 단계
3. 시스템 테스트
   - 통합된 단위 시스템의 기능이 시스템에서 정상적으로 수행되는지를 검증하는 테스트 단계
4. 인수 테스트
   - 계약상의 요구사항이 만족되었는지 확인하기 위한 테스트 단계

테스트 계획서
– 테스트 수행을 계획한 문서

테스트 베이시스
– 테스트 설계를 위한 기준이 되는 문서

테스트 케이스
– 입력값, 실행조건, 기대 결과로 구성된 테스트 항목의 명세서

테스트 슈트
– 테스트 케이스의 집합

테스트 시나리오
– 테스트가 필요한 상황을 작성한 문서

테스트 스크립트
– 테스트 케이스의 실행 순서를 작성한 문서

테스트 결과서
– 테스트 결과를 정리한 문서