
2018학년도
정보처리 산업기사 외부평가
대비용 교재



차 례

1. 애플리케이션 배포	3
2. 프로그래밍 언어 활용	20
3. 응용 SW 기초 기술 활용	30
4. 화면구현	49
5. SQL 활용	61
6. 애플리케이션 테스트 수행	73
7. UI테스트	88
8. 2018학년도 정보처리 산업기사 외부평가 필기시험(2018.09.08.)	108
9. 2018학년도 정보처리 산업기사 외부평가 필기시험(2018.10.13.)	112
10. 2018학년도 정보처리 산업기사 외부평가 구술시험(1,2차)	116
11. 2018학년도 정보처리 산업기사 외부평가 실기시험(2018.09.08.)	117
12. 2018학년도 정보처리 산업기사 외부평가 실기시험(2018.10.14.)	122

1. 애플리케이션 배포(2001020214_16v4)

1

애플리케이션 배포 환경 구성하기

- 애플리케이션 빌드와 배포를 위한 환경 구성 방안을 계획할 수 있다.
- 애플리케이션 배포를 위한 도구와 시스템을 결정할 수 있다.
- 결정한 애플리케이션 배포 환경을 위한 도구와 시스템을 설치할 수 있다.
- 설치한 시스템과 도구 운영을 위해 상세 구성 및 설정을 할 수 있다.

[1] 모듈

1. 모듈(화)의 개념

소프트웨어 설계에서 기능단위로 분해하고 추상화되어 재사용 및 공유 가능한 수준으로 만들어진 단위를 모듈로 규정하고, 소프트웨어의 성능을 향상시키거나 시스템의 디버깅, 시험, 통합 및 수정을 용이하도록 하는 소프트웨어 설계 기법을 모듈화라고 한다.

2. 모듈화의 장점

- 프로그램의 효율적인 관리 및 성능 향상
- 전체적인 소프트웨어 이해의 용이성 증대 및 복잡성 감소
- 소프트웨어 시험, 통합, 수정 시 용이성 제공
- 기능의 분리가 가능하고 인터페이스가 단순
- 오류의 파급 효과를 최소화
- 모듈의 재사용 가능으로 개발과 유지보수가 용이

3. 모듈화의 목표

- 모듈 간 결합도의 최소화
- 모듈 내 요소들 간의 응집도 최대화

[2] 소프트웨어 빌드

1. 소프트웨어 빌드 개념

- 소프트웨어 빌드는 소스 코드 파일을 컴퓨터에서 실행할 수 있는 제품 소프트웨어의 단위로 변환하는 과정을 말하거나 그에 대한 결과물을 일컫는다.
- 소프트웨어 빌드는 소스 코드 파일이 실행 코드로 변환되는 컴파일 과정을 핵심으로 수행되며, 빌드에 따른 결과물에 대한 상세 확인이 필요하다.

2. 빌드 도구 종류

Ant, Make, Maven, Gradle 등

1) Ant

- (1) 앤트는 공통적인 프로젝트 디렉토리 구조와 같은 공식적인 관례를 가지고 있지 않으며, 어디에서 소스를 찾을 수 있고, 어디에 산출물을 두어야 하는지 정확하게 명시하여야 한다. [시간낭비, 결과물 비체계화]
- (2) 앤트는 절차적이며 항상 앤트에게 '컴파일하고 복사하고 압축한다'와 같이 무엇을 언제 할 것인지 정확하게 알려주어야 한다.
- (3) 앤트는 goal과 goal 디펜던시를 정의할 수 있는 각 라이프사이클을 가지고 있지 않다.

따라서 각 goal에 태스크 순서를 붙여주어야 한다.

2) Maven

메이븐은 Make나 Ant와 같은 툴을 단순히 향상시킨것이 아니라 소프트웨어 프로젝트와 소프트웨어 개발에 관련된 새로운 시맨틱(Semantics)들을 포괄하는 플랫폼이다.

(1) 디펜던시(dependency)관리

그룹식별자, 산출물 식별자와 버전으로 구성된 중복되지 않은 코디네이트들을 프로젝트에 정의한다.

(2) 원격저장소

의존성 관리와 관련하여, 메이븐 산출물들의 저장소를 생성하기 위하여 메이븐 프로젝트 객체 모델(Project Object Model, POM)안에 정의된 코디네이트를 사용할 수 있다.

(3) 빌드 로직의 전역적인 재사용

플러그인은 POM에 따라 실행될 수 있도록 개발되었는데, 플러그인의 동작 시 어떠한 외부 파일도 참조하지 못하도록 설계되어 있다.

(4) 툴에 대한 이식성과 통합

이클립스나 넷빈즈 그리고 IntelliJ와 같은 툴은 프로젝트에 대한 정보를 찾을 수 있는 공통 공간을 가지고 있다.

(5) 프로젝트 산출물들의 쉬운 검색과 필터링 : 넥서스와 같은 툴은 POM안의 정보 저장소를 사용하여 저장소의 내역들을 인덱스하고 검색하는 것을 허용한다.

3) gradle

- Groovy, remote repository를 가져올 수 있음
- convention 정책과 함께 custom 빌드 스크립트(groovy) 작성가능
- 멀티플랫폼 프로젝트에도 적합(상위 스크립트, 하위 스크립트 독립적으로 작성)
- 상속이 아닌 주입

[3] 애플리케이션 빌드(Build)

1. 애플리케이션 빌드(Build) 절차 : 소스 코드 관리 저장소에서 코드를 가져옴
 - 1) 코드를 컴파일하고 종속성, 모듈을 확인함.
 - 2) 자동화 된 단위 테스트 실행
 - 3) 라이브러리, 코드, 파일 등을 적절하게 링크함.
 - 4) 성공적으로 완료하면 이슈를 빌드하고 저장함.
 - 5) 빌드 로그를 보관함.

[4] 배포

1. 배포 스크립트

배포 스크립트는 빌드를 랩 환경의 컴퓨터로 복사하는 BAT 또는 CMD 파일이다. 빌드에 설치 패키지가 포함된 경우 배포 스크립트를 사용하여 설치 패키지를 실행할 수도 있다. 배포 스크립트를 실행하면 빌드 컨트롤러에서 사용자 랩 환경의 지정된 컴퓨터에 있는 작업 디렉터리에서 해당 명령을 실행한다.

2. 배포 스크립트를 만들고 사용하는 절차

1) 배포용 빌드 파일 준비

설치 유형	설명
웹 서비스 배포 패키지	배포 스크립트에서 빌드 파일과 배포 패키지를 랩 환경의 대상 컴퓨터로 복사한 다음 배포 패키지를 실행해야 한다. 이렇게 하려면 배포 스크립트에 도메인 이름, 서비스 계정 이름 및 TFS 서버 이름을 매개 변수로 전달해야 한다. 이 유형의 설치에서는 일반적으로 응용 프로그램을 배포하기 전 웹 서버, 데이터베이스 서버와 같이 필수 구성 요소 소프트웨어를 수동으로 설치 및 구성해야 한다. 필수 구성 요소 소프트웨어를 구성하려면 대상 컴퓨터에서 응용 프로그램을 배포할 수 있도록 하는 특정 작업이 필요할 수 있다. 예를 들어, IIS를 사용하는 응용 프로그램의 경우 응용 프로그램을 배포하기 전 랩 환경의 IIS 서버에 웹 배포 에이전트 서비스가 구성되어 있어야 한다.
설치 패키지가 없는 빌드 파일	배포 스크립트는 응용 프로그램 및 빌드 파일을 랩 환경의 대상 컴퓨터로 복사해야 한다.
설치 패키지	배포 스크립트는 설치 패키지를 랩 환경의 대상 컴퓨터로 복사한 다음 설치 패키지를 실행해야 한다. 배포 스크립트에서 설치 패키지를 실행하면 자동 설치인지 확인해야 한다. 설치에서 요청된 사용자 입력은 배포 스크립트에서 처리해야 한다. 여기에는 자격 증명이 포함됩니다. 배포 스크립트에서 설치 패키지를 실행할 때 모든 사용자 입력을 처리하지 않을 경우 응용 프로그램 배포가 실패한다. 웹 응용 프로그램을 배포하지 않고 배포에 설치 패키지가 필요할 경우 InstallShield 설치 패키지를 사용하는 것이 좋다. InstallShield에 대한 자세한 내용은 Windows Installer 배포 를 참조.

응용 프로그램을 설치하기 위해 선택하는 방법에 따라 배포 스크립트가 완료해야 하는 기본 작업이 결정된다. 응용 프로그램을 설치할 때 EXE를 클라이언트 컴퓨터에 복사하는 경우도 있고 여러 컴퓨터에 파일을 설치하는 배포 패키지를 실행하는 경우도 있다.

2) 배포 스크립트 작성

- ① 빌드 컨트롤러에서 빌드 경로를 가져옵니다. 이 경로를 배포 스크립트에 명령 인수로 보낼 수 있다.
- ② 배포 경로를 지정한다.
- ③ 배포 디렉토리를 만듭니다. 또한 배포 스크립트에 작성하지 않고 수동으로 할 수 있다. 워크플로를 사용하여 배포 전 환경 스냅샷을 사용할 경우 스냅샷의 가상 환경에 디렉토리를 만들지만 하면 된다.
- ④ 빌드 경로에서 배포 경로로 배포 패키지를 복사한다.
- ⑤ 배포 디렉터리에 배포 패키지를 실행한다.

3) 배포 스크립트 빌드

배포 스크립트를 만든 후에는 버전 제어에 체크인한 다음 빌드 출력으로 복사되도록 구성해야 한다. 배포 스크립트를 빌드하려면 우선 솔루션뿐만 아니라 Visual Studio 프로젝트에도 저장되었는지 확인한다. 이렇게 하려면 Visual Studio에서 배포 스크립트를 선택한 다음 솔루션 탐색기의 속성 아래에서 출력 디렉터리로 복사를 항상 복사로 변경한다.

4) 작업 디렉터리 설정

빌드-배포-테스트 워크플로에 추가하는 각 명령 집합에 대해 작업 디렉터리를 지정할 수 있다. 작업 디렉터리를 지정하면 빌드 컨트롤러가 해당 역할에 할당된 각 컴퓨터의 작업 디렉터리에서 지정된 명령을 실행한다. 빌드-배포-테스트 워크플로에 명령을 추가할 때 작업 디렉터리를 지정하지 않으면 빌드 컨트롤러가 기본 디렉터리인 C:\Windows\System32에서 명령을 실행한다.

작업 디렉터리를 지정하는 경우 배포 스크립트가 디렉터리로 파일을 복사하기 전에 해당 디렉터리가 만들어졌는지 확인해야 한다. 배포 스크립트가 작업 디렉터리로 파일을 복사하기 전에 작업 디렉터리가 만들어지도록 하려면 랩 환경의 대상 컴퓨터에 디렉터리를 수동으로 만들거나 어떤 명령이 디렉터리로 파일을 복사하기 전에 배포 스크립트에 의해 디렉터리가 만들어져야 한다.

5) 배포 관리 도구에 배포 스크립트 추가

[5] 형상 관리

1. 형상 관리 개념

소프트웨어에서 형상 관리는 개발 중 발생하는 모든 산출물들이 변경됨으로써 점차 변해가는 소프트웨어 형상을 체계적으로 관리하고 유지하는 기법이다.

2. 버전 관리 항목

버전 관리 항목	항목 설명
가져오기(Import)	버전 관리 되지 않은 로컬 디렉토리 파일을 처음으로 저장소에 복사
체크아웃(Check-out)	저장소 파일 받기
체크인(Checkin)	저장소에 새로운 버전으로 갱신
커밋(Commit)	체크인 시 이전 갱신 사항이 있는 경우 충돌 알림, diff 도구 이용 수정, commit 과정 수행
저장소(Repository)	파일의 현재 버전과 변경 이력 정보를 저장하는 저장소

3. 버전 등록 순서



4. 버전 관리 도구 유형

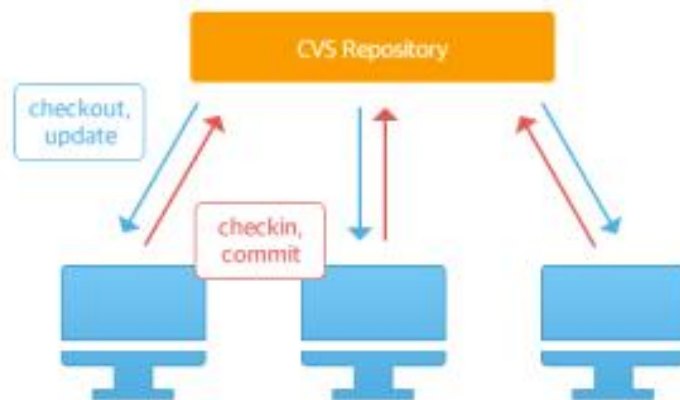
버전 관리 도구 유형	버전 관리 도구 설명
공유 폴더 방식	<ul style="list-style-type: none"> 매일 개발 완료 파일은 약속된 위치의 공유 폴더에 복사 담당자 한 명이 매일 공유 폴더의 파일을 자기 PC로 복사하고 컴파일 하여 에러 확인과 정상 동작 여부 확인 정상 동작일 경우 다음날 각 개발자들이 동작 여부 확인
클라이언트/서버방식	<ul style="list-style-type: none"> 중앙에 버전 관리 시스템이 항상 동작 개발자들의 현재 작업 내용과 이전 작업내용 추적 용이 서로 다른 개발자가 같은 파일을 작업했을 때 경고 출력 Trac이나 CVS view와 같은 GUI 툴을 이용 모니터링 가능
분산 저장소 방식	<ul style="list-style-type: none"> 로컬 저장소와 원격저장소 구조 중앙의 저장소에서 로컬에 복사한 순간 개발자 자신만의 로컬저장소에 생성 개발 완료한 파일 수정 이후 로컬 저장소에 커밋한 이후 다시 원격 저장소에 반영하는 방식

5. 버전 관리 도구 종류

구분	유형	예시
저장소 구분	로컬 버전 관리 시스템	rsc
	중앙 집중형 버전 관리 시스템	CVS, SVN, Clear Case
	분산형 버전 관리 시스템	Git, Mercurial
소스 공개 유형	Open Source 툴	CVS, SVN
	상용 버전 관리 툴	PVCS, Clear Case

1) CVS

CVS(Concurrent Versions System)은 가장 오래된 형상 관리 도구 중의 하나로서, 중앙 집중형 서버 저장소를 두고 클라이언트가 접속해서 버전 관리를 실행한다.



CVS	
주요기능	<ul style="list-style-type: none"> • checkout/checkin, update/commit 등 서버 저장소와 클라이언트의 변경 사항 전송 • diff를 통한 파일 내용 비교 • 파일 단위의 변경 사항 관리
장점	<ul style="list-style-type: none"> • 직관적이고 비교적 단순한 명령 세트
단점	<ul style="list-style-type: none"> • 유니코드 파일명 지원 부족 • 텍스트 기반 소스 코드만 지원 • 등록된 파일이나 디렉토리의 이름 변경이나 이동이 불편 • 커밋 실패시 롤백 미지원

2) SVN

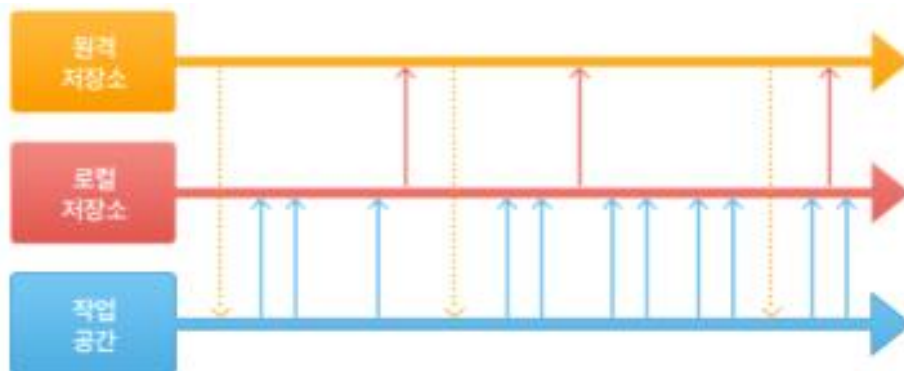
SVN(Subversion)은 CVS와 같은 중앙 집중형 클라이언트-서버 방식이지만 CVS의 단점을 보완하여 현재 가장 널리 사용되고 있는 형상 관리 도구이다. SVN은 CVS를 대체하기 위해 만든 도구로서 그 개념과 사용법이 CVS와 거의 유사해서, CVS 사용자가 어렵지 않게 SVN을 도입해서 사용할 수 있다.



SVN	
주요기능	<ul style="list-style-type: none"> • checkout/checkin, update/commit 등 서버 저장소와 클라이언트의 변경 사항 전송 • diff를 통한 파일 내용 비교 • 바이너리 문서 형상 관리 • 작업 단위의 변경 사항 관리 • atomic commit • svn:ignore를 통한 형상 관리 배제 자원 지정 기능
장점	<ul style="list-style-type: none"> • CVS 사용자가 쉽게 적응 가능 • 디렉토리나 파일을 자유롭게 이동해도 이력 유지 • gzip를 통한 압축으로 저장 공간 절약 • CVS에 비해 빠른 속도 • atomic commit으로 커밋 실패 시 롤백 지원 • 다양한 써드 파티 GUI 도구 존재
단점	<ul style="list-style-type: none"> • trunk, branch, tag가 모두 물리적인 저장 위치를 점유하므로 비효율적 • Git에 비해 branch, tag 작업이 무거움

3) Git







Git은 앞서 나왔던 CVS나 SVN과는 개념이 많이 다른 형상 관리 도구로서, 공개 SW 커뮤니티를 중심으로 사용 저변이 많이 확대되고 있다. Git는 중앙 집중형 방식이 아닌 분산형 방식으로 각 로컬 PC에 스스로 완전한 저장소가 구성되며, 필요에 따라 중앙 집중형 방식으로 운영할 수 있다.



Git	
주요기능	<ul style="list-style-type: none"> • branch, checkout, commit, tag 등 로컬 환경에서의 형상 관리 기능 • push, fetch, pull 등 원격 환경에서의 변경 사항 전송 기능 • 변경은 했지만 커밋에는 포함하지 않을 수 있는 staging 기능 • SVN으로 관리되던 저장소를 Git로 전환해주는 마이그레이션 기능 • diff를 통한 파일 내용 비교

	<ul style="list-style-type: none"> • 바이너리 문서 형상 관리 • 작업 단위의 변경 사항 관리 • .gitignore를 통한 형상 관리 배제 자원 지정 기능
장점	<ul style="list-style-type: none"> • branch 생성, 이동, 병합이 매우 가벼우므로 branch를 자주 사용하여 상황에 맞게 자주 분기하고 합칠 수 있어 코드 꼬임에 따른 위험 감소 • 각 로컬에 완전한 로컬 저장소가 있으므로 원격 저장소에 장애가 나더라도 쉽게 복구 가능 • 여러번의 커밋을 로컬 저장소에 실행하고, 모아진 커밋을 원격 저장소에 반영할 수 있으므로 네트워크 빈도는 줄고 속도는 항상 Pack 방식의 압축으로 SVN에 비해 저장 공간 절약 • 다양한 써드파티 GUI 도구 존재
단점	<ul style="list-style-type: none"> • CVS, SVN과 기본 개념이 많이 다르므로 적응에 시간 필요 • checkout, commit 등 텍스트는 같지만 의미나 동작이 SVN이나 CVS와 다른 명령어가 있어 혼란 야기 • 빈 디렉터리가 저장되지 않음

4) 추천가이드

적용조건	추천도구
<ul style="list-style-type: none"> • 개발자가 클라이언트-서버, 중앙집중형 형상 관리 도구만 사용해온 경우 	
<ul style="list-style-type: none"> • 개발자가 Mercurial 등 분산형 형상 관리 도구를 사용한 경험이 있는 경우 	
<ul style="list-style-type: none"> • 개발자가 각각 다른 공간 및 네트워크에서 하나의 서버 저장소를 공유하기 어려운 공유 	
<ul style="list-style-type: none"> • 개발자가 동일한 네트워크 내에서 개발하는 경우 	
<ul style="list-style-type: none"> • 소스의 변경이 잦지 않고 안정적으로 운영되고 있는 서비스의 경우 	
<ul style="list-style-type: none"> • 소스의 변경이 잦고 현재 개발이 활발하게 진행되고 있는 경우 	

6. 형상관리 명령어

1) GIT(버전관리, 형상관리 툴) 명령어

git config --global --list

현재 설정정보 조회할 수 있습니다.

--global 옵션은 전역설정에 대한 옵션이며 현재 프로젝트에만 적용할 때는 주지 않습니다.

git config --global user.name "사용자명"

사용자명을 등록합니다 (필수)

git config --global user.email "이메일주소"

이메일 주소를 등록합니다. (필수)

git config --global color.ui "auto"

터미널에 표시되는 메시지에 칼라를 표시해줌

- 2) git에 대해서 사전에 알아야 될 부분은 아래와 같습니다.
 - git의 저장소는 3가지 단계로 나누어 집니다.
커밋한 소스가 보관되는 저장소와 현재 프로젝트 파일들이 있는 작업트리, 저장소와 작업트리 사이의 버퍼영역으로 커밋될 대상이 저장되는 스테이징 영역입니다.
 - git은 빈 디렉토리는 추적하지 않습니다.
 - 형상관리를 하지 않을 파일은 .gitignore 파일에 추가합니다.
 - HEAD는 현재 브랜치의 가장 최신 커밋을 의미한다.
 - 기본원격 저장소를 origin이라고 부릅니다.

3) git 명령어

기본적인 명령어

git --version

현재 git의 버전을 확인합니다.

git init

현재 디렉토리에 git 저장소를 생성합니다.

git add 파일명

git add는 2가지를 하는데 untracked files의 파일들을 git가 추적하도록 하거나 파일은 수정했지만 아직 스테이징 영역에 올라가지 않은(Changed but not updated) 파일들을 스테이징 영역에 올립니다. -i 옵션을 주면 대화형모드가 시작되며 파일의 일부분만 선택해서 스테이징하는 것이 가능합니다. -p 옵션을 사용하면 -i 대화형 모드 없이 바로 패치모드를 사용할 수 있습니다.

git commit -m "커밋메시지"

스테이징 영역에 올라가 있는 파일들을 커밋합니다. -m 은 커밋 메시지를 주는 옵션으로 여러 줄의 커밋 메시지를 쓸 경우 -m 을 여러개 사용할 수 있습니다. -a 옵션을 사용하면 스테이징에 올리는 작업과 커밋을 동시에 할 수 있습니다.(추적되지 않는 파일은 추가하지 않습니다.) -m을 사용하지 않을때 -v옵션을 사용하면 편집기에 커밋하려는 변경사항의 다른 점을 보여줍니다. 특정파일만 커밋하려면 마지막에 파일명을 추가해주면 됩니다.

git commit -C HEAD -a --amend

지정한 커밋의 로그메시지를 다시 사용하여 기존커밋을 수정합니다. -c를 사용하면 기존메시지를 수정할 수 있는 편집기를 실행해 줍니다.

git status

커밋되지 않은 변경사항을 조회합니다.

git diff

스테이징영역과 현재 작업트리의 차이점을 보여줍니다. --cached 옵션을 추가하면 스테이징영역과 저장소의 차이점을 볼 수 있다. git diff HEAD를 입력하면 저장소, 스테이징영역, 작업트리의 차이점을 모두 볼 수 있다. 파라미터로 log와 동일하게 범위를 지정할 수 있으며 --stat를 추가하면 변경사항에 대한 통계를 볼 수 있습니다.

git mv 파일명 새파일명

기존에 존재하는 파일을 새파일로 이동합니다. 변경이력은 그대로 유지합니다.

git checkout -- 파일명

아직 스테이징이나 커밋을 하지 않은 파일의 변경내용을 취소하고 이전 커밋상태로 돌립니다. svn에서 revert와 동일합니다.

Branch와 Tag

git branch

현재 존재하는 브랜치를 조회합니다. -r 옵션을 사용하면 원격저장소의 브랜치를 확인할 수 있습니다.

git branch 브랜치명B 브랜치명A

브랜치명A에서 새로운 브랜치 브랜치명B를 만듭니다. (git에서 기본 브랜치는 master라는 이름을

사용합니다.)

git branch 브랜치명

브랜치명의 새로운 브랜치를 만듭니다.(체크아웃은 하지 않습니다.)

git branch -d 브랜치명

브랜치를 삭제합니다.

git branch -m 존재하는브랜치명 새로운브랜치명

존재하는 브랜치를 새로운브랜치로 변경합니다. 이미 존재하는 브랜치명이 있을 경우에는 에러가 나는데 -M 옵션을 사용하면 이미 있는 브랜치의 경우에도 덮어씁니다.

git tag 태그명 브랜치명

브랜치명의 현재시점에 태그명으로 된 태그를 붙입니다. git tag만 입력하면 현재 존재하는 태그 목록을 볼 수 있습니다.

git checkout 브랜치명/태그명

해당 브랜치나 태그로 작업트리를 변경합니다.

git checkout -b 브랜치명B 브랜치명A

브랜치명A에서 브랜치명B라는 새로운 브랜치를 만들면서 체크아웃을 합니다.

git rebase 브랜치명

브랜치명의 변경사항을 현재 브랜치에 적용합니다.

git merge 브랜치명

브랜치명의 브랜치를 현재 브랜치로 합칩니다. --squash 옵션을 주면 브랜치명의 모든 커밋을 하나의 커밋으로 만듭니다.

git cherry-pick 커밋명

커밋명의 특정 커밋만을 선택해서 현재 브랜치에 커밋으로 만듭니다. -n 옵션을 주면 작업트리에 합치지만 커밋은 하지 않기 때문에 여러개의 커밋을 합쳐서 커밋할 수 있습니다.

로그관리

git log

커밋로그들을 볼 수 있으면 -1나 -2같은 옵션을 주어 출력할 커밋로그의 갯수를 지정할 수 있습니다. --pretty=oneline 옵션을 주면 한줄로 간단히 보여주고 --pretty=format:"%h %s"처럼 형식을 정해줄 수 있습니다. -p 옵션을 사용하면 변경된 내용을 같이 보여줍니다. --since="5 hours" 이나 --before="5 hours"같은 옵션도 사용가능합니다. --graph 옵션을 주면 브랜치 트리를 볼 수 있습니다.

git log 커밋명

해당 커밋명의 로그를 볼 수 있습니다. 커밋명A..커밋명B (마침표2개)와 같이 입력하면 커밋명A이후 부터 커밋명B까지의 로그를 볼 수 있습니다. ^은 -1과 동일해서 HEAD^라고 하면 최신 바로 이전 커밋이고 HEAD^^와 같이 쓸 수 있으며 HEAD~3을 하면 HEAD의 3개 이전의 커밋을 뜻합니다.

git blame 파일명

갈 줄 앞에 커밋명과 커밋한 사람 등의 정보를 볼 수 있습니다.

git blame -L 10,15 파일명

-L 옵션을 사용하면 10줄부터 15줄로 범위를 지정해서 볼수 있고 15대신 +5와 같이 사용할 수 있습니다. 숫자의 범위 대신 정규식도 사용이 가능합니다.

git blame -M 파일명

-M 옵션을 사용하면 반복되는 패턴을 찾아서 복사하거나 이동된 내용을 찾아줍니다. -C -C 옵션을 사용하면 파일간의 복사한 경우를 찾아줍니다. -C -C는 git log에서도 사용가능하며 내용의 복사를 찾을 때는 git log에서 -p옵션을 사용합니다.

git revert 커밋명

기존의 커밋에서 변경한 내용을 취소해서 새로운 커밋을 만듭니다. -n옵션을 사용하면 바로 커밋하지 않기 때문에 revert를 여러번한 다음에 커밋할 수 있습니다.(항상 최신의 커밋부터 revert해야 합니다.)

git reset 커밋명

이전 커밋을 수정하기 위해서 사용합니다. --soft 옵션을 사용하면 이전 커밋을 스테이징하고 커밋은 하지 않으며 --hard 옵션은 저장소와 작업트리에서 커밋을 제거합니다. git reset HEAD^와 같이 입력하면 최근 1개의 커밋을 취소할 수 있습니다.

git rebase -i 커밋범위

-i 옵션으로 대화형모드로 커밋 순서를 변경하거나 합치는 등의 작업을 할 수 있습니다.

원격저장소**git clone 저장소주소 폴더명**

원격저장소를 복제하여 저장소를 생성합니다. 폴더명을 생략 가능합니다.

git fetch

원격저장소의 변경사항 가져와서 원격브랜치를 갱신합니다.

git pull

git fetch에서 하는 원격저장소의 변경사항을 가져와서 지역브랜치에 합치는 작업을 한꺼번에 합니다. 파라미터로 풀링할 원격저장소와 반영할 지역브랜치를 줄 수 있습니다.

git push

파라미터를 주지 않으면 origin 저장소에 푸싱하며 현재 지역브랜치와 같은 이름의 브랜치에 푸싱합니다. --dry-run 옵션을 사용하면 푸싱된 변경사항을 확인할 수 있습니다. 로컬에서 tag를 달았을 경우에 기본적으로 푸싱하지 않기 때문에 git push origin 태그명이나 모든 태그를 올리기 위해서 git push origin --tags를 사용해야 합니다.

git remote add 이름 저장소주소

새로운 원격 저장소를 추가합니다.

git remote

추가한 원격저장소의 목록을 확인할 수 있습니다.

git remote show 이름

해당 원격저장소의 정보를 볼 수 있습니다.

git remote rm 이름

원격저장소를 제거합니다.

서브모듈**git submodule**

연관된 하위모듈을 확인할 수 있습니다.

git submodule add 저장소주소 서브모듈경로

새로운 하위모듈을 해당경로에 추가합니다. 추가만하고 초기화 하지는 않으며 커밋해쉬앞에 마이 나스(-) 표시가 나타납니다.

git submodule init 서브모듈경로

서브모듈을 초기화 합니다.

git submodule update 서브모듈경로

서브모듈의 변경사항을 적용합니다.(저장소의 최신커밋을 추적하지 않습니다.)

기타 명령어**git archive --format=tar --prefix=폴더명/ 브랜치혹은태그 | gzip > 파일명.tar.gz****git archive --format=zip --prefix=폴더명/ 브랜치혹은태그 > 파일명.zip**

해당 브랜치나 태그를 압축파일로 만듭니다. --prefix를 주면 압축하일이 해당폴더 안에 생성되도록 할 수 있습니다.

git mergetool

설정에 merge.tool의 값에 있는 머지툴을 찾아서 실행합니다.

git gc

저장소의 로그를 최적화 합니다. 로그가 변경되지는 않고 저장하는 방식만 최적화 합니다.

--aggressive 옵션을 주면 더 자세하게 최적화합니다.

git rev-parse --show-toplevel

git 저장소내에서 입력하면 루트디렉토리를 알려줍니다.

7. 버전 관리 도구 사용 유의점

- 형상 관리 지침에 의거 버전에 대한 정보를 언제든지 접근할 수 있어야 한다.
- 제품 소프트웨어 개발자, 배포자 이외에 불필요한 사용자가 소스를 수정할 수 없도록 해야 한다.
- 버전 관리의 기본 개념: 동일한 프로젝트에 대해서 여러 개발자가 동시에 개발할 수 있어야 한다.
- 에러 발생 시 최대한 빠른 시간 내에 복구한다.

[6] 프로젝트 관리도구

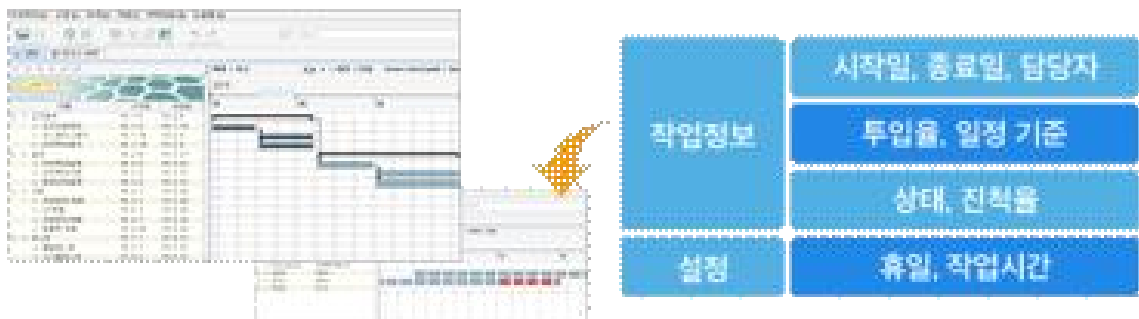
프로젝트 관리 도구는 범위 관리, 일정 관리, 예산 관리, 의사소통, 계획관리, 공정관리, 자원관리 등 프로젝트를 관리하는 활동을 편리하게 할 수 있게 해준다.

1. Redmine



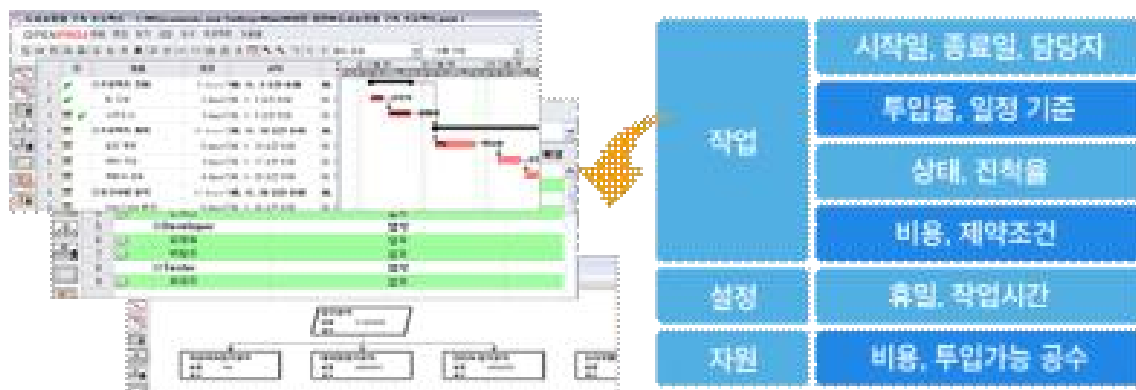
Redmine	
주요기능	<ul style="list-style-type: none">• 일감 관리(등록, 변경, 추적, 통보 등) 문서관리, 파일관리, 위키 사이트 구축• 이메일 통보
장점	<ul style="list-style-type: none">• 간단한 일정관리 가능• 웹서버를 이용하여 다수의 인원이 협업할 수 있음• 프로젝트, 사용자 권한 등을 관리할 수 있음• 일감의 진행에 따른 이메일 통보 가능
단점	<ul style="list-style-type: none">• 자원의 투입율을 관리할 수 없음• 휴일이나 일단위 작업시간을 설정할 수 없음• Critical Path 식별을 위한 선후행 작업의 조건설정이 불가함

2. GanttProject



GanttProject	
주요기능	<ul style="list-style-type: none"> • 작업 관리(등록, 변경 등) • 휴일, 작업시간 설정 기능 • 인적자원 관리 기능
장점	<ul style="list-style-type: none"> • 작업자 투입율과 연동하여 일정관리 가능 • Gantt 차트, 리소스 차트 지원 • 휴일 달력 사용 기능
단점	<ul style="list-style-type: none"> • 팀 협업이 어려움 • 전체적인 사용성이 좋지 못함 • 일부 기능에는 한글사용이 원활하지 못함

3. OpenProj



OpenProj	
주요기능	<ul style="list-style-type: none"> 작업 관리(등록, 진척확인, 변경이력관리 등) 휴일, 작업시간 설정 기능 인적자원 관리 기능 예산기반 WBS 작업 기능, Earned value 기반 관리기능
장점	<ul style="list-style-type: none"> 작업자 투입율과 연동하여 일정관리 가능 Gantt 차트, 리소스 차트 지원 프로젝트 관리 상용도구 수준의 기능, UI 지원 상용프로그램(MSProject)과 호환 가능
단점	<ul style="list-style-type: none"> 팀 협업이 어려움 간략하게 관리하려는 경우는 사용방법이 복잡할 수 있음

4. 추천가이드

적용조건	추천도구
<ul style="list-style-type: none"> 간단하게 작업의 시작/종료일자, 담당자, 종료여부 등 개괄적인 일정 관리하고자 하는 경우 	
<ul style="list-style-type: none"> 프로젝트 WBS 뿐만 아니라 진행중에 발생하는 단발성 작업(Action Item)이나 이슈 등을 등록하고 관리하고자 하는 경우 	
<ul style="list-style-type: none"> 여러 작업자간 일정 및 이슈 공유하고 진행상황을 통보하고자 할 경우 작업 진행에 대한 등록과 진척결과를 시스템에 작업자가 입력하고 시스템에서 전체 상황을 조회하고자 하는 경우 시작프로젝트 관리 이외의 기능이 필요하거나 외부 도구와 연동할 필요가 있는 경우 	
<ul style="list-style-type: none"> 휴일, 투입시간을 고려하여 일정계획/관리 하는 경우 작업자 별로 투입되는 시간이 미달되거나 초과되지 않도록 관리 하고자 하는 경우 	
<ul style="list-style-type: none"> 프로젝트 일정 수립 시 예산과 비용을 고려하여 수립하고자 하는 경우 계획과 진척관리 시 비용, 공수 등 엄밀하게 관리 하는 경우 일정계획이 변경되는 이력을 유지관리 하는 경우 	

2

애플리케이션 소스 검증하기

- 정상적으로 작동하는 소프트웨어 빌드를 위해 형상관리 서버로부터 소스코드를 체크 아웃할 수 있다.
- 소스코드 검증 도구를 활용하여 애플리케이션에서 사용한 라이브러리, 소스, 로직 등의 오류가 있는지 여부를 검증할 수 있다.
- 소스 코드의 환경 설정, 운영 환경 정보, 대상 시스템 정보 등에 오류가 있는지 확인할 수 있다.

[1] 테스트링과 디버깅의 차이

	테스팅(Testing)	디버깅(Debugging)
목적	알려지지 않은 결함의 발견	이미 알고 있는 결함의 수정
수행	시스템 내부 관련자 테스팅 팀 등 외부의 제 3자	시스템 내부 관련자
주요 작업	숨겨진 결함의 발견	결함의 정확한 위치 파악 결함의 타입 식별 결함의 수정

[2] 결함을 지칭하는 용어

- 에러(Error), 오류: 사람에 의해 생성된 실수
- 결함(Defect), 결점(Fault), 버그(Bug): 결함이 실행될 때 발생하는 현상
- 실패(Failure), 문제(Problem): 에러가 원인이 되어 제품에 포함된 결함.
소프트웨어 개발 산출물에서 잘못된 사항.

[3] 테스트케이스

- 의미: 테스트의 목적에 맞게 테스트 조건, 입력값, 예상 출력값, 실제 테스트 결과를 기록하는 것

2. 목적: 테스터가 테스트를 체계적으로 할 수 있도록 함.
개발자가 테스트 결과를 통해 디버깅을 하는 기준이 됨

[4] 테스트 종류

1. 블랙박스 테스트

- 1) 개념: 요구사항 명세서나 설계서를 참조하면서 수행하는 테스트
- 2) 방법
 - ① 신택스 테스트(Syntax Testing)
 - 블랙박스 테스트 기법 중 가장 단순한 방법
 - 입력 데이터가 미리 정의된 데이터 유형에 적합한지를 검증하는 방법
 - 입력 값을 적합(Valid)과 부적합(Invalid)으로 분류한 뒤, 예상되는 결과를 검증하는 기법
 - ② 동등분할(Equivalence Partitioning)
 - 입력값이 범위가 정해져 있을 경우, 각 범위의 대표값을 이용하여 테스트
 - ③ 경계값 분석(Boundary Value Analysis)
 - 입력 값의 주요 오류 대상인 경계값을 입력값으로 테스트 케이스를 작성하여 테스트
 - ④ 의사결정 테이블(Decision Table)
 - 입/출력값이 True, False로 결정될 수 있는 경우 모든 경우의 수를 확인해볼 수 있는 방법
 - ⑤ 상태 전이 테스트(State transition testing)
 - 프로그램의 각 상태를 다이어그램으로 나타내 상태간의 전이에 필요한 입력과 그 입력에 대해 실제로 상태 전이가 올바르게 일어나는지를 테스트하는 방법으로, 일반적으로 유저 인터페이스 테스트에 유용하게 사용됨

2. 화이트박스 테스트

- 1) 개념: 소스코드를 직접 참조하면서 수행하는 테스트 기술
- 2) 특징: 내부구조(소스 코드)를 기반으로 테스트 케이스 추출
- 3) 방법
 - ① 문장 커버리지(Statement Coverage): 프로그램을 구성하는 문장들이 최소한 한번은 실행될 수 있는 입력 값을 테스트 케이스로 선정함
 - ② 분기 커버리지(Branch Coverage): 프로그램에 있는 분기를 최소한 한번은 실행하게 하는 테스트하는 방법
 - ③ 조건 커버리지(Condition Coverage): &&, || 등의 조건을 가진 분기문이 전체 조건식의 결과와 관계없이 &&나 || 전후의 각 개별 조건식이 참 한 번, 거짓 한 번을 갖도록 테스트 케이스를 만드는 방법
 - ④ 다중조건 커버리지(Multiple Condition Coverage): 조건 커버리지가 각 개별 조건식의 조건을 검사하는 것이라면, 다중조건 커버리지는 전체 조건식의 조건을 검사하는 테스트 케이스를 만드는 방법

3. 소스코드 정적 분석 / 동적 분석

- 1) 정적 분석 : 소프트웨어에 다양한 입력을 하여 실행시킨 후 결과를 살펴보는 방법
- 2) 동적 분석 : 소프트웨어를 실행시키지 않고 소스 코드의 내용으로 판단하는 방법

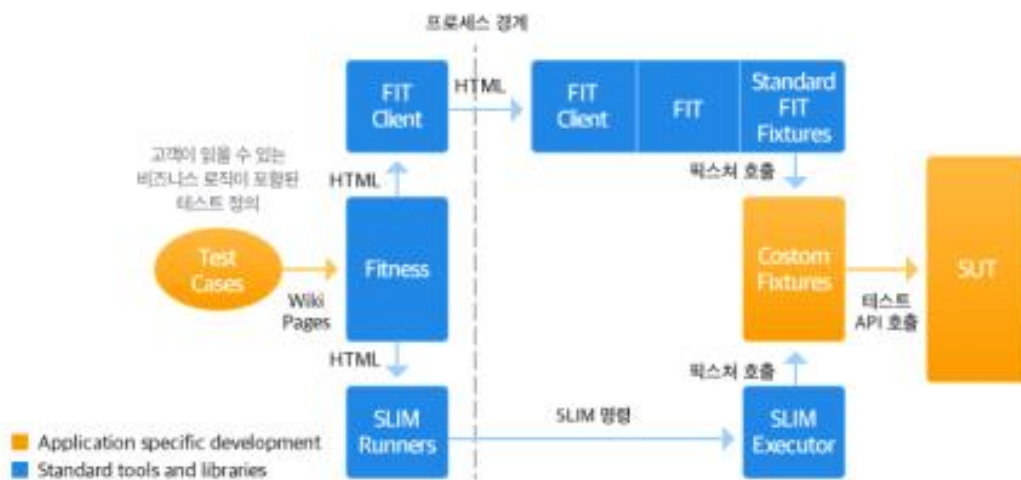
구분		내용
정적 분석	블랙박스 테스트	소스 코드와 소프트웨어 실행없이 테스트 명세서의 완결성, 정확성, 일관성 등을 확인
	화이트박스 테스트	소프트웨어 실행없이 소스 코드만으로 테스트 분석 도구로 소프트웨어의 오류 등을 확인
동적 분석	블랙박스 테스트	소스 코드와 소프트웨어 실행없이 테스트 테스트 케이스와 결과의 기대값을 만들어 기대하는 값이 나오는지 확인
	화이트박스 테스트	소프트웨어 실행없이 소스 코드만으로 테스트 디버깅, 유닛 테스트, 스크립트를 통한 테스트

[5] 웹 테스트 도구 비교

웹 기반 테스트는 테스트의 실행이 웹에서 진행되는 것을 말한다. 브라우저 동작을 모사할 수 있는 테스트 도구인 HttpUnit, Phantom.js, Casper.js 와 실제 브라우저를 구동해서 테스트를 실행하는 Selenium 이 웹 기반 테스트 범주에 포함되며, 넓게 보면 웹에서 테스트 케이스를 작성할 수 있는 Fitnesse도 포함될 수 있다.

1. Fitness

Fitness는 SW개발 협업 도구로서 wiki를 통한 의사소통을 지원하고, 인수 테스트를 자동화 해주는 프레임워크이다. 표(테이블) 형식으로 입력값, 파라미터, 결과값을 정의해서 개발자가 아닌 관련자들도 테스트 케이스를 작성하고 의사소통할 수 있다.



Fitness	
주요기능	<ul style="list-style-type: none"> • Wiki 페이지 생성 및 관리 • 테이블을 이용한 테스트 케이스 작성 • 엑셀/워드 업로드를 통한 테이블 및 테스트 케이스 작성 • 테스트 스위트를 통해 여러 Wiki 페이지에 있는 테스트를 모아서 실행 • Wiki 페이지 자체 내에 테스트 항목별 Green/Red 마크를 통해 테스트 결과 확인 • Wiki 기반으로 동작하는 테스트 코드 작성을 위한 Fixture 제공
장점	<ul style="list-style-type: none"> • 요구사항 명세 수준에서 쉽게 테스트 케이스를 작성할 수 있어 초기에 효율적인 의사소통 가능 • 하나의 Wiki 페이지에서 테스트 설계, 실행, 확인 모두 가능 • Selenium과 연계해서 Browser 테스트도 가능
단점	<ul style="list-style-type: none"> • Fitness Wiki 페이지 작성을 위한 문법 습득 필요 • 테스트 코드 작성을 위해 Fitness에서 제공하는 여러 가지 Fixture에 대한 습득 필요

2. HttpUnit

HttpUnit은 브라우저 없이도 API를 이용해 브라우저를 통해 서버에 접근하는 것처럼 모사해서 자바 웹 애플리케이션에 대한 브라우저 테스트를 자동화할 수 있게 해주는 도구이다.



HttpUnit	
주요기능	<ul style="list-style-type: none"> • WebConversation, WebRequest, WebResponse를 통해 브라우저 없이도 브라우저를 통해 웹 서버에 요청을 날리는 것처럼 모사 • Form을 통한 요청 기능 테스트 • ServletUnit를 통한 서블릿 모사 기능 • 쿠키, 페이지 리다이렉션 테스트
장점	<ul style="list-style-type: none"> • 단순한 API 세트로 쉽게 HTTP 연결 테스트 가능 • 사용하기 쉽고 낮은 학습 곡선 • 실제 브라우저를 띄우지 않으므로 테스트가 가벼움
단점	<ul style="list-style-type: none"> • CSS, JavaScript에 대한 브라우저 호환성 테스트 불가 • 브라우저 상에서의 액션 녹화 기능이 없어 모든 브라우저 테스트 코드를 수기로 작성해야 함 • JavaScript에 대한 지원이 제한적 • 2008년 이후에 업데이트가 되지 않고 있음 • Eclipse 플러그인이 없음

3. Selenium

Selenium은 Selenium IDE를 통해 Firefox 브라우저에서 사용자의 액션을 녹화하고, WebDriver를 통해 실제 브라우저를 실행해서 사용자의 액션을 재생해서 테스트할 수 있는 브라우저 테스트 자동화 도구이다.



Selenium	
주요기능	<ul style="list-style-type: none"> Firefox에서 Selenium IDE를 통해 사용자 액션 녹화 사용자 액션에 대한 테스트 코드 자동 생성 변경은 했지만 커밋에는 포함하지 않을 수 있는 staging 기능 실제 브라우저 실행을 통한 테스트 CSS Selector, XPath를 통한 화면 요소 제어
장점	<ul style="list-style-type: none"> Java, Ruby, Python, C#등 다양한 언어로 테스트 소스 생성 Maven과 연계해서 빌드 테스트로 활용 가능 Fitness 연계해서 인수 테스트에 활용 가능 실제 브라우저를 실행해서 테스트하므로 CSS, JavaScript에 대한 브라우저 호환성 테스트 가능 Eclipse 플러그인 제공 모바일 브라우저도 일부 자체 또는 써드파티로 지원
단점	<ul style="list-style-type: none"> Firefox, Chrome, IE, Safari, Opera 브라우저만 지원 실제 브라우저를 구동하므로 테스트가 무거움

4. 추천가이드

적용조건	추천도구
<ul style="list-style-type: none"> Java 외의 언어로 개발하는 경우 	
<ul style="list-style-type: none"> 브라우저 기반 테스트가 필요한 경우 	
<ul style="list-style-type: none"> 고객이 요구사항 전달/확인에 적극적인 경우 	
<ul style="list-style-type: none"> 브라우저 액션 녹화 기능이 필요한 경우 	
<ul style="list-style-type: none"> CSS나 JavaScript 관련 브라우저 호환성 테스트가 필요한 경우 	
<ul style="list-style-type: none"> 브라우저 호환성 테스트 필요없이 서버와의 웹 요청/응답에 대한 빠른 테스트가 필요한 경우 	

2. 프로그래밍 언어 활용(2001020215_15v3)

1-1

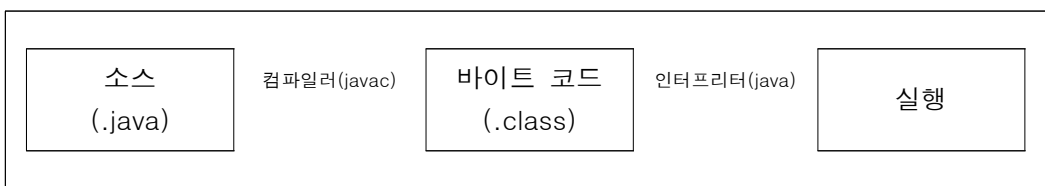
변수

응용소프트웨어 개발에 필요한 프로그래밍 언어의 데이터 타입을 적용하여 변수를 사용할 수 있다.

[1] 자바 언어의 개요

1) 자바 언어의 특징

- 단순하다
- 객체지향이다
- 플랫폼 독립적이다.
- 컴파일러와 인터프리터 언어의 두 가지 특징을 모두 갖는다.



- 분산 처리에 용이하다.
- 견고하다
- 안전하다.
- 멀티스레드(multi thread)를 지원한다.
- 동적이다.

2) 자바 개발 환경: 자바에는 보통의 프로그래밍 언어와 달리 하드웨어나 운영체제에 관계없이 실행될 수 있는 자체 플랫폼이 있는데, 이를 자바 개발 환경(JDK:Java Development Kit) 이라고 한다.

(1) JVM(Java Virtual Machine : 자바 가상 머신)

(2) Java API(Java Application Programming Interface)

3) 자바 프로그램의 종류

- 일반 프로그램: 자바 애플리케이션(application)
- 웹 클라이언트 프로그램: 애플릿(applet)
- 웹 서버 프로그램: 서블릿(servlet)과 JSP(Java Server Pages)
- 웹컴포넌트: EJB(Enterprise Java Beans)
- 모바일과 임베디드(embedded)

[2] 자바 언어의 기본 문법

1) 변수

(1) 변수 이름 작성 규칙

- 알파벳 대·소문자, 숫자, _, \$만 사용할 수 있다.
- 첫 문자로 숫자는 올 수 없다.
- 대·소문자를 구분한다.
- 자바 예약어는 사용할 수 없다.
- 길이 제한은 없다.

(2) 자바 예약어

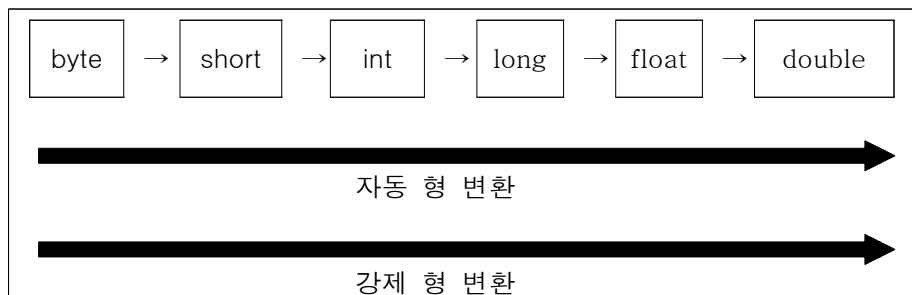
- 모두 소문자이며 대·소문자를 구분한다.
- `const`, `goto`는 현재 사용되지 않으나, 변수명으로는 사용할 수 없다.
- C/C++의 `sizeof`는 사용되지 않는다.

2) 데이터형

(1) 기본형

종류	데이터형	바이트 수
논리형	<code>boolean</code>	1바이트
단일 문자형	<code>char</code>	2바이트
정수형	<code>byte</code>	1바이트
	<code>short</code>	2바이트
	<code>int</code>	4바이트
	<code>long</code>	8바이트
실수형	<code>float</code>	4바이트
	<code>double</code>	8바이트

- (2) 데이터형 변환: 이미 선언된 데이터형을 다른 데이터형으로 변환하는 것을 데이터형 변환 (Data Type Casting)이라고 한다.



[3] 배열

자바는 같은 데이터형의 데이터를 여러 개 저장하기 위해 배열을 사용한다. 즉, 배열은 같은 종류의 데이터를 여러 개 저장하는 기억 장소이다. 배열을 사용하면 메모리도 절약하면서 쉽고 간결한 프로그램을 작성할 수 있다. 자바에서는 배열을 객체로 처리하며 참조형 변수이다.

- (1) 1단계 배열 선언: 배열객체가 저장된 기억 장소의 주소를 가리키는 변수를 선언한다. 배열 선언은 실제로 메모리 기억 공간이 확보되는 것이 아니기 때문에 배열의 크기를 지정할 수는 없다.

ex) `int a[];`

- (2) 2단계 배열 생성: 기억 공간을 할당하는 `new` 연산자로 배열객체가 저장될 기억 공간을 확보하고 이 기억 공간의 주소를 1단계의 배열 변수에 저장한다. 반드시 몇 개의 데이터를 저장할 것인지 크기를 지정해야 하며, 정해진 배열의 크기는 변경할 수 없다. 배열객체 각 원소의 기억 공간을 접근하는 인덱스는 0부터 시작하다.

ex) `a = int [3];`

- (3) 3단계 배열 초기화: 배열객체 각 원소의 기억 공간에 데이터를 초기화한다.

ex) `a[0] = 10; a[1] = 20; a[2] = 30;`

연산자

- 프로그래밍 언어의 연산자와 명령문을 사용하여 애플리케이션에 필요한 기능을 정의하고 사용할 수 있다.
- 프로그래밍 언어의 사용자 정의 자료형을 정의하고 애플리케이션에서 사용할 수 있다.

[1] 연산자

1. 산술 연산자

종류	기능	연산식	결과
+	덧셈	20+10	30
-	뺄셈	20-10	10
*	곱셈	20*10	200
/	나눗셈	20/10	2
%	나머지	20%10	0

2. 관계 연산자

종류	기능	연산식	결과
>	크다	20>10	true
>=	크거나 같다	20>=10	true
<	작다	20<10	false
<=	작거나 같다	20<=10	false
==	같다	20==10	false
!=	같지않다.	20!=10	true

3. 논리연산자

주어진 조건식이 참인지 거짓인지를 판단하여 true 또는 false의 논리형 결과를 반환한다. 논리 연산의 대상이 되는 피연산자 역시 논리형이어야 하며 연산 횟수를 축소시키는 단축 논리 연산자인 &&, ||도 제공한다.

종류	연산식	결과
&	(1>2) & (1<2)	false
&&	(1>2) && (1<2)	false
	(1<2) (1>2)	true
	(1<2) (1>2)	true
!	!(1<2)	false

4. 증감(증가/감소) 연산자

종류	연산식	결과
++a	a = 10; b = ++a + 10;	a = 11 b = 21
a++	a = 10; b = a++ + 10;	b = 20 a = 11
--a	a = 10; b = --a + 10;	a = 9 b = 19
a--	a = 10; b = a-- + 10;	b = 20 a = 9

5. 비트 연산자

종류	기능	연산식	결과
$a \& b$	a와 b의 비트 별로 두 비트가 모두 1이면 1, 하나라도 0이면 0	00000110 & 11111101	00000100
$a b$	a와 b의 비트 별로 두 비트가 모두 0이면 0, 하나라도 1이면 1	00000110 00001000	00001110
$a \wedge b$	a와 b의 비트 별로 두 비트가 서로 다르면 1, 같으면 0	00000110 ^ 00001111	00001001
$\sim a$	a와 b의 비트 별 반전	~ 00000110	11111001
$a \gg b$	a의 비트를 b 비트 수만큼 우측으로 이동	$00010100 \gg 2$	00000101
$a \ll b$	a의 비트를 b 비트 수만큼 좌측으로 이동	$00010100 \ll 2$	01010000
$a \ggg b$	a의 비트를 b 비트 수만큼 우측으로 이동	$00010100 \ggg 2$	00000101

6. 대입 연산자

종류	기능	연산식	결과
$a=b$	b 값을 a에 대입	$a = 2;$	$a = 2$
$a+=b$	a 값에 b를 더한 후, 다시 a에 대입 ($a = a + b$)	$a += 2;$	$a = 4$
$a-=b$	a 값에 b를 뺀 후, 다시 a에 대입 ($a = a - b$)	$a -= 2;$	$a = 0$
$a*=b$	a 값에 b를 곱한 후, 다시 a에 대입 ($a = a * b$)	$a *= 2;$	$a = 4$
$a/=b$	a 값에 b를 나눈 후, 다시 a에 대입 ($a = a / b$)	$a /= 2;$	$a = 1$
$a\%=b$	a 값을 b로 나눈 나머지를, 다시 a에 대입($a = a \% b$)	$a \% = 2;$	$a = 0$

7. 조건 연산자(삼항 연산자)

종류	기능	연산식	결과
$?:$	조건이 참이면 처리1을 수행하고, 거짓이면 처리2를	$a = 2;$ $b = 3;$ $c = (a > b) ? a : b;$	$c = 2$

8. 연산자 우선순위

	우선순위	연산자
1	함수 괄호 배열	() []
2	전치	++ -- ~ !
3	승제	* / %
4	가감	+ -
5	비트 이동	<< >> >>>
6	비교	< <= > >=
7	비교(등식)	== !=
8	비트 AND	&
9	비트 XOR	^
10	비트 OR	
11	논리 AND	&&
12	논리 OR	
13	조건	?:
14	대입	= 및 축약 대입 연산자

[2] 제어문

1. 선택문

자바의 조건문에는 if, if-else, if-else if-else, switch문이 있다.

2. 반복문

자바의 반복문에는 for, while, do-while이 있다.

3. 분기문

자바의 분기문에는 break, continue등이 있다.

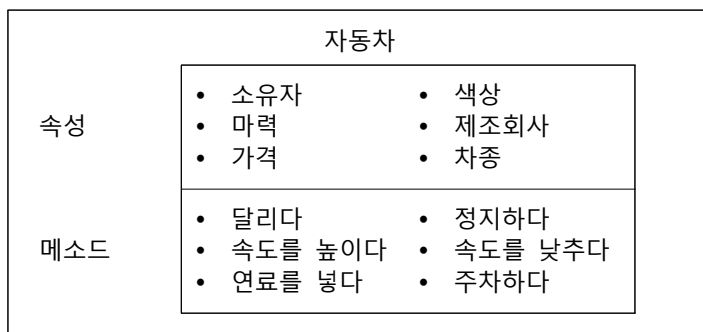
[3] 객체지향의 개념

1. 객체지향 언어는 시물라에서 시작하여 스몰토크(smalltalk), 이펠(efel)등 순수 객체지향 언어로 발전해왔다. 그러나 이러한 순수 객체지향 언어는 범용적인 언어로 널리 활용되지 못하였고, C언어에 객체지향 개념을 도입한 C++이 객체지향 언어로는 널리 이용되고 있으며 C++를 기반으로 해서 탄생하게 된 객체지향 언어가 바로 자바이다.
2. 객체지향 언어와 대비되는 언어로서 기존의 프로그램 개발 방법인 절차지향(procedural-oriented) 언어가 있다. 절차지향 언어가 공유된 데이터와 문제 해결 절차를 중시하는 반면, 객체지향 언어는 프로그램을 객체들의 집합으로 보며 이러한 객체들이 프로그램의 중시이 된다.

[4] 객체와 클래스

- 객체는 실세계에 존재하는 모든 것이 될 수 있으며 명사로서의 성질을 갖춘 모든 것은 객체로 만들어질 수 있다.
- 객체의 정적인 특성을 표현하는 속성(attribute)
- 객체의 동적인 특성인 행위를 표현하는 메소드(method)

1) 1단계 : 실세계 객체 모델링



2) 2단계 : 클래스 정의

```
class 자동차{
    String 소유자;
    String 색상;
    String 제조회사;
    void 달리다(){ }
    void 정지하다(){ }
}
```

3) 3단계 : 객체의 생성 및 사용

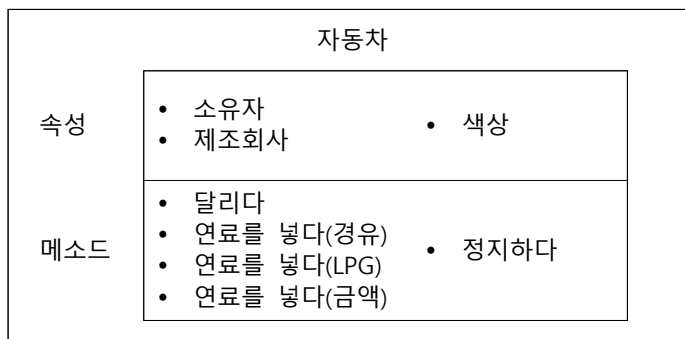

```

자동차 김현우 = new 자동차();
김현우.소유자 = "김현우";
김현우.색상 = "빨간색";
김현우.제조회사 = "현대";
김현우.달리다();
김현우.정지하다();

```

[5] 객체지향 언어의 특징

- 1) 캡슐화(encapsulation): 클래스는 캡슐과 같은 형태로 만들어진다. 객체는 속성과 속성을 처리하는 메소드가 캡슐화되어 있어서 객체를 사용하는 쪽에서는 캡슐화된 속성과 메소드가 실제 어떻게 처리되는지는 알 필요없이 단지 객체를 사용할 수 있는 인터페이스를 통해 사용만 하면 된다.
- 2) 상속(inheritance): 기존의 클래스에서 모든 속성과 메소드를 상속받고 더 필요한 속성과 메소드를 추가하여 새로운 클래스를 생성하는 것을 상속이라고 하다.
- 3) 다형성(polymorphism)
 - (1) 메소드 오버로딩(overloading) : 하나의 클래스에서 동일한 이름의 메소드가 매개 변수의 개수나 타입을 달리해서 여러 가지 형태로 존재하는 것



- (2) 메소드 오버라이딩(overriding) : 상속 관계에 있는 하위 클래스가 상위 클래스의 메소드를 재정의하는 것
- 4) 메시지(message) : 객체 간의 서로 통신하는 방법

[6] 클래스

1. 클래스의 정의

클래스는 선언부와 멤버로 나뉜다. 클래스 멤버에는 객체 생성 및 초기화를 담당하는 생성자, 속성을 정의하는 멤버 변수, 데이터를 조작하는 메소드가 포함된다. 경우에 따라서는 클래스 안에 또 다른 클래스가 선언되기도 하는데 이를 내부 클래스 라고 한다.

2. 객체 생성

- 1단계 : 객체 참조형 변수 선언
Student kim;
- 2단계 : 객체 생성
kim = new Student()
- 1,2단계를 하나의 문장으로 표현할 수 있다.
Student kim = new Student();

3. 멤버 변수

(1) 접근권한

접근제어자	사용 범위				설명
	클래스	하위 클래스	동일 패키지	모든 클래스	
private	○	×	×	×	<ul style="list-style-type: none"> 자신 클래스 안에서만 사용할 수 있는 멤버 선언
(default)	○	×	○	×	<ul style="list-style-type: none"> 접근 권한 modifier를 지정하지 않은 경우 동일 패키지의 클래스에서만 접근 가능
protected	○	○	○	×	<ul style="list-style-type: none"> 동일 패키지의 클래스 또는 동일 패키지는 아니지만 하위 클래스 관계인 클래스에서만 접근 가능
public	○	○	○	○	<ul style="list-style-type: none"> 모든 클래스에서 접근 가능

(2) 클래스 변수(static)

멤버 변수에는 변수 이름 앞에 static이 붙은 클래스 변수와 그렇지 않은 객체 변수가 있다. 객체 변수는 객체가 생성될 때마다 기억 공간이 각각 확보되지만, 클래스 변수는 클래스가 로딩되는 과정에서 기억 공간이 한 번만 확보되고 해당 클래스 타입의 모든 객체들은 클래스를 공유하게 된다. 그러므로, 클래스 변수는 하나의 클래스로부터 생성된 객체들 사이의 공통된 속성을 표현하는 데 주로 사용된다.

변수	생성시기	생성위치	생성개수
클래스변수	클래스 로딩 시 (객체 생성 이전)	클래스	1개
객체 변수	객체 생성 시	각 객체	객체 생성 수

(3) 상수(final)

멤버 변수에 final이 붙으면 해당 멤버 변수는 더 이상 값을 변경할 수 없다는 의미이다. 즉, final이 붙은 변수는 단 한번 초기화할 수 있으며 그 뒤도를 값을 변경할 수 없는 상수이다. 상수의 이름은 대문자를 사용한다.

4. 생성자(constructor)

(1) 생성자 정의

- ① new 연산자와 함께 사용되며 객체 생성 시에 호출된다.
- ② 객체 생성 시에 객체의 멤버 변수(객체 변수)를 초기화하는 일을 주로 한다.
- ③ 생성자의 이름은 클래스 이름과 동일하며, 일반적으로 첫 문자는 대문자이다.
- ④ return 유형이 없다.
- ⑤ 생성자가 없는 클래스는 자바 가상 머신(JVM)이 자동으로 삽입하는 기본 생성자를 갖는다.
- ⑥ 사용자가 정의한 생성자가 있을 경우에는 자바 가상 머신(JVM)이 삽입한 기본 생성자는 자동으로 없어진다.

(2) 생성자와 객체

생성자는 객체 생성 시 반드시 호출되어야 하는 메소드이다. 그러나, 생성자가 없는 클래스의 경우에는 기본 생성자를 갖기 때문에 클래스가 반드시 생성자를 가지고 있어야 하는 것은 아니다.

(3) 생성자 오버로딩

클래스는 매개 변수의 개수, 타입, 순서를 달리하는 같은 이름의 생성자를 여러 개 정의할

수 있으며 이를 생성자 오버로딩(overloading)이라고 한다. 이러한 다양한 형태의 생성자를 통해 다양한 형태의 객체를 생성함으로써 다형성이 구현될 수 있다.

(4) this

- ① this: 객체 자신을 의미하며 this.속성, this.메소드 형태로 사용된다. 보통 생성자나 메소드의 매개 변수 이름이 멤버 변수 이름과 같을 경우에 많이 사용된다.
- ② this(): 객체의 생성자를 의미한다. 보통 생성자 오버로딩에서 생성자 안에서 다른 생성자를 호출하는 경우에 많이 사용된다.

2

프로그래밍 언어별 특성

- 프로그래밍 언어별 특성을 파악하고 설명할 수 있다.
- 파악된 프로그래밍 언어의 특성을 적용하여 애플리케이션을 구현할 수 있다.
- 애플리케이션을 최적화하기 위해 프로그래밍 언어의 특성을 활용할 수 있다.

[1] 저급 언어의 종류와 특징

1. 기계어

2진수 코드로 CPU 종류마다 고유의 기계어가 존재

명령어	데이터	
1001	0001	0001 위치값을 누산기에 저장하라.
1100	0010	누산기에 0010 위치값을 더하라
1010	0011	누산기값을 0011 위치에 저장하라

2. 어셈블리어

기계어의 2진수 코드를 기호화 코드(Mnemonics)로 대체한 것

명령어	데이터	
LOAD	Y	Y값을 누산기에 저장하라.
ADD	Z	누산기에 Z값을 더하라.
STORE	X	누산기 값을 X에 저장하라.

[2] 초기 고급 언어의 종류

1. FORTRAN(FORmula TRANslation)

- 1957년에 IBM의 John Backus가 개발한 최초의 고급 언어
- 과학 계산용 언어로, 프로그램을 단순화된 영어 단어와 수학 공식으로 표현

2. COBOL(COMmon Business Oriented Language)

- 1960년초에 개발된 사무용 프로그램을 개발하기 위한 언어
- COBOL로 작성된 많은 급여, 회계 및 기타 업무용 소프트웨어는 지금도 많이 사용됨.

3. BASIC(Beginner's All-purpose Symbolic Instruction Code)

- 초보자가 배우기 쉬운 프로그래밍 언어로 개발
- 원래 학생들에게 프로그래밍 언어를 가르치기 위해 개발
- 컴파일 과정 없이 작성한 소스를 바로 실행할 수 있다.

[3] 현재 가장 많이 사용되고 있는 고급 언어의 종류

1. C언어

- AT&T 벨 연구소의 Dennis Ritchie가 개발한 시스템 프로그래밍용 언어
- 1970년대 'UNIX'라는 운영체제를 개발하기 위하여 사용된 언어
- 고급 언어이면서도 저급 언어(하드웨어에 가까운 언어)의 특성을 가지고 있음

2. C++

- AT&T 벨 연구소의 B.Stroustrup에 의해 개발
- C언어의 기능을 확장하여 만든 객체 지향 프로그래밍 언어

3. Java

- 1990년대에 선 마이크로시스템 사의 James Gosling이 개발
- 인터넷 환경을 위해 개발된 객체 지향 언어로, 운영체제에 독립적 코드 실행을 지원

3

애플리케이션 라이브러리

- 애플리케이션에 필요한 라이브러리를 검색하고 선택할 수 있다.
- 애플리케이션 구현을 위해 선택한 라이브러리를 프로그래밍 언어 특성에 맞게 구성할 수 있다.
- 선택한 라이브러리를 사용하여 애플리케이션 구현에 적용할 수 있다.

[1] 패키지의 개념

패키지(package)는 자바 개발 환경에서 제공되는 거대한 클래스 라이브러리로 관련된 클래스와 인터페이스의 모임을 말한다. 자바에서 하나의 패키지는 하나의 디렉토리와 대응되며, 각 디렉토리에는 관련된 클래스와 인터페이스가 저장되어 있다. 자바 패키지의 구조를 보기 위해 자바에서는 자바 API(Application Programming Interface)를 제공한다.

[2] 패키지의 종류

1. java.lang

자바 언어의 기초적인 사항을 정의한 클래스와 관련된 패키지로 Object 클래스, 문자열 관련 클래스, 시스템 관련 클래스, 멀티 스레드 클래스 등과 같이 자바 프로그램 작성에 필요한 기본적인 기능을 제공한다.

2. java.io

자바 입·출력 기능과 관련된 패키지로 파일이나 버퍼 등의 입·출력 관련 기능을 제공한다.

3. java.awt

GUI 작성과 관련된 패키지로 javax.swing 패키지와 함께 자바 GUI 어플리케이션 작성 시 필수적으로 사용되는 버튼, 텍스트 필드, 메뉴 등 관련 컴포넌트와 이벤트 기능을 제공한다.

4. java.applet

애플릿 작성에 필요한 기능을 모아 놓은 패키지이다.

5. java.net

자바의 네트워크에 관련된 패키지로 소켓과 관련된 기능을 제공한다.

6. javax.swing

java.awt 패키지와 더불어 자바 GUI 애플리케이션 작성과 관련된 패키지로 java.awt보다 다양하고 융통성 있는 컴포넌트를 제공한다.

7. java.util

유틸리티성 기능과 관련된 패키지로 날짜 표현이나 여러 자료형을 하나로 취급하는 컬렉션과 관련된 기능을 제공한다.

8. java.sql

데이터베이스 기능과 관련된 패키지로 데이터베이스 연결과 그 연결을 관리하며, 데이터베이스 명령 처리에 필요한 기능을 제공한다.

[3] 패키지의 이용

자바가 이용하는 클래스를 이용하려면 그 클래스의 패키지를 프로그램에서 명시해야 한다.

```
import 패키지명.*;  
import 패키지명.클래스명;
```

import 문은 프로그램의 맨 앞에 기술해야 한다. 만약 String 클래스를 사용하기 위해서는 import java.lang.* 또는 import java.lang.String과 같이 기술해야 한다.

3. 응용 SW 기초 기술 활용(2001020216_15v3)

1-1

운영체제

응용 소프트웨어를 개발하기 위하여 다양한 운영체제의 특징을 설명할 수 있다.

[1] 운영체제 정의

운영체제(OS; Operating System)는 컴퓨터 하드웨어와 일반 컴퓨터 사용자 또는 컴퓨터에서 실행되는 응용 프로그램의 중간에 위치하여 사용자들이 보다 쉽고 간편하게 컴퓨터 시스템을 이용할 수 있도록 제어하는 시스템 소프트웨어의 일종

[2] 운영체제의 목적

처리능력(Throughput)	일정 시간 내에 시스템이 처리하는 일의 양
반환 시간(Turn Around Time)	시스템에 작업을 제시하고 나서부터 결과를 받을 때까지의 경과시간
사용 가능성(Availability)	시스템을 사용할 필요가 있을 때 즉시 사용 가능한 정도
신뢰도(Reliability)	시스템이 주어진 문제를 정확하게 해결하는 정도

[3] 운영체제의 기능

- 사용자와 컴퓨터 간의 인터페이스 제공
- 시스템의 효율적인 운영 및 관리
- 자원 스케줄링 및 주변장치 관리
- 사용자 간의 데이터 호환
- 각종 하드웨어와 네트워크 관리
- 데이터 공유 및 관리
- 초기 설정 기능 및 이식성 기능
- 시스템의 오류 검사 및 복구

[4] 운영체제의 운용방식

1. 일괄처리시스템
2. 다중 프로그래밍 시스템
3. 시분할 시스템
4. 다중처리시스템
5. 실시간 처리 시스템
6. 분산 처리 시스템

[5] 운영체제의 종류

1. DOS(Disk Operating System)

- 개인용 컴퓨터에서 디스크와 파일 관리를 위한 운영체제이다.
- 단일 처리 방식이며, CUI(Character user Interface) 환경의 대표적인 운영체제이다.
- 디렉터리 : 파일의 저장, 보관, 관리 등 효율적이 파일 사용을 위해 디스크에 존재하는 파일에 대한 여러 정보를 갖고 있는 특수한 형태의 파일

2. Window

- 그래픽 사용자 인터페이스(GUI)사용
- 선점형 멀티태스킹(Preemptive Multi-Tasking)
- 32비트 또는 64비트 데이터 처리
- FAT32 파일 시스템
- 플러그 앤 플레이(PnP; Plug & Play)
- OLE(Object Linking and Embedding)
- 255자의 긴 파일 이름
- 향상된 네트워크 기능
- DOS와 호환

3. Unix

- 시분할 시스템(Time Sharing System)을 위해 설계된 대화식 운영체제
- 소스가 공개된 개방형 시스템
- 대부분 C언어로 작성되어 있어 이식성이 높으며 장치, 프로세스 간의 호환성이 높다.
- 크기가 작고 이해하기가 쉽다.
- Multi-User, Multi-Tasking을 지원한다.
- 많은 네트워킹 기능을 제공하므로 통신망 관리용 운영체제로 적합
- 계층적 트리 구조의 파일 시스템
- 표준 입출력을 통해 명령어들이 파이프라인으로 연결된다.



커널(Kernel)	<ul style="list-style-type: none"> • UNIX의 가장 핵심적인 부분 • 컴퓨터가 부팅될 때 주기억장치에 적재된 후 상주하면서 실행 • 프로그램과 하드웨어 간의 인터페이스 역할 • 프로세스(CPU스케줄링)관리, 기억장치 관리, 파일관리, 입출력관리, 데이터 전송 및 변환 등 기능 수행
셸(Shell)	사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 명령어 해석기
유틸리티 프로그램	일반 사용자가 작성한 응용프로그램을 처리하는데 사용

1-2

운영체제 기본 명령어

CLI(Command Line Interface) 및 GUI(Graphic User Interface) 환경에서 운영체제의 기본명령어를 활용할 수 있다.

[1] Windows 운영체제 기본 명령어

명령어	기능
msconfig	시작프로그램 및 서비스를 중지 시킨다거나 하는 등의 시스템 리소스 확보를 위한 윈도우 기초적인 환경설정 가능
netstat -na	현재 내 컴퓨터에 접속되어 있는 IP를 출력
dir	디렉토리에 있는 파일과 하위 디렉토리 목록을 보여줌.
ping	상대방 IP로 접속이 가능 여부 출력 명령
xcopy	복사 명령어 (하드 OS까지 복사 가능함)
ipconfig	명령프롬프트에서 실행 시키면 내컴퓨터의 IP 확인
cmd	명령 프롬프트를 실행시키는 명령
regedit	레지스트리 편집기 실행
shutdown -a	컴퓨터의 로컬 및 원격 종료를 허용함.

[2] 리눅스/유닉스 계열 운영체제 기본 명령어

명령어	기능	명령어	기능
login	사용자 인증	passwd	패스워드 변경
ls	파일이나 디렉토리를 보는 명령어	cd	작업 디렉터리 변경
cp	파일 복사	mv	파일이름 변경, 파일 위치 변경
mkdir	디렉터리 만들기	rm	디렉토리 삭제
pwd	현재의 디렉토리 경로 보여주기	rmdir	디렉터리 제거
cat	파일 연결 및 출력	grep	패턴에 따른 문자열 검색
chgrp	파일 그룹 소유권 변경	echo	인수를 표준 출력에 기록
sort	텍스트 파일의 정렬, 병합, 시퀀스 검사	find	파일 찾기
chmod	파일 모드/특성/권한 변경 예) -rwxr-xr-x guestbook.html rwx :처음 3개 문자 = 사용자 자신의 사용 권한 r-x :그다음 3개 문자 = 그룹 사용자의 사용 권한 r-x :마지막 3개 문자 = 전체 사용자의 사용 권한		읽기(read) : 화일 읽기 권한 쓰기(write) : 화일 쓰기 권한 실행(execution) : 화일 실행 권한 없음(-) : 사용권한 없음

[1] 메모리 관리 기법

1. CMA : 연속 메모리 할당

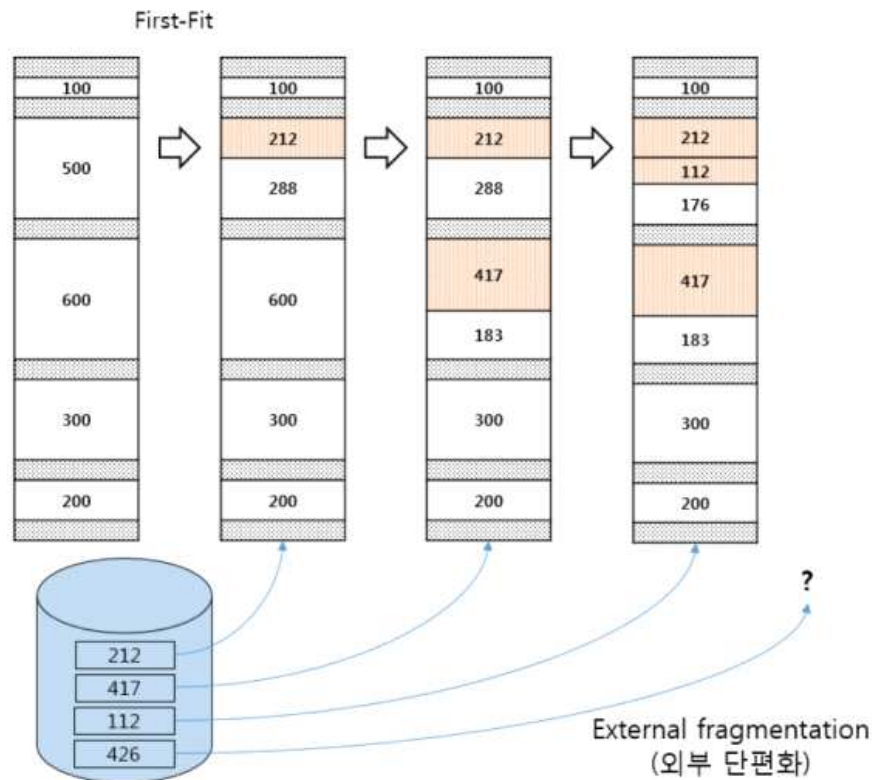
(1) 하나의 프로세스는 연속된 메모리 공간에 할당을 받게 된다.

(2) 외부 단편화 발생

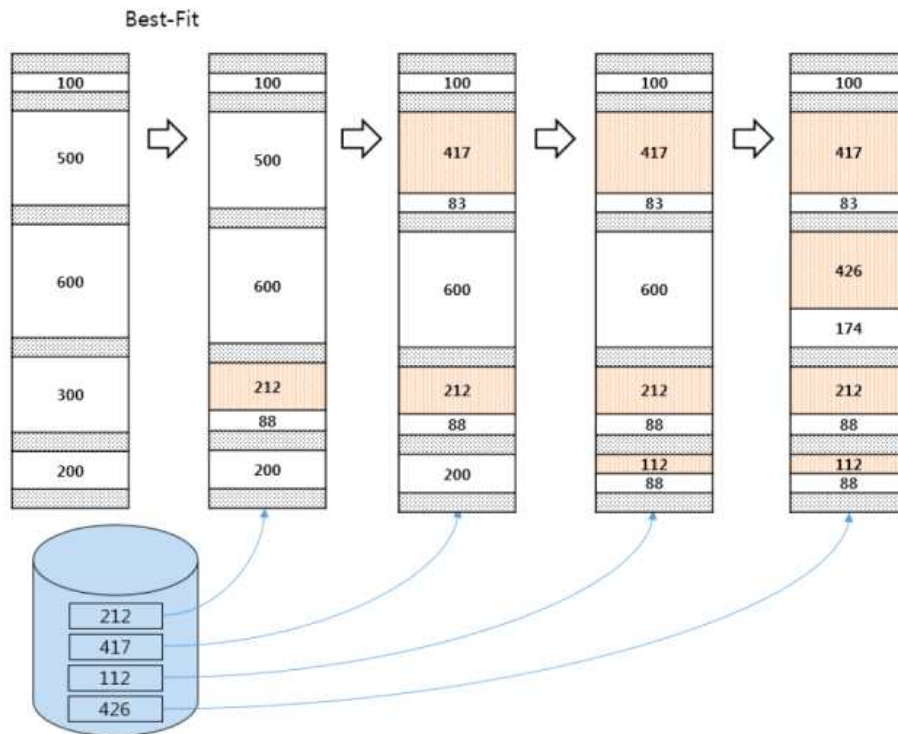
빈 메모리 공간이 있지만 이 공간이 여러 개의 불연속적인 작은 공간으로 나뉘어져 있어서 프로세스를 할당할 수 없는 상태일 때 이 조각난 공간들에 대한 현상을 외부 단편화라고 한다. 외부 단편화는 심한 메모리 낭비의 원인이 된다.

(3) 메모리 할당 방식

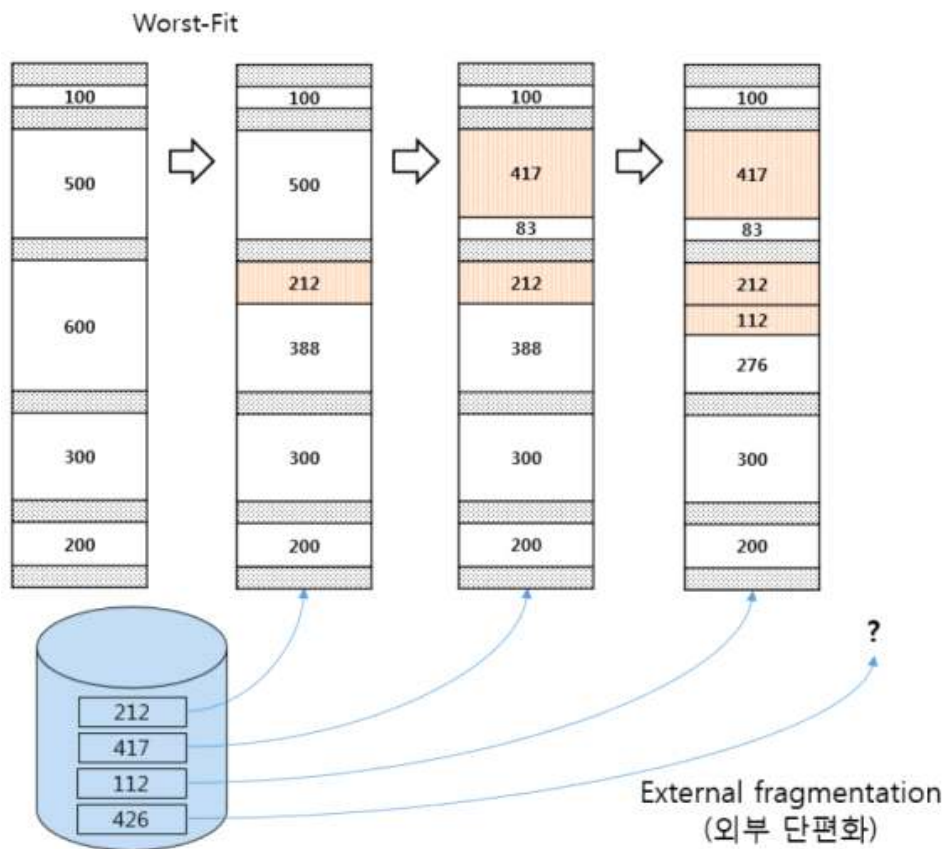
① First-fit(최초 적합) : 최초로 할당받을 수 있는 크기의 공간에 무조건 할당됨. 이용률 좋음



② Best-fit(최적 적합) : 크기가 가장 근접한 공간에 할당됨. 속도 가장 빠름



③ Worst-fit(최악 적합) : 크기가 가장 많이 차이가 나는 공간에 할당됨.



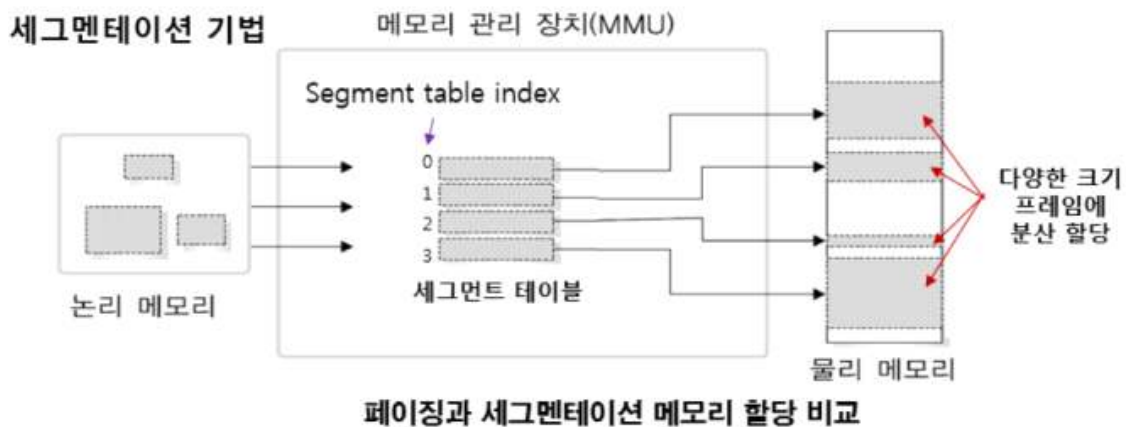
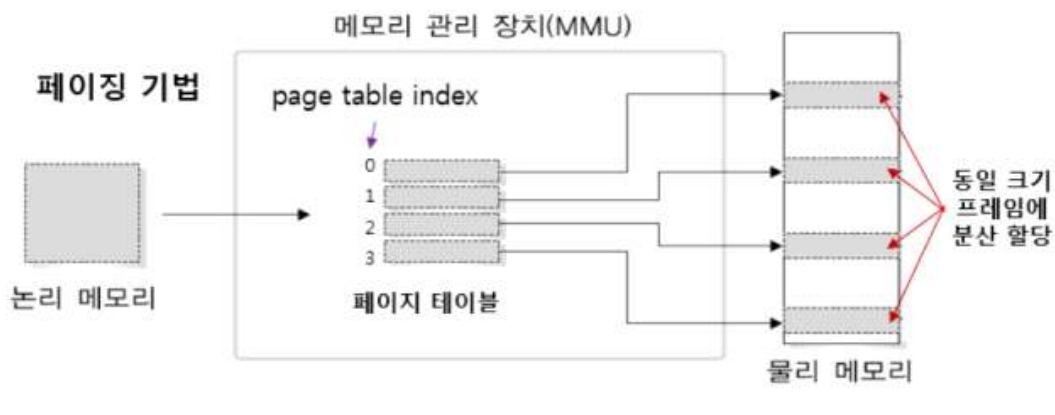
(4) 압축(Compaction) : 외부 단편화의 해결방안으로 흩어져 있는 홀들을 하나의 커다란 홀로 통합하여 사용한다.

2. 페이징(Paging)

- (1) 프로세스를 일정한 크기로 잘라서 메모리에 할당함.
 - 일정한 크기의 프로세스 : 페이지(page)
 - 일정한 크기의 메모리 공간 : 프레임(frame)
- (2) 물리 메모리 내에 하나의 프로세스가 연속된 메모리를 할당받을 필요 없이 여러 개의 페이지 조각으로 나누어져서 분산 할당된다.
- (3) 장점 : 프로세스와 메모리가 동일한 크기로 규격화함으로써 외부 단편화가 발생하지 않는다.
공유 페이지의 이용 가능
압축 불필요, 메모리 활용을 통한 다중처리 프로그래밍 실현
- (4) 단점 : 내부 단편화 발생(미미한 수준), 페이징 매핑을 위한 하드웨어로 속도 저하

3. 세그멘테이션(Segmentation)

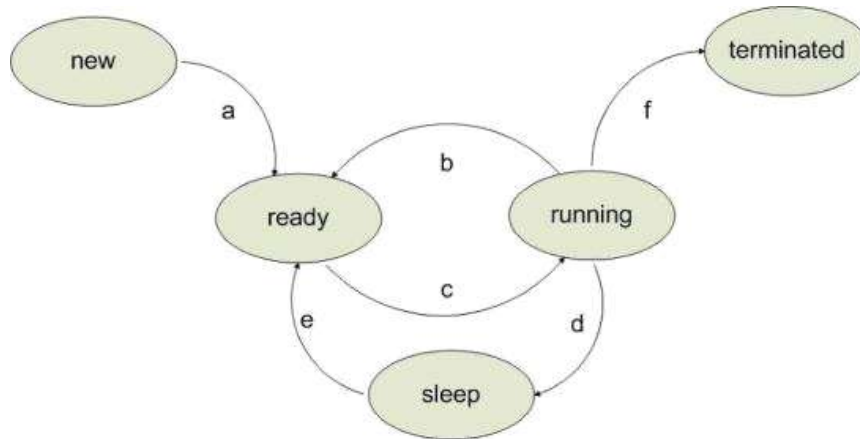
- (1) 프로세스를 논리적 내용(세그먼트)으로 잘라서 메모리에 배치
 - 프로세스는 세그먼트의 집합
- (2) 동일한 크기를 갖는 페이징과는 달리 메모리를 프로그램을 구성하는 메인루틴, 서브루틴, 프로시저, 함수 또는 모듈, 전역 변수, 스택 등의 각각 다른 크기를 갖는 세그먼트로 나눔
- (3) 각 세그먼트는 연관된 기능을 수행하는 하나의 모듈 프로그램이므로 논리적 구조화가 쉬움.
- (4) 각 세그먼트가 페이징처럼 메모리에서 서로 인접할 필요 없음
- (5) 장점 : 메모리 보호, 메모리 공유
- (6) 단점 : 외부 단편화 발생



[2] 프로세스 스케줄링 기법의 이해

1. 프로세스 생명 주기

- (1) 생성(new) : 디스크에서 메모리로 프로그램이 올라가 실행준비를 하는 상태
- (2) 준비(ready) : 순서에 맞게 처리를 기다리는 상태
- (3) 수행(running) : 작업이 처리되고 있는 상태
- (4) 대기(waiting) : 프로세스가 잠시 어떠한 조건에 의해 작업이 이루어질 수 없는 상태로 그 조건이 해결되기를 기다리는 상태
- (5) 종료(terminated)



- a. 생성과정에서 준비상태로의 이동과정에서는 프로세스가 처리되기 위한 메모리와 필요한 주변장치들이 체크된다.
- b. 준비상태에서 수행상태로의 이동과정에서는 미리 정해진 알고리즘에 따라 프로세스가 선택 처리된다.
- c. 수행상태에서 다시 준비상태로 가는때는 현재 처리되고 있는 프로세스의 타임 쿼텀이 다 되었을 경우이다.
- d. 수행상태에서 sleep상태로의 이동은 입력/출력 요청에 의해 이뤄진다. 하나의 프로세스가 수행상태에서 입력/출력이 요청되면 그 작업이 마쳐질 때 까지 그 프로세스는 sleep 상태로 잠시 이동되고 커널의 스케줄링 알고리즘에 따라 적절한 순위의 프로세스가 수행 상태로 이동되는 것이다.
- e. sleep 상태에서 준비상태로의 이동은 입력/출력이 끝났다는 신호에 의해 이루어진다.
- f. 수행상태에서 종료상태로의 이동은 작업이 성공적으로 완료되었거나 혹은 에러가 발생하여 작업이 미리 종료되는 경우이다.

2. 선점형 스케줄링

(1) 특징

- 어떤 프로세스가 CPU를 할당받아 실행 중에 있어도 다른 프로세스가 실행 중인 프로세스를 중지하고 CPU를 강제로 점유할 수 있다.
- 모든 프로세스에게 CPU 사용 시간을 동일하게 부여할 수 있다.
- 빠른 응답시간을 요하는 대화식 시분할 시스템에 적합하며 긴급한 프로세서를 제어할 수 있다.

(2) 종류

- ① RR 스케줄링

- 시분할 시스템에서 사용됨
- FCFS와 비슷하지만 제한시간이 지난 후 다음 프로세스에게 자원을 할당해줌
- 할당되는 시간이 클 경우 FCFS와 비슷해짐
- ② SRT 스케줄링 : 최소 잔류 시간 우선 스케줄링 (shortest remaining time)
 - 현재 작업 중인 프로세스를 중단시키고 새로 들어온 프로세스의 처리를 시작하는 방식
- ③ 다단계 큐 스케줄링
 - 프로세스의 우선순위에 따라 시스템 프로세스, 대화형 프로세스, 일괄처리 프로세스 등으로 나누어 준비상태 큐를 상위 중위 하위 단계로 배치
- ④ 다단계 피드백 큐 스케줄링
 - 다단계 큐의 단점을 보완함
 - 큐마다 time out을 설정하여 time out 초과시 우선순위가 낮은 다음 단계 큐로 이동함

3. 비선점형 스케줄링

(1) 특징

- 어떤 프로세스가 CPU를 할당 받으면 그 프로세스가 종료되거나 입출력 요구가 발생하여 자발적으로 중지될 때까지 계속 실행되도록 보장한다.
- 순서대로 처리되는 공정성이 있고 다음에 처리해야 할 프로세스와 관계없이 응답 시간을 예상할 수 있다.
- 선점 방식보다 스케줄러 호출 빈도가 낮고 문맥 교환에 의한 오버헤드가 적다.
- 일괄 처리 시스템에 적합하며, CPU 사용 시간이 긴 하나의 프로세스가 CPU 사용 시간이 짧은 여러 프로세스를 오랫동안 대기시킬 수 있으므로, 처리율이 떨어질 수 있다는 단점이 있다.

(2) 종류

- ① FCFS 스케줄링 : 먼저 들어온 프로세스가 먼저 처리되는 FIFO 구조의 알고리즘
- ② SJF 스케줄링 : 단기작업 우선 스케줄링
 - 준비상태큐에 있는 프로세스들 중에 실행시간이 가장 작은 프로세스에게 자원을 할당하는 방법
 - 가장 적은 평균대기시간을 가짐
- ③ HRN : SJF의 단점을 보완하기 위해서 등장
 - 우선순위 계산식을 이용하여 자원할당N 스케줄링

4. 스케줄링 알고리즘 평가 기준

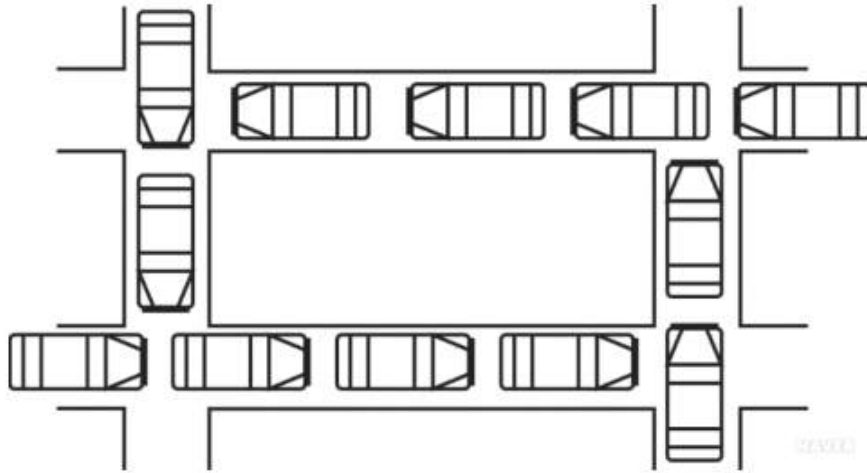
- (1) CPU 사용률(CPU Utilization) : 전체 시스템 시간 중 CPU가 작업을 처리하는 시간의 비율.
- (2) 처리량(Throughput) : CPU가 단위 시간당 처리하는 프로세스의 개수.
- (3) 응답 시간(Response Time) : 대화식 시스템에서 요청 후 응답이 오기 시작할 때까지의 시간.
- (4) 대기 시간(Waiting Time) : 프로세스가 준비 큐 내에서 대기하는 시간의 총합.
- (5) 반환 시간(Turnaround Time) : 프로세스가 시작해서 끝날 때까지 걸리는 시간.

[3] 교착상태

1. 정의

- 1) 교착상태는(Dead Lock)는 다중 프로그래밍 상에서 두 개의 프로세스가 실행중에 있을 때 각 프로세스는 자신이 필요한 자원을 가지고 실행하다가 서로 자신이 점유하고 있는 자원을 포기하지 않은 상태에서 다른 프로세스가 자원을 요구하여 두 프로세스 모두 실행을 할 수 없게 되는 현상을 의미한다.

- 2) 아래 그림과 같이 자동차(프로세스)들이 현재 위치한 길(자원)을 점유함과 동시에 다른 차가 사용하는 길을 사용하려고 대기하고 있지만 다른 길을 사용할 수 없으며 현재 길에서도 벗어나지 못하는 상태이다.



2. 교착 상태 발생의 필요 충분 조건

교착 상태가 발생하기 위해서는 다음의 네 가지 조건이 충족되어야 하며, 이 네 가지 조건 중 하나라도 충족되지 않으면 교착 상태가 발생하지 않는다.

상호 배제 (Mutual Exclusion)	한 번에 한 개의 프로세스만이 공유 자원을 사용할 수 있어야 함
점유와 대기 (Hold and Wait)	최소한 하나의 자원을 점유하고 있으면서 다른 프로세스에 할당되어 사용되고 있는 자원을 추가로 점유하기 위해 대기하는 프로세스가 있어야 함
비선점 (Non-preemption)	다른 프로세스에 할당된 자원은 사용이 끝날때까지 강제로 빼앗을 수 없어야 함.
환형 대기, 순환 대기 (Circular Wait)	공유 자원과 공유 자원을 사용하기 위해 대기하는 프로세스들이 원형으로 구성되어 있어 자신에게 할당된 자원을 점유하면서 앞이나 뒤에 있는 프로세스의 자원을 요구해야 함

2-1

데이터베이스 종류와 선정

데이터베이스의 종류를 구분하고 응용 소프트웨어 개발에 필요한 데이터베이스를 선정할 수 있다.

[1] DBMS 각 유형별 특징

1. DBMS 개념

데이터베이스를 관리하며 응용 프로그램들이 데이터베이스를 공유하며 사용할 수 있는 환경을 제공하는 소프트웨어

2. DBMS 분류

1) File 시스템 : 파일 중심의 전통적인 데이터 처리 시스템.

각 응용프로그램이 개별적으로 자신의 데이터를 파일로 관리

2) 계층형 DBMS : 데이터의 관계를 트리 구조로 정의하고 부모, 자식, 형태를 갖는 구조

- 장점: 데이터의 액세스 속도가 빠르고, 데이터의 사용량을 쉽게 예측 할 수 있다.

- 단점: 상하 종속적인 관계로 구성되어 있어 초기 셋팅 후 변화하는 프로세스를 수용하기가 쉽지 않다.

- 3) Network DBMS : 레코드 간의 다양한 관계를 그물처럼 갖는 구조
 - 장점 : 계층형 데이터베이스 관리 시스템의 문제점인 상하 종속적인 관계는 해결되었다.
 - 단점 : 구성과 설계가 복잡하고 궁극적으로 데이터의 종속성을 해결하지 못하였다.
- 4) 관계형 DBMS : 행, 열로 구성된 테이블 간의 관계를 나타낼 때 사용
 - 장점 : 업무 변화에 대한 적응력 높아 변화하는 업무에 쉽게 활용하며 유지보수 편리하다. 따라서 생산성도 향상된다.
 - 단점 : 다른 DBMS 보다 더 많은 자원이 필요로 해 시스템의 부하가 높다.
- 5) 객체지향 DBMS : 계층에 따라 데이터 구조를 표현하고 데이터와 그 조작 절차를 함께 다룸
- 6) 객체관계형 DBMS : 데이터 형과 메서드를 자유롭게 정의하여 데이터베이스를 개발할 수 있음

3. DBMS 종류

1) Oracle

오라클사의 Oracle DBMS는 데이터베이스 관리 시스템 중에서 가장 인기 있는 제품중 하나다. Oracle DBMS 윈도우, 리눅스, 유닉스, 등의 여러 운영체제에서의 동작을 지원하며 여러 유틸리티를 제공하고 다양한 언어등과 연계되어 동작하는 융통성을 보인다.

- 오라클에서 만들어 판매중인 상업용 데이터베이스로 윈도우즈 리눅스 유닉스 등 다양한 운영체제에 설치를 할 수 있다.
- MS_SQL , MY_SQL보다 대량의 데이터를 처리하기 좋음
- 대기업에서 주로 사용하며 글로벌 DB시장 점유율 1위
- 비공개 소스, 폐쇄적인 운영

2) MY_SQL

선 마이크로시스템즈사의 MySQL은 오픈소스 제품으로 오픈소스의 특징을 살려 처음엔 주로 개인 사용용으로 사용 되었다. 그러나 기능이 점차적으로 향상되고 상대적으로 다른 DBMS 제품보다 저렴한 가격적 요인으로 인해 여러 소규모 조직에서 그 영역 을 확대하고 있다

- MySQL사에서 개발, 썬마이크로시스템즈를 거쳐 현재 오라클에 흡수 합병됨
- 윈도우즈 리눅스 유닉스 등 다양한 운영체제에 설치를 할 수 있다.
- 오픈소스로 이루어져있는 무료 프로그램(상업적 사용 시 비용 있음)
- 가격 등의 장점을 앞세워 다수의 중소기업에서 사용 중이다.

3) MS_SQL

마이크로소프트사의 윈도우 운영체제를 기반으로 동작하는 SQL Server는 여러 종류의 에디션을 제공하여 차별화된 가격과 기능을 선보임으로써 기업의 인원이나 특성을 고려한 다양한 조직의 요구를 만족시키고 있다.

- 마이크로소프트사에서 개발한 상업용 데이터베이스
- 다른 운영체제도 사용가능하지만 윈도우즈에 특히 특화되어있음
- 비공개 소스로 폐쇄적인 정책 (리눅스버전은 오픈소스)
- 비교적 중소기업에서 주로 사용함

4) DB2

IBM사의 DB2는 오라클과 경쟁구도를 이루며 다양한 운영체제에서 동작하는 데이터베이스 관리 시스템이다. DB2는 스토리지 최적화 기술을 통해 막대한 양의 데이터 저장 시 필요한 관련 시스템의 비용을 절감할 수 있다는 장점을 갖고 있다.

5) 티베로

티베로는 데이터베이스 관리 솔루션을 제공하는 티맥스소프트사의 제품이다. 티맥스는 외국 소프트웨어가 주류를 이루는 국내 기업용 소프트웨어 시장에서 국산 소프트웨어 보급을 성공적으로 이끌고 있는 기업이다. 티베로 RDBMS는 대용량 기업 데이터 처리에 최적화된 개발 환경 및 운영 환경을 제공하며 안정성과 성능면에서 인정을 받고 있다.

6) 알티베이스

알티베이스는 MMDBMS분야를 선도하고 있는 국산 DBMS시장의 선두 주자다. MMDBMS는 데이터를 메모리에 저장하고 이를 바로 응용프로그램에 전달함으로써 실시간 데이터 처리에 강점을 지니는 DBMS로서 통신및 금융권을 중심으로 시장을 형성하고 있다.

MMDBMS의 빠른 데이터 처리도 대용량 데이터 처리라는 문제에는 한계가 있었는데, 알티베이스는 대용량 데이터의 실시간 처리를 지원하는 하이브리드 DBMS제품을 출시하여 DBMS의 새로운 대안을 제시했다는 평가를 받기도 했다.

[2] 데이터베이스 설계

1. 요구조건 분석 : 요구조건 명세서 작성
2. 개념적 설계 : 개념스키마, 트랜잭션 모델링, E-R모델
3. 논리적 설계 : 목표 DBMS에 맞는 스키마 설계, 트랜잭션 인터페이스 설계
4. 물리적 설계 : 목표 DBMS에 맞는 물리적 구조의 데이터 변환
5. 구현 : 목표 DBMS의 DDL로 데이터베이스 생성




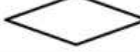


2-2

관계형 데이터베이스 정의

주어진 E-R 다이어그램을 이용하여 관계형 데이터베이스의 테이블을 정의할 수 있다.

[1] ER 다이어그램 작성 방법

1) ERD 표준 기호

기 호	의 미
	개체
	속성
	기본키
	관계
	개체 타입과 속성을 연결
	개체간의 관계 타입

2) 관계 타입의 유형

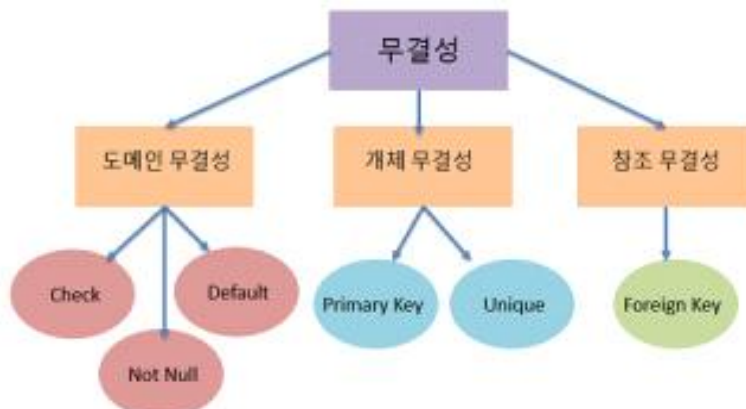
기호	의미	설명
1 — 관계 — 1	일대일 관계	하나의 개체가 하나의 개체에 대응
1 — 관계 — N	일대다 관계	하나의 개체가 여러 개체에 대응
N — 관계 — 1	다대일 관계	여러 개체가 하나의 개체에 대응
M — 관계 — N	다대다 관계	여러 개체가 여러 개체에 대응

3) 용어 정리

- 엔티티(Entity) : 개체
- Relationship : 관계
- Attribute : 속성 (열)
- Cardinality : 객체 간의 관계성, 튜플의 수
- 식별자 : 어떤 대상을 유일하게 구별할 수 있는 이름
- 정규화 : 어떤 대상을 기준에 따르는 정규적인 상태로 바꾸거나 비정상적인 대상을 정상적으로 되돌리는 과정
- 스키마 : 데이터베이스의 전체적인 구조와 제약 조건에 대한 명세를 기술한 것
외부 스키마, 개념 스키마, 내부 스키마

[2] 무결성 제약조건의 이해

1. 개념 : 데이터베이스 내에 있는 데이터의 정확성 유지하며 바람직하지 않은 데이터가 저장되는 것을 방지하는 것
2. 장점 : 테이블 생성시 무결성 제약조건 정의 가능
제약조건을 조작할 수 있는 융통성 (활성화, 비활성화)
3. 제약 조건의 종류



1) 개체 무결성 제약 조건

기본키와 관련된 제약조건으로, 한 릴레이션의 기본키를 구성하는 어떠한 속성값도 절대 널

값이나 중복 값을 가질 수 없음

학번	성명	주민번호	학과	학번	성명	주민번호	학과
100	일순이	111111-1111111	멀티	100	일순이	111111-1111111	멀티
200	이순이	222222-2222222	전자	200	이순이	222222-2222222	전자
300	삼순이	333333-3333333	디자인	300	삼순이	333333-3333333	디자인
	사순이	444444-4444444	컴퓨터	300	사순이	444444-4444444	컴퓨터

Null 값 중복된 값

2) 도메인 제약 조건

- 각 애트리뷰트 값이 반드시 원자값이어야 함
- ex) 학년 필드(int(도메인))에 1, 2 값은 가능하지만 3학년(string)은 불가능

학번	성명	학년	학과
100	김일순	1	보안과
200	김이순	2	디바이스과
300	김삼순	3학년	콘텐츠과

3) 참조 무결성 제약 조건

- 두 테이블이 외래키와 기본키로 서로 연결되어 있을 때, 외래키가 실제 존재하는 기본키와 연결되어 있어야 함.
- 외래키와 관련된 제약조건으로 릴레이션 R1에 저장된 튜플이 릴레이션 R2에 있는 튜플을 참고하려면, 참조되는 튜플이 반드시 R2에 존재해야 함

학과코드	학과명	전화번호	학번	성명	학과
D001	전기	123-4567	100	홍길동	D001
D002	기계	234-5678	200	이순신	D002
D004	컴퓨터	345-6789	300	강감찬	D003
			400	안중근	D005

→ 학번이 400번인 안중근 학생의 학과를 입력하면 학과 릴레이션에는 D005라는 학과가 존재하지 않으므로 참조무결성 교착 위배

- 참조 무결성 제약을 만족시키기 위해 DBMS가 제공하는 옵션
제한(Restricted), 연쇄(Cascade), 널값(Nullify), 디폴트 값(Default)

[3] 테이블 선언 및 조작어

Create table 테이블 이름 (
변수명 자료형(크기) 제약조건,
변수명1 자료형(크기) 제약조건
);

무결성 제약 조건	역할
NOT NULL	NULL을 허용하지 않음
UNIQUE	중복된 값을 허용하지 않고, 항상 유일한 값을 갖도록 함
PRIMARY KEY	NULL을 허용하지 않고, 중복된 값도 허용하지 않음 NOT NULL 조건과 UNIQUE 조건을 결합한 형태
FOREIGN KEY	참조되는 테이블의 컬럼의 값이 존재하면 허용
CHECK	저장 가능한 데이터 값이 범위나 조건을 지정하여 설정한 값만을 허용

[4] 릴레이션의 특징

1. 릴레이션의 특징

- 릴레이션에 포함된 튜플들은 모두 다르다.
- 릴레이션에 포함된 튜플 사이에는 순서가 없다.
- 튜플들의 삽입, 삭제 등의 작업으로 인해 릴레이션은 시간에 따라 변한다.
- 릴레이션 스키마를 구성하는 애트리뷰트들 간의 순서는 중요하지 않다.
- 애트리뷰트의 유일한 식별을 위해 애트리뷰트의 명칭은 유일해야 하지만, 애트리뷰트를 구성하는 값은 동일값이 나올 수 있다.
- 릴레이션을 구성하는 튜플을 유일하게 식별하기 위해 애트리뷰트들의 부분집합을 키로 설정한다.
- 애트리뷰트는 더 이상 쪼갤 수 없는 원자값만을 저장한다.

2. 릴레이션 용어

- 튜플 : 릴레이션을 구성하는 각각의 행. 카디널리티=튜플의 수
- 애트리뷰트 : 데이터베이스를 구성하는 가장 작은 논리적 단위. 차수=애트리뷰트의 수
- 도메인 : 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자값들의 집합
- 릴레이션 인스턴스 : 데이터 개체를 구성하고 있는 속성들에 데이터 타입이 정의되어 구체적인 데이터 값을 갖고 있는 것

2-3

CRUD연산

데이터베이스의 기본연산을 CRUD(Create, Read, Update, Delete)로 구분하여 설명할 수 있다.

[1] CRUD 연산에 대한 이해

CRUD는 대부분의 컴퓨터 소프트웨어가 가지는 기본적인 데이터 처리 기능인 Create(생성), Read(읽기), Update(갱신), Delete(삭제)를 묶어서 일컫는 말이다. 사용자 인터페이스가 갖추어야 할 기능(정보의 참조/검색/갱신)을 가리키는 용어로서도 사용된다.

각 문자는 다음과 같이 표준 SQL문으로 대응 가능하다.

이름	조작	SQL
Create	데이터 생성	INSERT
Read	데이터 읽기, 조회	SELECT
Update	데이터 갱신	UPDATE
Delete	데이터 삭제	DELETE

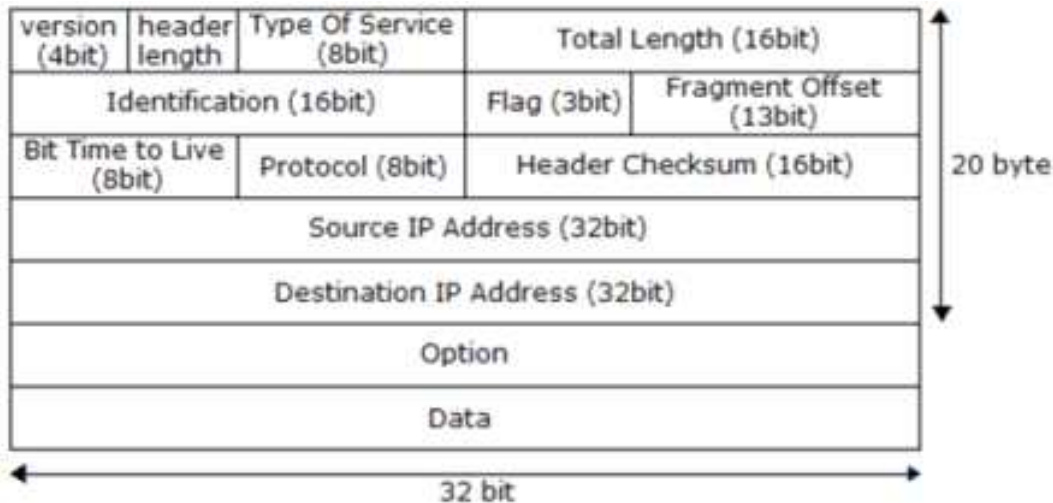
[1] OSI 7레이어

계층		특징
7계층	응용 계층 (Application Layer)	<ul style="list-style-type: none"> • 사용자(응용프로그램)가 OSI 환경에 접근할 수 있도록 서비스를 제공함. • 여러가지 점검 사항을 완벽하게 진행한 후 표현 계층으로 전달 • 예 : TELNET, FTP, SMTP, SNMP, WWW
6계층	표현 계층 (Presentation Layer)	<ul style="list-style-type: none"> • 응용계층으로부터 받은 데이터를 세션 계층에 보내기 전에 통신에 적당한 형태로 변환하고, 세션 계층에서 받은 데이터는 응용 계층에 맞게 변환하는 기능을 함
5계층	세션 계층 (Session Layer)	<ul style="list-style-type: none"> • 개체들 간의 관련성을 유지하고 대화 제어를 담당하는 계층임
4계층	전송 계층 (Transport Layer)	<ul style="list-style-type: none"> • 논리적 안정과 균일한 데이터 전송 서비스를 제공함으로써 종단 시스템(End-to-End) 간에 투명한 데이터 전송을 가능하게 함. • 예 : TCP, UDP • 장비 : L4스위치
3계층	네트워크 계층 (Network Layer)	<ul style="list-style-type: none"> • 개발 시스템들간의 네트워크 연결을 관리하는 기능과 데이터의 교환 및 중계 기능을 함 • 주소 부여(IP), 경로 설정(Route) • 장비 : 라우터
2계층	데이터링크 계층 (Data Link Layer)	<ul style="list-style-type: none"> • 두 개의 인접한 개방 시스템 간에 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 함. • 프레임에 주소 부여(MAC 주소), 에러 검출, 재전송, 흐름제어 • 장비 : 브릿지, 스위치
1계층	물리 계층 (Physical Layer)	<ul style="list-style-type: none"> • 전송에 필요한 장치 간의 실제 접속과 절단 등 기계적, 전기적, 기능적, 절차적 특성을 정의함 • 장비 : 리피터, 허브

[2] TCP/IP

계층	특징	
응용 계층	<ul style="list-style-type: none"> • 전송 계층을 기반으로 한 다수의 프로토콜과 이 프로토콜을 사용하는 응용프로그램을 포괄 	TELNET, FTP, HTTP, SMTP, MINE, SNMP 등
전송 계층	<ul style="list-style-type: none"> • 최종 통신 목적지를 지정하고 오류 없이 데이터를 전송하는 역할 	TCP, UDP
인터넷 계층	<ul style="list-style-type: none"> • 네트워크 접근 계층의 도움을 받아 데이터를 목적지 호스트까지 전달하는 역할 • 네트워크 접근계층과는 달리 논리적 주소인 IP 주소를 사용함 	IP
네트워크 접근 계층	<ul style="list-style-type: none"> • 물리적 네트워크(MAC)를 통한 실제 데이터 송수신 담당 	장치 드라이버 네트워크 하드웨어

2) IP 패킷 구조



[3] 인터넷 구성의 개념

1. 인터넷(Internet)의 개요

- TCP/IP 프로토콜을 기반으로 하여 전 세계 수 많은 컴퓨터와 네트워크들이 연결된 광범위한 컴퓨터 통신망
- 미 국방성의 ARPANET에서 시작
- 유닉스 운영체제 기반
- 인터넷에 연결된 모든 컴퓨터는 고유한 IP를 갖는다.

2. 인터넷 구성

- 인터넷 시스템 : 우리가 매일 사용하는 인터넷을 구성하고 있는 시스템을 인터넷 시스템이라고 함.
인터넷 시스템은 서버와 클라이언트로 구성
 - 서버: 인터넷 서비스를 제공하는 프로그램이나 컴퓨터를 의미하며, 웹 사이트의 정보를 담고 있는 프로그램이나 컴퓨터를 웹 서버라고 부름
 - 클라이언트: 서버가 제공하는 인터넷 서비스를 이용하는 사용자나 사용자가 사용하는 기기를 의미
- 웹 서버 : 사용자가 요청하는 웹 페이지나 프로그램을 실행하여 해당 파일이나 그 결과를 사용자에게 제공하는 역할
- 웹 브라우저 : 웹 서버에서 웹 페이지를 가져오거나 웹 서버로 정보를 보낼 때 사용하는 프로그램

3. 프로토콜 개념

- 개념 : 데이터 통신을 원활하게 하기 위해 필요한 통신 규약
- 주요 기능 : 캡슐화, 연결 제어, 흐름 제어, 오류 제어, 동기화, 순서 제어, 경로 선택, 단편화의 재결합, 다중화 등

3) 프로토콜의 종류

- FTP : 인터넷 상의 컴퓨터들 간에 파일을 교환하기 위한 표준 프로토콜
- Telnet : 다른 사람의 호스트 컴퓨터에 사용권한을 가지고 있다는 가정하에 원격지에서 액세스할 수 있도록 도와주는 프로토콜
- SSH : 신뢰할 수 있는 데이터 스트림을 통해 파일 접근, 파일 전송, 파일 관리를 제공하는 네트워크 프로토콜
- HTTP : 웹문서를 전송하기 위한 규약

- (5) SMTP : 인터넷에서 이메일을 전송하기 위한 프로토콜
- (6) POP : 인터넷에서 이메일을 수신하기 위한 프로토콜
- (7) TCP : 종단간 흐름 제어, 에러 제어 등의 기능을 수행한 연결형 프로토콜
- (8) UDP : 비연결형 전송 프로토콜
- (9) IP : TCP/IP 기반의 인터넷 망을 통해 데이터그램 전달을 담당하는 프로토콜
- (10) ARP : IP와 MAC 주소 간의 동적 매핑 제공
- (11) RARP : MAC 주소와 IP 주소 간의 동적 매핑 제공

4. IP 주소 체계

- 1) IPv4 : 8비트씩 4부분으로 10진수 표시, 32비트로 구성

CLASS	범위
A 클래스	1.0.0.1 ~ 126.255.255.254
B 클래스	128.0.0.1 ~ 191.255.255.254
C 클래스	192.0.0.1 ~ 223.255.255.254
D 클래스	224.0.0.0 ~ 239.255.255.255
E 클래스	240.0.0.0 ~ 255.255.255.254

- 2) IPv6

- 현재 사용되고 있는 IPv4의 단점을 개선하기 위해 개발됨.
- 주소는 CIDR를 기반으로 계층적으로 할당
- IP주소의 길이가 16비트씩 8부분으로 16진수로 표시, 128비트로 구성
- 네트워크 속도의 증가
- 특정한 패킷 인식을 통한 높은 품질의 서비스 제공
- 헤더 확장을 통한 패킷 출처 인증과 데이터 무결성 및 비밀의 보장

5. 트랜스포트 방식 개념

- 1) Stop and Wait 방식

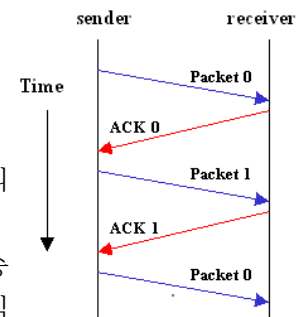
- 검출 후 재전송 방식(ARQ)의 일종
- 송신측에서 1개의 프레임을 송신하고, 수신측에서 수신된 프레임의 에러 유무를 판단하여 송신측에 ACK나 NAK를 보내는 방식
- 송신측은 수신측으로부터 ACK를 수신했을 경우에만 다음 프레임 전송
- 수신측으로부터 NAK를 수신하거나, 일정 시간까지 응답 프레임 (ACK, NAK)를 수신하지 못하면 해당 프레임 재전송
- 구현 방법 단순. 전송 효율 떨어짐.

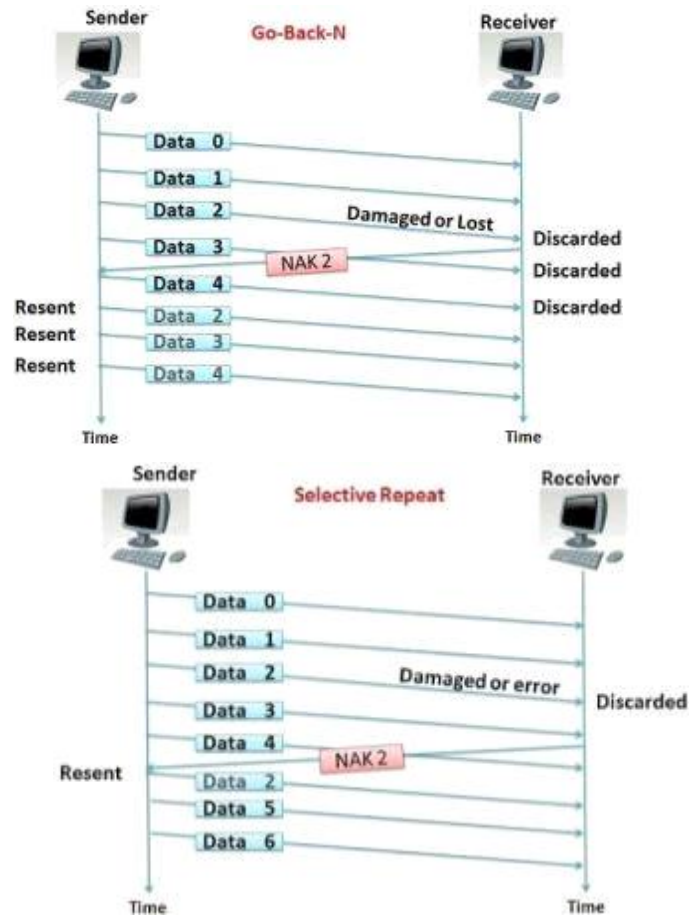
- 2) Go-back-N 방식

- 전송된 프레임이 손상되면 확인된 마지막 프레임 이후로 보내진 프레임을 재전송하는 기법
- 손상된 프레임부터 다시 재전송

- 3) Selective Repeat 방식

- 손상되거나 분실되지 않아도, NAK 혹은 타임아웃 등의 이유로 인해 재전송되는 이후의 모든 프레임을 재전송해야하는 GBn ARQ의 비효율적 문제점을 개선한 방식
- 손상된 프레임만 재전송





3-2

TCP와 UDP

응용의 특성에 따라 TCP와 UDP를 구별하여 적용할 수 있다.

[1] TCP(Transmission Control Protocol)

- 먼저 통신을 할 두 대의 컴퓨터가 연결이 된 상태에서 데이터를 주고 받는 방식
- 연결을 먼저 맺어놓고 연결된 회선을 통해 데이터를 주고 받는 방식
- 신뢰성이 보장되지만 네트워크에 부담이 됨

[2] UDP(User Datagram Protocol)

TCP/IP 프로토콜에서 TCP와 함께 4계층의 주요 프로토콜 중의 하나이다. 간단한 데이터그램 전달계층 프로토콜로 전송방식이 단순하고 서비스의 신뢰성이 낮고 데이터그램의 도착 순서가 바뀐다. 때로는 누락되기도 한다. 그러나 빠른 처리와 실시간 전송에 적합하여 실시간 음성, 방송 서비스에 적용되는 프로토콜이다. 대표적인 서비스로는 TFTP, SNMP등이 있다.

1. UDP 특징

- 8바이트로 포트 번호와 최소한의 에러체크만 하는 매우 단순한 프로토콜이며 신뢰성이 낮다.
- 목적지와 연결설정이 없으므로 비연결형 서비스이다.
- TCP와 달리 Sequence/ACK 번호와 윈도우 사이즈가 없어서 순서제어와 흐름제어가 불가능하다.
- 연결 설정 및 제어기능이 빠지는 대신 빠른 처리가 가능하며 실시간성을 요구하는 서비스에 적합하다.

[3] TCP/UDP 비교

구분	TCP	UDP
연결설정	연결기반(3웨이 핸드셰이킹)	비연결기반(단방향전송)
헤더길이	20바이트(옵션제외)	8바이트
데이터손실	없음	있음
순서/흐름제어	있음	없음
주요특징	신뢰성	실시간, 빠른처리

3-3

패킷 스위칭 시스템

패킷 스위칭 시스템을 이해하고, 다양한 라우팅 알고리즘과 ip프로토콜을 설명할 수 있다.

[1] IP프로토콜(Internet Protocol)

패킷의 목적지 주소를 보고 네트워크에 대한 최적의 경로를 찾아 패킷을 전송해주는 역할이 IP의 주요 기능이다. IP는 비연결성(Connection-less)으로 패킷 손실, 중복, 지연, 순서 변경 등이 가능한 상태에서 최적의 전달 시스템이다. IP는 연결에 어떠한 정보도 관리하지 않으며 신뢰성보다는 효율성에 중점을 두고 최적의 전송 서비스를 하도록 설계되었다. 네트워크에 과부하가 걸리면 패킷을 폐기시킨다. 멀티캐스트 기능이 제공이 가능하다.

4. 화면구현(2001020225_16v4)

1-1

UI설계 내용 확인

설계된 화면과 품의 흐름을 확인하고 제약사항과 화면의 품 흐름을 구현에 반영하도록 설계를 확인할 수 있다.

[1] UI(User Interface)

사용자가 하드웨어나 시스템에 연결되는 과정에서 사용자의 편리성과 가독성을 높여 주기 위한 것

1. CLI(명령어 라인 인터페이스, Command Line Interface): 사용자와 시스템이 상호 작용을 하는 방식으로 명령과 출력이 텍스트 형태로 표현.
2. GUI(그래픽 유저 인터페이스, Graphical User Interface): 사용자가 컴퓨터의 자원을 사용할 수 있도록 구성요소들이 그래픽 형태로 표현.
3. NUI(자연어 유저 인터페이스, Natural User Interface): 몸짓, 말소리만으로도 IT 기기를 사용할 수 있는 사용자 인터페이스.

[2] UI표준 및 지침

1. UI 표준은 시스템 전반에 걸쳐 모든 UI에 공통적으로 적용되는 내용이며, 화면 구성 내용이나 화면 이동 등이 포함됨.
2. UI지침은 UI 개발 과정에서 UI요구사항이나 구현 제약사항 등 지켜져야 할 공통의 조건을 의미함.

[3] 소프트웨어 아키텍처(Software Architecture)

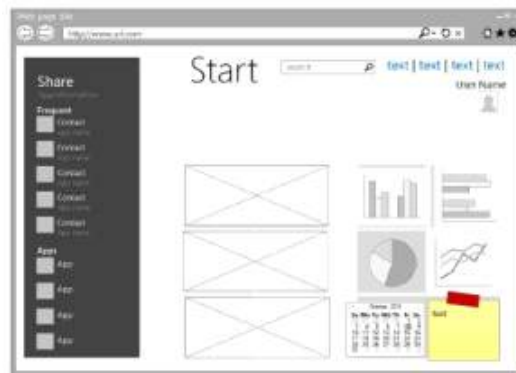
소프트웨어 구성요소 중 외부에 드러나는 요소 특성과 요소들 간의 관계를 표현한 것.

[4] 제약사항(Constraints Requirements)

1. 시스템을 설계하거나 구현할 때 관련되는 기술이나 표준, 규정들을 의미.
2. 개발과 관련된 정책이나 업무규칙, 특정 소프트웨어나 프로그램의 사용, 데이터 사용과 관련된 제약 등에 대한 기술 제약사항(Technical constraints), 개발할 때 적용할 업무 영역의 표준이나 법규의 표준 적합 제약사항(Standard compliance constraints)으로 나눌 수 있음.

[5] UI 개발 도구 분류

1. 화면 설계: 파워포인트(Powerpoint), 스토리보드(Storyboard), 와이어프레임(Wireframe), 목업(Mockup) 등이 있음.



- (1) 와이어프레임: 기획 단계에서 페이지 레이아웃이나 화면 이동, 구성요소에 대한 내용을 기술한 문서.
- (2) 스토리보드: 와이어프레임의 내용에 디스크립션 등을 포함한 설계 문서.

(3) 목업: 설계 단계에서 실제 화면과 같은 형태의 모형.

2. 프로토타입(Prototype): 인터랙션(Interaction, 상호작용)이 포함되어 테스트가 가능한 형태

3. UI디자인 : 화면의 모양이나 기능 등을 표현하는 것.

[6] UI설계서

웹 사이트의 페이지 구성요소를 기록한 설계도이다. 정적인 형태의 화면 형태로 와이어프레임이나 목업 등을 이용하여 작성한다.

1-2

UI 메뉴 구조 확인

- UI요구사항과 UI표준 및 지침에 따라 설계된 메뉴 구조를 해석할 수 있다.
- 구현을 위해 하위 시스템 단위의 내·외부 화면과 품의 설계를 확인할 수 있다.

[1] 사용성(Usability)

특정한 목적을 성취하고자 하는 특정 사용자에게 의해 어떤 제품이 사용될 때와 같은 특정한 맥락의 사용에서의 효과성, 효율성, 만족도에 관한 것

[2] 유용성(Usefulness)

사용자가 업무를 수행하는 데 있어 얼마나 정확하게 수행할 수 있는지를 나타내는 것.

[3] 정보 구조(Information Architecture)

설계 단계에서 사이트를 구성하는 처리 내용이나 메뉴의 구조를 표현함으로써 사이트의 구조를 파악할 수 있도록 하는 것으로 사이트 맵이라고도 함. 정보 구조를 표현하는 방법에 따라 계층적 구조, 계열 구조, 그리드 구조, 네트워크 구조 등으로 나눔.

[4] 내비게이션(Navigation)

1. 사용자가 사이트에서 원하는 정보를 찾도록 안내하는 것.
2. 사이트 맵(사이트 구조)과 관계가 있고 설계나 구현할 때 사용자가 중심이 되어야 함.
3. 내비게이션은 메뉴, 버튼, 링크 등으로 구성되며 이들 구성요소는 모든 페이지에서 일관성을 지켜 사용자로 하여금 혼동되지 않도록 설계되어야 함.



[그림] 사이트 메뉴 구조도

[5] 유스케이스(Use Case)

1. 사용자 측면에서의 요구사항이며 주로 기능 개선에 대한 내용으로, 사용자가 원하는 목표를 위하여 시스템에서 수행해야 하는 내용을 기술.
2. 유스케이스에 기록된 내용을 토대로 실제 수행 방법을 구현하게 됨.
3. 사용자에게 외부 시스템이나 서버 시스템과의 의사소통 수단으로 사용됨.
4. 하나의 단위 업무에 대한 독립적인 기능을 수행할 수 있도록 표현함.

2-1

UI구현 지침 확인

소프트웨어 아키텍처 세부 구현 지침과 UI표준 및 지침을 반영하여 UI설계를 구현할 수 있다.

[1] W3C(World Wide Web Consortium)

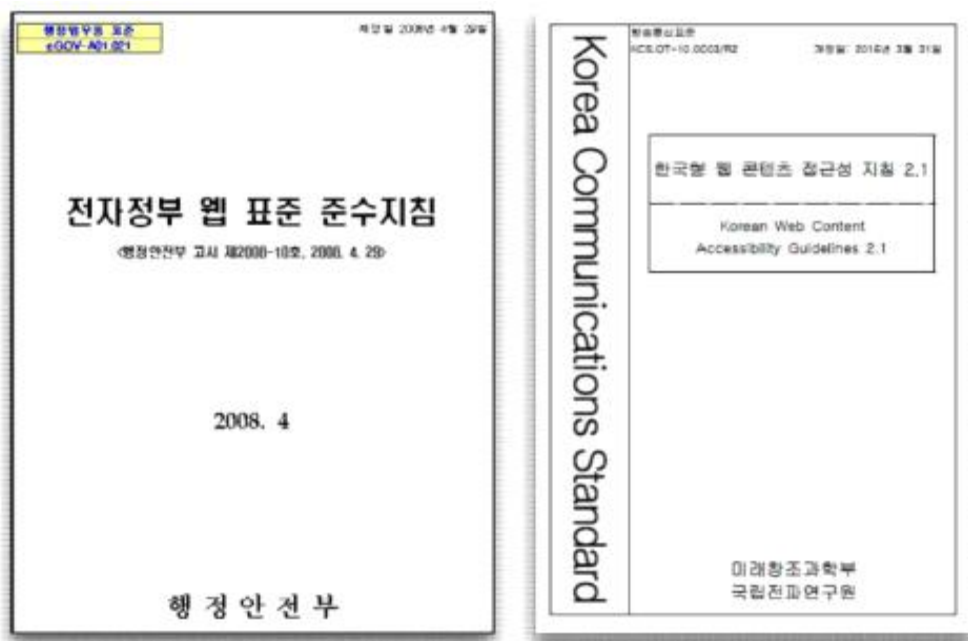
월드 와이드 웹을 위한 표준을 개발하고 장려하는 국제적인 컨소시엄 조직

[2] 한국형 웹 콘텐츠 접근성 지침(KWCAG, Korean Web Content Accessibility Guidelines)

1. 장애인이 비장애인과 동등하게 웹 콘텐츠에 접근할 수 있도록 웹 콘텐츠를 제작하는 방법에 관해 기술한 것
2. 웹 콘텐츠 저작자 및 개발자, 웹 사이트 설계자 등이 장애인 접근성을 준수하여 콘텐츠를 쉽게 제작할 수 있도록 돕는 지침들을 제공하는 것이 목적임.
3. 기존의 ‘인터넷 웹 콘텐츠 접근성 지침’에 해외 웹 관련 표준 및 기술 동향을 최대한 반영하여 개정한 것
4. 해외의 ‘웹 콘텐츠 접근성 가이드라인 2.0을 국내 실정에 맞게 반영한 것

[3] 전자정부 웹 표준 준수 지침

전자정부 웹 사이트 이용자가 특정 운영 체제나 웹 브라우저에 상관없이 접속할 수 있도록, 정부에서 전자정부시스템 구축 시 반영해야 하는 최소한의 규약을 정의한 것.



[그림]전자정부 웹 표준 준수 지침서와 한국형 웹 콘텐츠 접근성 지침서

[4] 웹의 3요소

1. 웹 표준(Web Standards)

웹에서 사용되는 기술이나 규칙을 의미하며 웹 사이트를 작성할 때 이용하는 HTML, CSS, JavaScript 등에 대한 규정과 다른 기종이나 다른 플랫폼에서도 웹페이지가 구현되도록 제작하는 기법 등을 포함.

2. 웹 접근성(Web Accessibility)

어떠한 사용자(장애인, 노인 등), 어떠한 기술 환경에서도 사용자가 전문적인 능력 없이 웹 사이트에서 제공하는 모든 정보에 접근할 수 있도록 보장하는 것을 뜻함.

3. 웹 호환성(Cross Browsing)

서비스 이용자 단말기의 하드웨어 및 소프트웨어 환경이 다른 경우에도 동등한 서비스 제공하는 것.



- 확인된 화면과 폼 흐름 설계에 따라, 사용자 접근성을 고려한 화면과 폼의 흐름 제어를 구현할 수 있다.
- 확인된 화면과 폼 흐름 설계에 따라, 감성공학 기법을 고려하여 접하는 화면, 홈, 메뉴, 흐름을 구현할 수 있다.

[1] 서버(Server)와 클라이언트(Client)



1. 서버: 클라이언트에게 네트워크를 통해 정보나 서비스를 제공하는 컴퓨터 또는 프로그램.

1) 웹 서버는 웹 브라우저(클라이언트)로부터 HTTP 요청을 받아들이고, HTML 문서(웹 페이지)를 반환.

가) 웹 서버의 기능: HTML 문서(웹 페이지)를 클라이언트로 전달

나) HTML 문서(웹 페이지)에는 그림, CSS, 자바스크립트가 포함된다.

다) 콘텐츠 제공뿐 아니라 클라이언트로부터 콘텐츠를 전달 받는 것도 웹 서버의 기능에 속함.

2) 서버 프로그램의 종류 : Apache, IIS, nginx, GWS 등.

종류	내용
Apache	Apache 재단에서 제작한 오픈 소스로 거의 모든 운영체제에서 사용 가능
IIS	마이크로소프트 제작, 사용으로 사용 가능하며 시장 점유율이 높아지고 있음
nginx	NGINX 제작한 오픈소스로 프록시 서버를 포함
GWS	구글에서 제작

2. 클라이언트

1) 네트워크를 통해 다른 서버 시스템의 컴퓨터 원격 서비스에 접속할 수 있는 응용 프로그램이나 서비스.

2) 기능: 웹 브라우저에 URL을 입력하여 그 URL에 해당하는 웹 서버로 웹 페이지에 대한 요청 전달.

3) 종류: 파이어폭스, 크롬, IE 등.

종류	내용
인터넷익스플로러	마이크로소프트웨어에서 개발. 윈도우 운영체제에서만 사용 가능
파이어폭스	개방형 브라우저
크롬	구글에서 개발한 브라우저, 다양한 운영체제에서 사용 가능
사파리	애플이 개발한 웹 브라우저
오페라	낮은 인지도에 비해 웹 표준을 많이 구현한 브라우저

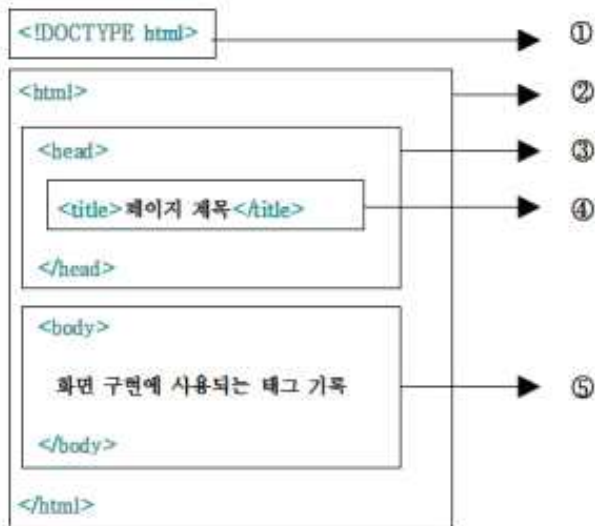
[2] 웹 사이트(Web Site)와 웹 페이지(Web Page)

1. 웹 사이트는 인터넷 프로토콜 기반의 네트워크에서 URL을 통하여 보이는 웹 페이지들의 의미있는 묶음

2. 웹 페이지는 월드 와이드 웹(WWW, World Wide Web) 상에 있는 개개의 문서를 의미함.

[3] HTML(Hyper Text Markup Language)

1. HTML의 구조

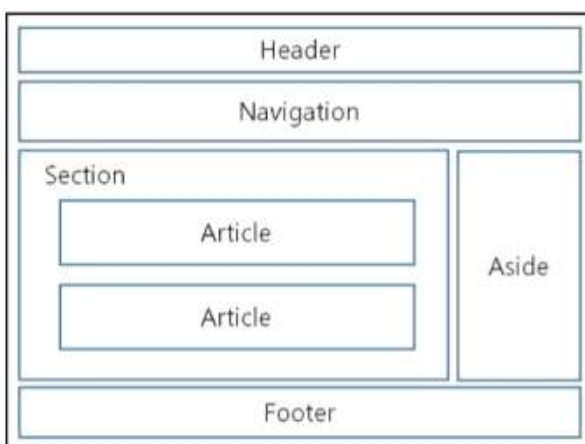


구분	이름	역할 또는 기능
①	<code><!DOCTYPE html></code>	웹 페이지가 HTML5 문서임을 의미. HTML5에는 반드시 표기해야 함.
②	<code><html>...</html></code>	HTML 문서의 시작과 끝을 의미
③	<code><head>...</head></code>	스타일과 스크립트를 선언하는 부분
④	<code><title>...</title></code>	브라우저의 제목 표시
⑤	<code><body>...</body></code>	사용자에게 보여주는 실제 내용이 구현되는 부분

2. HTML 태그의 이해

- 1) HTML의 모든 태그는 시작과 끝 태그로 이루어진다.
- 2) HTML 태그에는 속성을 지정할 수 있다.
- 3) 웹 브라우저에 보여지는 내용들을 표현한다.

3. HTML 레이아웃 관련 태그



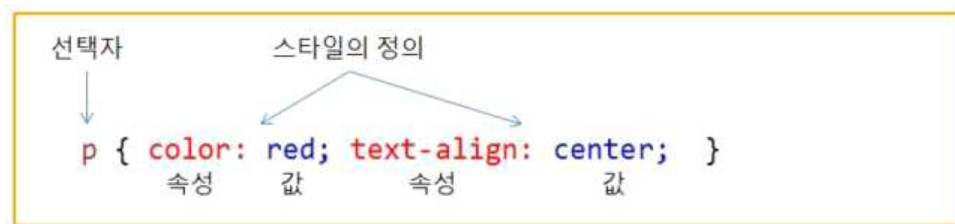
- ① Header: 해당 페이지의 헤더 영역을 지정하고 주로 로고나 회사명, 사이트 맵, 로그인/회원가입 버튼, 검색 버튼 등이 위치.
- ② Navigation: 본문의 주요 내비게이션(메인 메뉴) 영역을 지정.
- ③ Section: 해당 페이지의 콘텐츠 영역을 지정할 때 사용하며 헤더, 푸터 태그와 비교해서 영역을 구분 지정할 때 사용.
- ④ Article: 독립적인 콘텐츠 항목에 대한 영역을 지정할 때 사용.

- ⑤ Aside: 본문 내용 이외에 표현하고자 하는 기타 내용이 있을 경우에 영역을 지정 할 때 사용하거나 서브 메뉴를 표시하고자 할 때 사용.
- ⑥ Footer: 본문 내용의 아래에 위치하도록 지정하며 주로 개인정보 보호정책, 회사 주소 등을 작성

[4] CSS(Cascading Style Sheets)

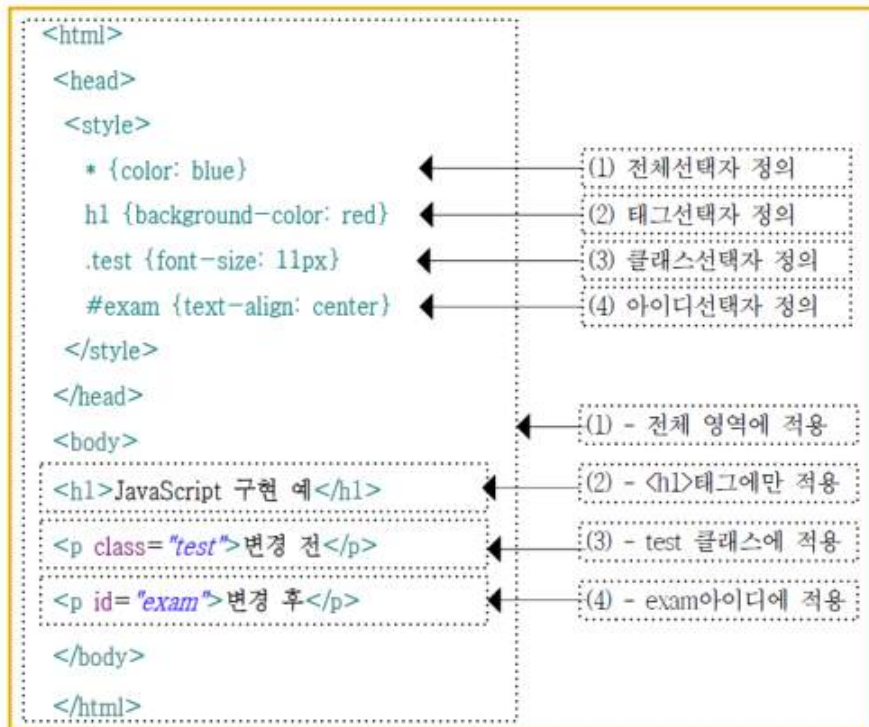
웹 페이지 전체의 일관성을 유지할 수 있도록 스타일을 미리 저장해둔 시트를 의미함. 웹 페이지를 HTML로만 작성할 경우에는 스타일 변경에 제약이 있으나, 스타일을 미리 저장해 두고 일부 내용의 스타일을 변경할 때 전체 페이지의 내용을 한꺼번에 처리할 수 있어 간편하고 페이지의 일관성을 유지할 수 있음.

- 1) CSS문법: CSS는 HTML 문서 내에 작성되거나 외부 파일 형태로 작성 가능하며, CSS문법은 다음의 예와 같이 구성됨.



2) CSS선택자

- ① 공용 선택자(Universal Selector, 일반 선택자) : HTML 요소를 선택자로 하여 스타일을 적용할 수 있으며, 이 경우 해당 HTML 요소 모두에 스타일이 적용됨.
예) * { 속성이름 : 값; 속성이름 : 값; 속성이름 : 값; }
- ② 태그 선택자(Type Selector) : 지정한 태그에 대하여 스타일이 적용됨.
예) 태그{ 속성이름 : 값; 속성이름 : 값; 속성이름 : 값; }
- ③ 클래스 선택자(Class Selector) : 클래스 선택자를 활용하면 특정 HTML 요소들을 그룹화하여 스타일을 지정하는 것이 가능함.
예) . 클래스명{ 속성이름: 값; 속성이름: 값; 속성이름: 값; }
- ④ 아이디 선택자(ID Selector) : 특정 ID를 부여하여 ID에 스타일을 지정할 수 있음.
예) id명{ 속성이름: 값; 속성이름: 값; 속성이름: 값; }
- ⑤ CSS 선택자 사용



[5] 자바스크립트(JavaScript)

자바스크립트는 객체지향의 프로그래밍 언어로서 웹 브라우저에서 주로 사용된다. 자바스크립트를 이용하여 웹 페이지에서 발생하는 사용자 이벤트에 대한 처리가 가능하고, 자바스크립트의 내장 객체를 활용하면 다양한 형태의 웹 페이지를 구현할 수 있다.

1. 자바스크립트 구현 예

```

<!DOCTYPE html>
<html>
<head>
<script>
    function myFunction() {
        document.getElementById("exam").innerHTML = "변경 후"
    }
</script>
</head>
<body>
<h1>JavaScript 구현 예</h1>
<p id="exam">변경 전</p>
<button type="button" onclick="myFunction()">변경</button>
</body>
</html>

```


2. 자바스크립트 작성 방법 : 자바스크립트는 HTML 문서 내에서 `<script></script>` 태그를 통해 작성되고, 작성되는 위치는 `<head>` 영역, `<body>` 영역이며, js확장자를 갖는 외부 파일 형태로 작성할 수 있다.

(1) `<body>` 영역 내부에, `<script>` 태그 내부에 자바스크립트 코딩을 작성한다.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript 구현 예</h1>
<p id="exam">변경 전</p>
<button type="button" onclick="myFunction()">변경</button>
<script>
function myFunction() {
document.getElementById("exam").innerHTML = "변경후"
}
</script>
</body>
</html>
```

- (2) .js 확장자를 갖는 별도의 외부 자바스크립트 코드 파일을 생성하고, 이를 HTML 페이지에 적용할 수 있다. 이 방식의 장점은 동일한 자바스크립트 코드를 여러 HTML 페이지에 적용할 수 있다는 점이다.

```
<!DOCTYPE html>
<html>
<body>
<script src="mainScript.js"></script>
</body>
</html>
```

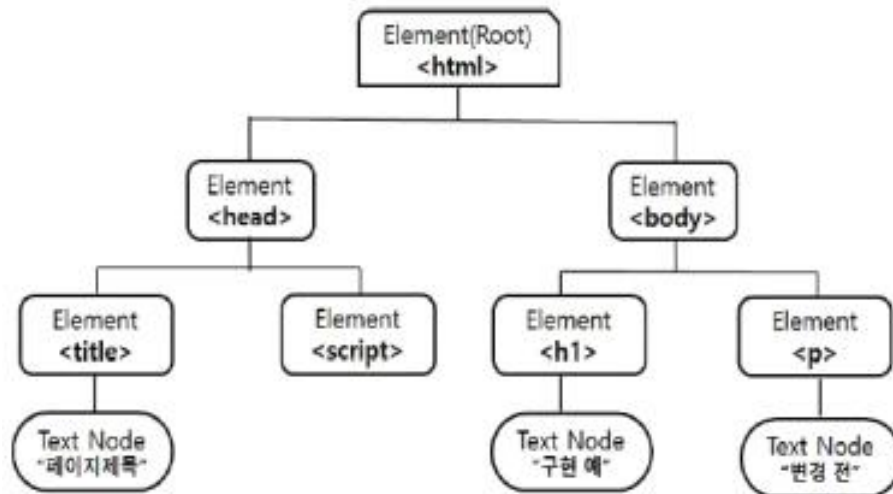
3. HTML의 이벤트 처리: `<script>` 태그 내부의 자바스크립트 코드는 해당 HTML 파일이 로딩될 때 한 번 실행된다. 이런 정적인 코드만을 사용할 경우 사용자 이벤트에 대한 다양한 처리에 어려움이 있으며, 이 때문에 HTML에서는 HTML 태그의 여러 속성을 통해 자바스크립트 코드를 처리할 수 있도록 하고 있다.

- (1) onclick
- (2) onmousedown, onmouseup
- (3) onmouseover, onmouseout
- (4) onchange
- (5) onload

4. DOM(Document Object Model)과 자바스크립트의 관계

- (1) DOM은 자바스크립트에서 HTML 페이지의 요소에 접근하거나 변경할 수 있도록 한다.
- (2) 웹 브라우저는 웹 페이지가 로딩될 때 해당 페이지에 대한 DOM을 생성하며, 이 HTML,

DOM은 해당 페이지의 요소들을 아래와 같이 계층 구조를 갖는 형태로 나열한다.



(3) HTML DOM을 통해 자바스크립트는 다음과 같이 구현할 수 있다.

- (가) 해당 페이지의 요소 변경이나 추가, 삭제
- (나) 해당 페이지의 요소에 대한 속성 변경이나 추가
- (다) 해당 페이지의 CSS의 변경이나 추가, 삭제
- (라) 새로운 HTML 이벤트의 생성

[1] 사용성 테스트(Usability Test)

1. 목적

- (1) 사용성 테스트 기법은 UI테스트에 대한 보편적인 방법으로, 크게 사용성 테스트에 대한 계획 수립, 사용성 테스트 설계, 사용성 테스트 수행과 결과 검토로 구분하여 진행된다.



- (2) 사용자의 제품 만족도 제고, 제품의 완성까지의 노력 대비 높은 효율성, 개발 생산성 대비 낮은 에러, 쉬운 학습 이해도, 높은 품질의 완성도 상승, 사용자 실수 치소화 등의 효과를 달성할 수 있다.

2. 특징

- (1) 사용자가 웹 사이트를 사용하는 것을 전제로 분석하여 사이트 구성 메뉴의 문제점과 요구사항의 반영 여부를 점검하는 것이다.
- (2) 웹 사이트가 개발된 조건을 바탕으로 사용자가 접근하고 이용하는 측면에서 선호할만한 웹 사이트의 기능을 테스트하는 것이다.
- (3) 사용자의 요구사항에 맞는 웹 사이트를 만들어 내기 위하여 구현 전체 절차에 사용자의 요구사항이 반영되도록 하는 것이다.

3. 사용성 테스트 주의 사항

- (1) 사용성에만 치중한 나머지 고객의 감성적인 측면에 그칠 수 있다.
- (2) 기본 기능이 정확히 구현되었는지 검사하는 기본적인 기능 테스트에 대한 누락이 발생하지 않도록 주의한다.
- (3) 테스트에 참여하는 인원이 자율적으로 테스트할 수 있는 환경이어야 좋은 품질 보장이 가능하다.
- (4) 테스트할 때 자세한 기능 설명, 구현 방법에 대해서 충분한 테스트가 가능하도록 제공되어야 한다.
- (5) 테스트가 오랜 시간 진행될 경우 중간에 어떤 사유에서 지연되는지 체크해야 한다.

Web page title

http://www.url.com

Markup 검증 서비스

파일업로드 URL 코드입력

파일 업로드 유효성검사

파일 : D:\.....\main.html

추가옵션

Tooltip Text Tooltip Text Tooltip Text Tooltip Text

[그림] HTML 검증 서비스 프로그램 사용 예시

Web page title

http://www.url.com

CSS 검증 서비스

파일업로드 URL 코드입력

파일 업로드를 통한 검사

CSS 문서 : D:\.....\page\data\main.....

추가옵션

프로파일: CSS 3 중간: Tooltip Text
경고: Tooltip Text 제조사: Tooltip Text

[그림] CSS 검증 서비스 프로그램 사용 예시

5. SQL 활용(2001020413_16v3)

1-1

DDL 활용

테이블의 구조와 제약 조건을 생성, 삭제하고 수정하는 DDL(Data Definition Language) 명령문을 작성할 수 있다.

[1] DDL 개요

1. DDL(Data Definition Language)은 ‘데이터를 정의하는 언어’

DDL 대상	설명
스키마(Schema)	<ul style="list-style-type: none"> DBMS 특성과 구현 환경을 감안한 데이터 구조 직관적으로 하나의 데이터베이스로 이해 가능
도메인(Domain)	<ul style="list-style-type: none"> 속성의 데이터 타입과 크기, 제약 조건 등을 지정한 정보 속성이 가질 수 있는 값의 범위로 이해 가능
테이블(Table)	<ul style="list-style-type: none"> 데이터 저장 공간
뷰(View)	<ul style="list-style-type: none"> 하나 이상의 물리 테이블에서 유도되는 가상의 논리 테이블
인덱스(Index)	<ul style="list-style-type: none"> 검색을 빠르게 하기 위한 데이터 구조

2. DDL 조작 방법

구분	DDL 명령어	내용
생성	• CREATE	데이터베이스 오브젝트 생성
변경	• ALTER	데이터베이스 오브젝트 변경
삭제	• DROP • TRUNCATE	데이터베이스 오브젝트 삭제 데이터베이스 오브젝트 내용 삭제

[2] DDL 활용

구분	문법	
테이블 생성	CREATE TABLE 테이블이름 (열이름 데이터 타입 [DEFAULT 값] [NOT NULL] {열이름 데이터 타입 [DEFAULT 값] [NOT NULL] }* [PRIMARY KEY (열 리스트),] {[FOREIGN KEY (열 리스트) REFERENCES 테이블이름 [(열이름)] [ON DELETE 옵션] [ON UPDATE 옵션]], }* [CHECK (조건식) UNIQUE(열이름)]) ;	
다른 테이블 정보를 이용한 테이블 생성	CREATE TABLE 테이블이름 AS SELECT 문;	
테이블 변경	열 추가	ALTER TABLE 테이블이름 ADD 열이름 데이터타입 [DEFAULT 값]
	열 데이터 타입 변경	ALTER TABLE 테이블이름 MODIFY 열이름 데이터타입 [DEFAULT 값]
	열 삭제	ALTER TABLE 테이블이름 DROP 열이름
테이블 삭제, 이름 변경	테이블 삭제	DROP TABLE 테이블이름
	테이블 내용 삭제	TRUNCATE TABLE 테이블이름
	테이블이름 변경	RENAME TABLE 이전테이블이름 TO 새로운테이블이름 ALTER TABLE 이전테이블이름 RENAME 새로운테이블이름

[3] 제약 조건 적용

제약 조건	설명
PRIMARY KEY	테이블의 기본키를 정의함. 기본으로 NOT NULL, UNIQUE 제약이 포함됨.
FOREIGN KEY	외래키를 정의함. 참조 대상을 테이블이름(열이름)으로 명시해야 함. 참조 무결성 위배 상황 발생 시 처리 방법으로 옵션 지정 가능 - NO ACTION, SET DEFAULT, SET NULL, CASCADE
UNIQUE	테이블 내에서 열은 유일한 값을 가져야 함. 테이블 내에서 동일한 값을 가져서는 안 되는 항목에 지정함.
NOT NULL	테이블 내에서 관련 열의 값은 NULL일 수 없음. 필수 입력 항목에 대해 제약 조건으로 설정함.
CHECK	개발자가 정의하는 제약 조건 상황에 따라 다양한 조건 설정 가능

[4] 제약 조건 변경 SQL 문

제약 조건 변경 내용	SQL 문
제약 조건 추가	ALTER TABLE 테이블이름 ADD [CONSTRAINT 제약조건이름] 제약조건(열이름)
제약 조건 삭제	ALTER TABLE 테이블이름 CONSTRAINT 제약조건이름 테이블이름 PRIMARY KEY FOREIGN KEY(열이름) UNIQUE(열이름)
제약 조건 비활성화	ALTER TABLE 테이블이름 DISABLE CONSTRAINT 제약조건이름
제약 조건 활성화	ALTER TABLE 테이블이름 ENABLE CONSTRAINT 제약조건이름

- PRIMARY KEY 제약을 추가하려면 관련 열이 NULL 값을 가져도, 동일한 값을 가져도 안 된다.
- UNIQUE 제약은 관련 열이 유일한 값 또는 NULL 값을 가져야 한다.
- FOREIGN 키와 관련된 열은 해당 열의 값이 NULL이거나 참조하는 테이블에 있어야 한다.

1-2

DML 활용

한 개의 테이블에 대해 데이터를 삽입, 수정, 삭제하고 행을 조회하는 DML(Data Manipulation) 명령문을 작성할 수 있다.

[1] DML개요

구분	DML명령어	내용
데이터 생성	INSERT	삽입 형태로 신규 데이터를 테이블에 저장
데이터 조회	SELECT	테이블의 내용을 조회
데이터 수정	UPDATE	테이블의 내용을 변경
데이터 삭제	DELETE	테이블의 내용을 삭제

[2] DML 명령문

1. 데이터 삽입(INSERT)

INSERT 명령어
INSERT INTO table_name (column1, column2, ..) VALUES (value1, value2, ...);
INSERT INTO table_name VALUES (value1, value2, ...);

2. 데이터 조회(SELECT)

SELECT 명령어		
SELECT [OPTION] columns FROM table [WHERE절] ;		
SELECT문 요소	요소값	내용
OPTION	- ALL - DISTINCT	- 중복 포함한 조회 결과 출력 - 중복 제거한 조회 결과 출력
columns	- 컬럼명 목록 - 와이드카드	- SELECT 통해 조회할 컬럼명 지정 - 모두 또는 전체를 의미하는 *

3. 데이터 수정(UPDATE)

UPDATE 명령어
UPDATE table SET column1 = value1, column2 = value2, ... [WHERE 절];

4. 데이터 삭제(DELETE)

DELETE 명령어
DELETE FROM table [WHERE 절];

1-3

DCL 활용

업무 단위인 트랜잭션의 완료와 취소를 위한 DCL(Data Control Language) 명령문을 작성할 수 있다.

[1] DCL(Data Control Language)개요

1. DCL 유형

오브젝트	목적	내용
사용자 권한	접근 통제	사용자를 등록하고, 사용자에게 특정 데이터베이스를 사용할 수 있는 권리를 부여하는 작업
트랜잭션	안전한 거래 보장	동시에 다수의 작업을 독립적으로 안전하게 처리하기 위한 상호 작용 단위

2. DCL 명령어

유형	명령어	용도
DCL	GRANT REVOKE	데이터베이스 사용자 권한 부여 데이터베이스 사용자 권한 회수
TCL	COMMIT ROLLBACK CHECKPOINT	트랜잭션 확정 트랜잭션 취소 복귀지점 설정

[2] DCL 활용

1. 사용자 권한 부여

권한	명령어 문법
시스템 권한	GRANT 권한1, 권한2 TO 사용자계정
객체 권한	GRANT 권한1, 권한2 ON 객체명 TO 사용자계정

구분	권한	내용
시스템 권한	CREATE USER	계정 생성 권한
	DROP USER	계정 삭제 권한
	DROP ANY TABLE	테이블 삭제 권한
	CREATE SESSION	데이터베이스 접속 권한
	CREATE TABLE	테이블 생성 권한
	CREATE VIEW	뷰 생성 권한
	CREATE SEQUENCE	시퀀스 생성 권한
	CREATE PROCEDURE	함수 생성 권한
객체 권한	ALTER	테이블 변경 권한
	INSERT	데이터 조작 권한
	DELETE	
	SELECT	
	UPDATE	
	EXECUTE	PROCEDURE 실행 권한

2. 사용자 권한 회수

권한	명령어 문법
시스템 권한	REVOKE 권한1, 권한2 FROM 사용자계정
객체 권한	REVOKE 권한1, 권한2 ON 객체명 FROM 사용자계정

[3] DCL 이론적 배경인 접근 통제

1. 접근 통제 개념 : 보안 정책에 따라 접근 객체(시스템 자원, 통신 자원 등)에 대한 접근 주체(사용자, 프로세스 등)의 접근 권한 확인 및 이를 기반으로 한 접근 제어를 통해 자원에 대한 비인가된 사용을 방지하는 정보 보호 기능

2. 접근 통제 정책에 따른 유형

구분	권한	내용
임의 접근 통제 (DAC: Discretionary Access Control)	<ul style="list-style-type: none"> - 시스템 객체에 대한 접근을 사용자 개인 또는 그룹의 식별자를 기반으로 제한하는 방법 - 여기서 임의적이라는 말은 어떤 종류의 접근 권한을 갖는 사용자는 다른 사용자에게 자신의 판단에 의해서 권한을 줄 수 있다는 것임. - 주체와 객체의 신분 및 임의적 접근 통제 규칙에 기초하여 객체에 대한 주체의 접근을 통제하는 기능 	<ul style="list-style-type: none"> - 통제 권한이 주체에 있음. - 주체가 임의적으로 접근 통제 권한을 배분하여 제어할 수 있음.
강제 접근 통제 (MAC: Mandatory Access Control)	<ul style="list-style-type: none"> - 정보시스템 내에서 어떤 주체가 특정 객체에 접근하려 할 때 양쪽의 보안 레이블(Security Label)에 기초하여 높은 보안 수준을 요구하는 정보(객체)가 낮은 보안 수준의 주체에게 노출되지 않도록 접근을 제한하는 통제 방법 	<ul style="list-style-type: none"> - 통제 권한이 제3자에게 있음. - 주체는 접근 통제 권한과 무관함.

3. 접근 통제와 DCL 관계

강제 접근 통제의 경우, 제3자의 종류에 따라 보다 세분화된 정책이 존재한다. 접근 통제 정책의 두 가지 가운데 데이터베이스관리시스템(DBMS)에서 채택한 접근 통제 정책은 임의 접근 통제, DAC 방식이다. 데이터베이스관리, 특히 접근 통제 용도로 SQL에서 사용하는 명령어가 바로 DCL(Data Control Language)인 것이다.

[4] TCL 활용 방법

1. 트랜잭션

- 트랜잭션은 논리적 연산 단위이다.
- 한 개 이상의 데이터베이스 조작이다. 즉, 하나 이상의 SQL 문장이 포함된다.
- 트랜잭션은 '거래'다. 이때 거래 결과가 모두 반영되거나 또는 모두 취소되어야 한다.
- 분할할 수 없는 최소 단위이다.

특성	내용	비고
원자성 (Atomicity)	트랜잭션 안에 정의된 연산은 모두 실행되거나 모두 실행되지 않아야 함.	All or Nothing
일관성 (Consistency)	트랜잭션 실행 전과 후 동일하게 오류가 없어야 함.	무결성
고립성 (Isolation)	트랜잭션 실행 중 다른 트랜잭션에 영향을 받지 않아야 함.	독립성
지속성 (Durability)	트랜잭션 결과는 항상 보존됨.	장애 대응성

2. 트랜잭션 제어

명령어	내용	메모리 조작
COMMIT	메모리의 내용을 하드디스크에 저장	영구히 저장
ROLLBACK	메모리의 내용을 하드디스크에 저장하지 않고 버림	메모리 내용 무효화
CHECKPOINT	ROLLBACK 범위 설정을 위해 메모리상에 경계를 설정	상태기억

1-4

데이터 사전 검색

생성된 테이블의 목록, 테이블의 구조와 제약 조건을 파악하기 위해 사전을 조회하는 명령문을 작성할 수 있다.

[1] 데이터 사전

1. 데이터사전

데이터 사전(Data Dictionary)에는 데이터베이스의 데이터를 제외한 모든 정보가 있다. 데이터 사전의 내용을 변경하는 권한은 시스템이 가지며, 사용자에게는 읽기 전용 테이블이 제공되므로 단순 조회만 가능하다.

데이터를 제외한(데이터를 구성하는) 모든 정보라는 것은 데이터의 데이터를 의미한다. 따라서 데이터 사전은 메타데이터(Meta data)로 구성되어 있다고 할 수 있다.

2. 데이터 사전 내용

- 사용자 정보(아이디, 패스워드 및 권한 등)
- 데이터베이스 객체 정보(테이블, 뷰, 인덱스 등)
- 무결성 제약 정보
- 함수, 프로시저 및 트리거 등

3. 데이터 사전 용도

사용자에게 데이터 사전은 단순 조회의 대상일 뿐이다. 하지만 데이터베이스 엔진을 이루는 컴파일러, 옵티마이저 등과 같은 구성 요소에 데이터 사전은 작업을 수행하는 데 필요한 참조 정보일 뿐만 아니라 작업의 대상이기도 하다.

[2] 데이터 사전 검색

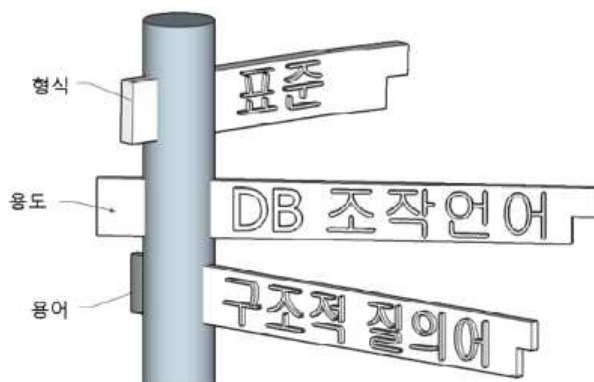
1. 데이터 사전검색

오라클 사용자는 뷰로 데이터 사전에 접근할 수 있다. 오라클에서 데이터 사전과 관련된 뷰는 세 가지 영역이 있으며, 이때 오브젝트에 접근할 수 있는 사용자 권한에 따라 다음과 같이 구분된다.

DBA_ > ALL_ > USER_		
영역	검색 범위	데이터 사전 검색 쿼리문(예시)
DBA_	데이터베이스의 모든 객체 조회 가능 (DBA_는 시스템 접근 권한 의미)	select * from DBA_TABLES select * from DBA_INDEXES select * from DBA_VIEWS
ALL_	자신의 계정으로 접근 가능한 객체와 타 계정의 접근 권한을 가진 모든 객체 조회 가능	select * from DBA_TABLES select * from DBA_INDEXES select * from DBA_VIEWS
USER_	현재 자신의 계정이 소유한 객체 조회 가능	select * from DBA_TABLES select * from DBA_INDEXES select * from DBA_VIEWS

[3] SQL 개요

1. SQL을 바라보는 관점



관점	SQL 의미	시사점
용어	구조적 질의 언어	구조적 개념에 대한 이해 필요
용도	데이터베이스 조작 언어	데이터베이스에 대한 이해 필요
형식	ISO/ANSI 표준으로 정의됨	표준의 변화에 대한 이해 필요

2. 용어 관점에서의 SQL

SQL은 Structured Query Language를 의미하며, 우리말로 옮기면 '구조적 질의어'가 된다. 즉, 구조적 질의어라는 말은 '非절차적 질의어'라는 것으로, 절차적으로 세세하게 묘사하는 것이 아니라는 뜻이다. 그 어떤 절차가 아닌 수학적과 같은 하나의 표현으로 목적하는 대상을 표현하는 수단으로 이해하는 것이 적절하다.

3. 표준 관점에서의 SQL

SQL은 데이터베이스를 조작하는 언어의 표준이다. SQL에 대한 표준은 ANSI(American National Standard Institute)라는 기관이 최초로 만들었으며, 이후 국제 표준화 기구(ISO: International Standard Organization)를 통해 관리되고 있다.

4. 용도 관점에서의 SQL

1) 데이터베이스의 개념

데이터베이스는 '연관된 데이터의 모음'이라 이해할 수 있으며, 보다 엄격하게 '데이터를 일정한 형태로 저장해 놓은 것'을 의미한다. 이러한 데이터 모음을 조작하는 것이 SQL이다. 하지

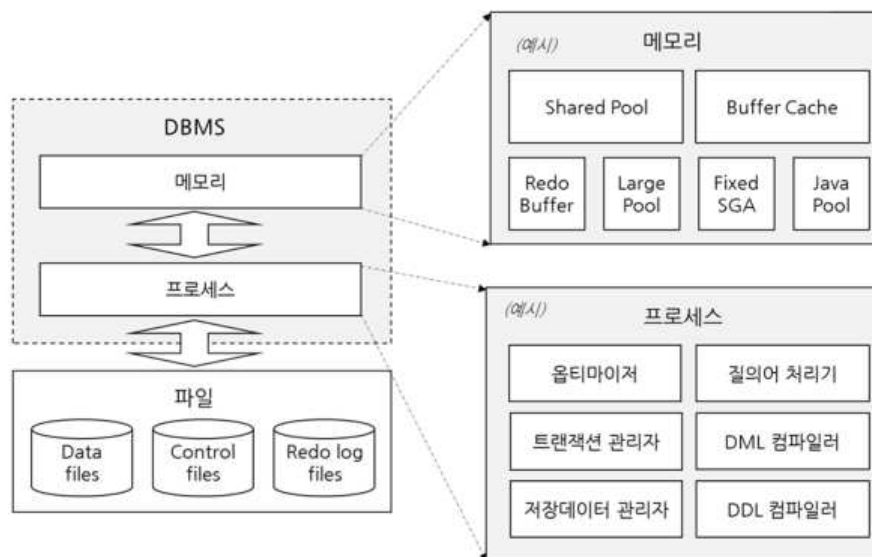
만 SQL이 직접 데이터를 조작하는 것이 아니라, SQL과 데이터베이스의 중간에 데이터베이스 관리시스템을 통해 데이터베이스를 조작하게 된다.

2) 데이터베이스 관리시스템

조직의 목적을 위해 존재하는 운영 데이터를 통합 저장하여 공동으로 사용 가능하도록 관리하는 시스템

관점	내용	비고
저장 데이터	컴퓨터를 통해 접근 가능한 저장 매체에 저장된 데이터	Stored Data
통합 데이터	중복이 최소화된 데이터	Integrated Data
공유 데이터	여러 응용 프로그램이 공동으로 사용하는 데이터	Shared Data
운영 데이터	조직의 목적을 위해 존재 가치가 확실하고 반드시 필요한 데이터	Operational Data

3) 데이터베이스 관리시스템 구조



2-1

인덱스 활용

테이블 조회 시간을 단축하기 위해 사용하는 인덱스의 개념을 이해하고, 인덱스를 생성하는 DDL(Data Definition Language) 명령문을 작성할 수 있다.

[1] 인덱스 개요

인덱스는 데이터를 빠르게 찾을 수 있는 수단으로서, 테이블에 대한 조회 속도를 높여 주는 자료구조

index_name		table_greate_men			
Index (이름)	주소	일련번호	이름	생년월일	출생지
김구	5	1	장보고	07871111	전라남도 완도군
박문수	4	2	홍길동	14431111	전라남도 장성군
윤동주	6	3	이순신	15450428	서울시 중구 인현동
이순신	3	4	박문수	16911111	경기도 평택시
장보고	1	5	김구	18760829	황해도 해주시
홍길동	2	6	윤동주	19171230	중국 연변 연정시

PK 컬럼은 PK를 생성할 때 자동으로 인덱스가 생성된다. 즉, PK컬럼은 PK를 생성할 때 자동으로

PK인덱스가 생성된다. 예를 들어 위의 그림과 같은 테이블에서 일련번호를 기본키(Primary Key)로 하는 경우, 일련번호에 대한 인덱스는 자동으로 생성되나, 생년월일이나 이름을 기준으로 하는 인덱스는 자동으로 생성되지 않는다.

```
select * from table_grate_men where 이름 = '이순신';
```

조건문 where절에서 '이름'을 비교하고 있다. 이 경우 해당 테이블의 '이름' 컬럼에 인덱스가 없는 경우, 테이블의 전체 내용을 검색(Table full scan)하게 된다. 반면, 인덱스가 생성되어 있다면 테이블의 일부분을 검색(Range scan)하여 데이터를 빠르게 찾을 수 있다. 조건절에 '='로 비교되는 컬럼을 대상으로 인덱스를 생성하면 검색 속도를 높일 수 있다. 하지만 자동으로 생성되지 않기 때문에 DB 사용자는 질의문을 분석하여 인덱스를 생성해야 한다.

[2] 인덱스 사용

분류	문법
인덱스 생성	CREATE [UNIQUE] INDEX <index_name> ON <table_name> (<column(s)>);
인덱스 삭제	DROP INDEX <index name>;
인덱스 변경	ALTER [UNIQUE] INDEX <index name> ON <table name> (<column(s)>);

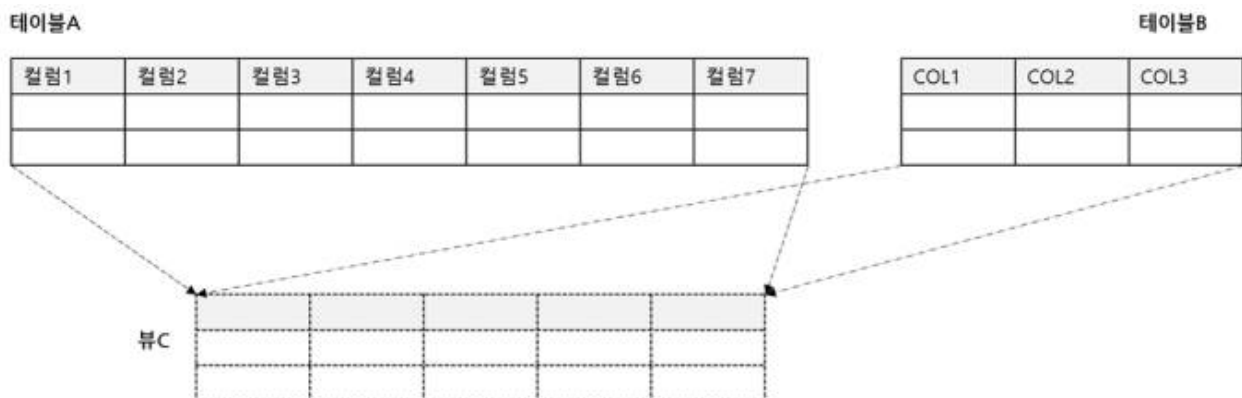
2-2

뷰 활용

먼저 생성된 테이블들을 이용하여 새로운 테이블과 뷰를 생성하는 DDL(Data Definition Language) 명령문을 작성할 수 있다.

[1] 뷰(View)의 개요

뷰는 논리 테이블로서 사용자에게(생성 관점 아닌 사용 관점에서) 테이블과 동일하다



[2] 뷰 활용 상세

1. 뷰 사용

단순한 질의어 사용

```
select * from <View Name>;
```

장/단점	뷰	내용
장점	논리적 독립성 제공	논리 테이블(테이블의 구조가 변경되어도 뷰를 사용하는 응용 프로그램은 변경하지 않아도 됨).
	사용자 데이터 관리 용이	복수 테이블에 존재하는 여러 종류의 데이터에 대해 단순한 질의어 사용이 가능함.
	데이터 보안 용이	중요 보안 데이터를 저장 중인 테이블에는 접근 불허하고, 해당 테이블의 일부 정보만을 볼 수 있는 뷰에는 접근을 허용하는 방식으로 보안 데이터에 대한 접근 제어 가능
단점	뷰 자체 인덱스 불가	인덱스는 물리적으로 저장된 데이터를 대상으로 하기에 논리적 구성인 뷰 자체는 인덱스를 가지지 못함.
	뷰 정의 변경 불가	뷰의 정의를 변경하려면 뷰를 삭제하고 재생성하여야 함.
	데이터 변경 제약 존재	뷰의 내용에 대한 삽입, 삭제, 변경 제약이 있음.

2. 뷰 생성

```
CREATE VIEW <뷰이름>(컬럼목록) AS <뷰를 통해 보여줄 데이터 조회용 쿼리문>
```

상황	뷰 생성 쿼리문
테이블A 그대로	CREATE VIEW 뷰A AS select * from 테이블A;
테이블A 일부 컬럼	CREATE VIEW 뷰X AS select 컬럼1, 컬럼2, 컬럼3 from 테이블A;
테이블A와 테이블B 조인 결과	CREATE VIEW 뷰Y AS select * from 테이블A a, 테이블 B b where a.컬럼1=b.COL1;

3. 뷰 삭제 및 변경

뷰 정의 자체를 변경하는 것은 불가능하다. 일단 뷰를 정의하면, 뷰의 물리적 내용은 뷰의 이름과 데이터를 조회하기 위한 쿼리문뿐이다. 이때 뷰의 이름이나 쿼리문을 변경하는 수단은 제공되지 않는다. 이 경우 뷰의 삭제와 재생성을 통해 뷰에 대한 정의 변경이 가능하다.

```
DROP VIEW <View Name>;
```

4. 뷰 내용 변경

뷰를 통해 접근 가능한 데이터에 대한 변경이 가능하다. 하지만 모든 경우에 데이터의 변경이 가능한 것이 아니라 일부 제약이 존재한다. 이러한 제약은 뷰 자체가 논리적 개념이기에 물리적 상황에 의존적임을 의미한다. 예를 들어 PK에 해당하는 컬럼이 뷰에 정의되어 있지 않은 경우 INSERT는 당연히 불가능하다.

5. 뷰 업데이트가 불가능한 상황

UPDATE가 불가능한 상황	내용
뷰를 구성하는 테이블의 기본키를 포함하지 않은 경우	PK, FK 등이 포함되지 않은 뷰
뷰의 컬럼이 산술식, 집계 함수, 상수로부터 유도된 경우	sum, count, avg 등이 포함된 뷰
뷰 정의에서 Group by 절이 포함된 경우	물리적으로 존재하는 레코드를 특정할 수 없음
다수 테이블로 뷰가 정의된 경우	대부분 불가능하지만 뷰 생성 조건이 엄격하다면 update가 가능할 수 있음

6. 뷰 변경 가능/불가능한 경우

경우	변경 여부
뷰가 하나의 테이블에서 정의된 경우	가능
뷰 생성에 사용된 테이블의 PK를 포함하는 경우	가능
뷰 정의에서 집계 함수로 정의된 컬럼이 있는 경우	불가능
뷰 정의에서 DISTINCT가 포함된 경우	불가능
뷰 정의에서 GROUP BY 또는 HAVING이 포함된 경우	불가능
뷰 정의에서 서브쿼리가 포함된 경우	불가능
뷰 정의에 상수, 문자열 등이 포함된 경우	불가능

2-3

다중 테이블 검색

조인, 서브쿼리, 집합 연산자를 사용하여 두 개 이상의 테이블로부터 데이터를 조회하는 DML(Data Manipulation Language) 명령문을 작성할 수 있다.

[1] 다중 테이블 검색 방법

다중 테이블 검색 방법	내용
조인	두 개의 테이블을 결합하여 데이터를 추출하는 기법
서브쿼리	SQL문 안에 포함된 SQL문 형태의 사용 기법
집합 연산	테이블을 집합 개념으로 조작하는 기법

[2] 조인

1. 조인 개념

두 개의 테이블을 결합하여 데이터를 추출하는 기법, PK, FK값을 결합하여 사용

2. 조인유형

조인 분류	내용
논리적 조인	사용자의 SQL문에 표현되는 테이블 결합 방식
물리적 조인	데이터베이스의 옵티마이저에 의해 내부적으로 발생하는 테이블 결합 방식 - Nested Loop Join - Merge Join - Hash Join

조인 유형	내용
내부 조인(INNER JOIN)	두 테이블에 공통으로 존재하는 컬럼을 이용하는 방식 (공통 컬럼 기반)
동등 조인(EQUI JOIN)	공통 존재 컬럼의 값이 같은 경우를 추출
자연 조인(NATURAL JOIN)	두 테이블의 모든 컬럼을 비교하여 같은 컬럼명을 가진 모든 컬럼 값이 같은 경우를 추출
교차 조인(CROSS JOIN)	조인 조건이 없는 모든 데이터의 조합을 추출
외부 조인(OUTER JOIN)	특정 테이블의 모든 데이터를 기준으로 다른 테이블의 정보를 추출(다른 테이블에 값이 없어도 출력됨.)
왼쪽 외부 조인(LEFT OUTER JOIN)	왼쪽 테이블의 모든 데이터와 오른쪽 테이블의 동일 데이터를 추출
오른쪽 외부 조인(RIGHT OUTER JOIN)	오른쪽 테이블의 모든 데이터와 왼쪽 테이블의 동일 데이터를 추출
완전 외부 조인(FULL OUTER JOIN)	양쪽의 모든 데이터를 추출

[3] 서브쿼리(Sub Query)

1. 개념



- SQL문 안에 포함된 또 다른 SQL문을 의미한다
- 메인쿼리와 서브쿼리 관계는 주종 관계로서, 서브쿼리에 사용되는 컬럼 정보는 메인쿼리의 컬럼 정보를 사용할 수 있으나 역으로는 성립하지 않는다.

2. 서브쿼리 유형

서브쿼리 종류	설명
비연관(Un-Correlated) 서브쿼리	서브쿼리가 메인쿼리의 컬럼을 가지고 있지 않은 형태(메인쿼리에 서브쿼리에서 실행된 결과값의 제공 용도)
연관(correlated) 서브쿼리	서브쿼리가 메인쿼리의 컬럼을 가지고 있는 형태(메인쿼리가 먼저 수행되어 얻은 데이터를 서브쿼리의 조건에 맞는지 확인하고자 할 경우에 사용)

반환되는 데이터 형태에 따른 서브쿼리의 종류는 다음과 같다.

서브쿼리 종류	설명
Single Row(단일 행) 서브쿼리	- 서브쿼리의 결과가 항상 1건 이하인 서브쿼리 - 단일 행 비교 연산자(=, <, <=, >, >=, <>)가 사용됨.
Multiple Row(다중 행) 서브쿼리	- 서브쿼리 실행 결과가 여러 건인 서브쿼리 - 다중 행 비교 연산자(IN, ALL, ANY, SOME, EXISTS)가 사용됨.
Multiple Column(다중 컬럼) 서브쿼리	- 서브쿼리 결과가 여러 컬럼으로 반환되는 서브쿼리 - 메인쿼리의 조건절에 여러 컬럼을 동시에 비교할 때, 서브쿼리와 메인쿼리에서 비교하는 컬럼 개수와 위치가 동일해야 함.

[4] 집합 연산

테이블을 집합 개념으로 보고, 두 테이블 연산에 집합 연산자를 사용하는 방식이다. 집합 연산자는 여러 질의 결과를 연결하여 하나로 결합하는 방식을 사용한다. 즉, 집합 연산자는 2개 이상의 질의 결과를 하나의 결과로 만들어 준다. 일반적으로 집합 연산자를 사용하는 상황은 서로 다른 테이블에서 유사한 형태의 결과를 반환하는 것을 하나의 결과로 합치고자 할 때와 동일 테이블에서 서로 다른 질의를 수행하여 결과를 합치고자 할 때 사용할 수 있다.

집합 연산자	설명
UNION	여러 SQL문의 결과에 대한 합집합(중복 행 제거함.)
UNION ALL	여러 SQL문의 결과에 대한 합집합(중복 행 제거하지 않음.)
INTERSECTION	여러 SQL문의 결과에 대한 교집합(중복 행 제거함.)
EXCEPT (MINUS)	앞의 SQL문의 결과와 뒤의 SQL문의 결과 사이의 차집합(중복 행 제거, 일부 제품의 경우 MINUS 사용)

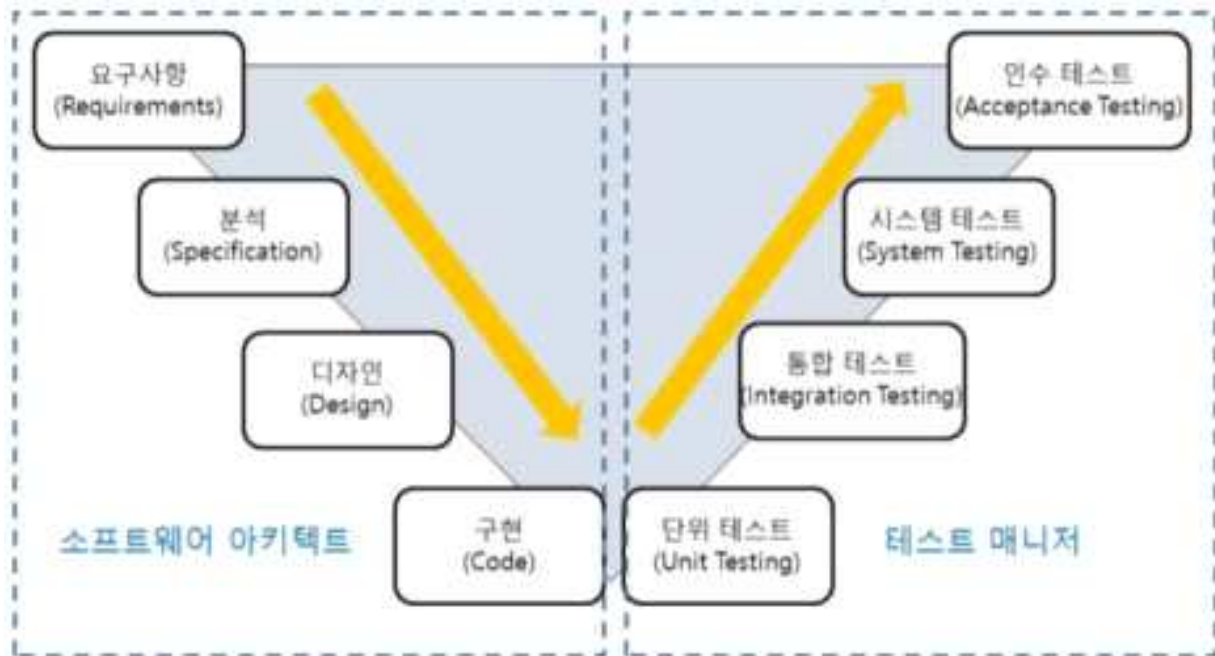
6. 애플리케이션 테스트 수행(2001020227_16v4.1)

1-1

테스트 수행

애플리케이션 테스트 계획에 따라 서버 모듈, 화면 모듈, 데이터 입출력, 인터페이스 등 기능 단위가 요구사항을 충족하는지에 대한 테스트를 수행할 수 있다.

[1] 테스트의 개요



1) 단위 테스트

- 작은 소프트웨어(컴포넌트 또는 모듈)를 테스트하는 것으로 개발자 자신에 의한 테스트
- 중요한 부분이므로 개발 도구에서 지원하지 않아도 반드시 수행해야 함.

2) 통합 테스트

- 모듈 사이의 인터페이스, 통합된 컴포넌트 간의 상호 작용 테스트
- 하나의 프로세스가 완성된 후 부분적으로 통합 테스트 수행

3) 시스템 테스트

- 통합된 단위 시스템의 기능이 시스템에서 정상적으로 수행되는지 성능 및 장애 테스트

4) 인수 테스트

- 최종 사용자와 업무에 따른 이해관계자 등이 테스트를 수행함으로써 개발된 제품에 대해 운영 여부를 결정하는 테스트로 실제 업무 적용 전에 수행

[2] 프로젝트 수행 단계에 따른 테스트의 접근 방법

1. 단위 테스트

테스트 가능한 단위로 작게 분리된 소프트웨어 내에서 결함을 찾고 기능을 검증하는 테스트 활동

1) 특징

구조적 테스트, 기능성 테스트, 리소스 관련 테스트, 강건성 테스트 등 특정 비기능성 테스트 등이 포함되어 수행됨.

컴포넌트 명세, 소프트웨어 상세 설계, 데이터 모델 명세 등을 이용하여 테스트.

2) 방법과 목적

- ① 구조 기반 : 업무 단위별 제어 흐름과 조건 결정에 따른 결과를 테스트하는 데 목적이 있음.
- ② 명세 기반 : 동등 분할과 경계 값 분석을 위하여 사용자의 입력, 출력, 내부, 이벤트 등을 확인하는 데 목적이 있음.

테스트 방법	테스트 내용	테스트 목적
구조 기반	프로그램 내부 구조 및 복잡도를 검증하는 화이트박스(White Box) 테스트	제어 흐름, 조건 결정
명세 기반	목적 및 실행 코드 기반의 실행을 통한 블랙박스(Black Box) 테스트	동등 분할, 경계 값 분석

2. 통합 테스트

통합 테스트는 컴포넌트 간 인터페이스 테스트를 하고 운영체제(OS), 파일 시스템, 하드웨어 또는 시스템 간 인터페이스와 같은 각각 다른 부분과 상호 연동이 정상적으로 작동하는지 여부를 테스트한다.

- (1) 특징 : 빅뱅 방식보다는 순차적(Incremental) 형태와 아키텍처에 대한 이해를 바탕으로 진행.
- (2) 종류 : 빅뱅, 상향, 하향, 샌드위치, Central, Collaboration, 레이어 통합 테스트 등

3. 시스템 테스트

컴퓨터 시스템을 완벽하게 검사하기 위한 목적 또는 성능 목표를 가지고 테스트한다.

(1) 특징

- 개발 프로젝트 차원에서 정의된 전체 시스템의 동작과 관련되어 있음
- 환경 제한적 장애 관련 리스크를 최소화하기 위하여 실제의 최종 사용자 환경과 유사하게 시스템 성능, 관련된 고객의 기능, 비기능적인 요구사항 등이 완벽하게 수행되는지를 테스트
- 요구사항 명세서, 비즈니스 절차, 유스케이스, 리스크 분석 결과 등을 이용함.

(2) 테스트 방법

테스트 방법	테스트 내용
기능적 요구사항	요구사항 명세서, 비즈니스 절차, 유스케이스 등 명세서 기반의 블랙박스(Black Box) 테스트
비기능적 요구사항	성능 테스트, 회복 테스트, 보안 테스트, 내부 시스템의 메뉴 구조, 웹 페이지 네비게이션 등의 구조적 요소에 대한 화이트박스(White Box) 테스트

4. 인수 테스트

시스템의 일부 또는 특정한 비기능적인 특성에 대해 인수테스트를 통해 확인

테스트 종류	테스트 내용
사용자 인수 테스트	비즈니스 사용자가 시스템 사용의 적절성 여부를 확인
운영상의 인수 테스트	시스템 관리자가 시스템 인수 시 수행하는 테스트 활동으로 백업/복원 시스템, 재난 복구, 사용자 관리, 정기 점검 등을 확인
계약 인수 테스트	계약상의 인수/검수 조건을 준수하는지 여부를 확인
규정 인수 테스트	정부 지침, 법규, 규정 등 규정에 맞게 개발하였는지 확인
알파 테스트	개발하는 조직 내 잠재 고객에 의해 테스트 수행
베타 테스트	실제 환경에서 고객에 의해 테스트 수행

[3] 테스트 기반(Test Bases)에 따른 테스트의 종류

테스트 종류	테스트 내용	세부 기법
구조 기반	소프트웨어 내부의 논리 흐름에 따른 테스트 케이스 작성 및 결함 발견 활동	구문 기반, 결정 기반, 조건 기반, 조건결정 기반, 변경 조건 기반, 멀티 조건 기반 커버리지
명세 기반	사용자의 요구사항 분석서에 주어진 명세를 빠뜨리지 않고 테스트 케이스화	동등 분할, 경계 값 분석, 결정 테이블 테스트, 상태 전이 테스트, 유스케이스 테스트
경험 기반	유사 소프트웨어나 기술에서의 테스트의 경험, 직관, 기술 능력을 바탕으로 하는 테스트 기법	탐색적 테스트, 리스크 기반 테스트

[4] 테스트 자동화 도구

테스트의 정확성을 유지하면서 시간과 비용을 줄일 수 있는 자동화 도구 필요

(1) 테스트 자동화의 개념

테스트 자동화란 사람이 하던 반복적 테스트 절차를 자동화 도구를 활용하여 준비, 구현, 수행, 분석 등을 스크립트 형태로 구현함으로써, 시간과 인력 투입의 부담을 최소화하면서 운영 중인 시스템의 모니터링 또는 UI가 없는 서비스의 경우에도 정밀한 테스트가 가능하도록 하는 것.

(2) 테스트 도구의 장점

- 테스트 데이터의 재입력과 재구성 같은 반복 작업의 자동화를 통하여 테스트 인력과 시간을 최소화
- 향상된 요구사항 정의, 성능 및 스트레스 테스트, 품질 측정을 최적화
- 빌드 확인, 회귀, 다중 플랫폼 호환성, 소프트웨어 구성, 기본 테스트 등의 향상된 테스트 품질 보장

(3) 테스트 도구의 단점

- 도입 후 테스트 도구 전문가를 양성 또는 고용이 필요하며, 초기에 프로세스 적용에 대한 시간, 비용, 노력에 대한 추가 투자가 필요.
- 비공개 상용 소프트웨어의 경우 고가이며, 인력과 교육에 대한 유지관리 비용 높음.

(4) 테스트 자동화 수행 시의 고려사항

- 테스트 절차를 고려하여 재사용 및 측정이 불가능한 테스트 프로그램은 제외해야 한다.
- 설계기준 고려하여 반복적인 빌드에서 스크립트 재사용성이 가능해야 한다.
- 도구의 한계성으로 모든 수동 테스트 과정을 자동화 할 수 있는 도구는 없다. 따라서 용도에 맞는 적절한 도구 사용이 필요하다.
- 도구 환경 설정과 도구 습득 기간을 고려하여 프로젝트의 지연을 방지해야 한다.
- 테스트 엔지니어 늦은 투입은 프로젝트의 이해 부족으로 불완전한 테스트를 초래할 수 있으므로, 적절한 투입 시기와 계획을 프로젝트 초기에 수립해야 한다.

(5) 테스트 활동에 따른 자동화 도구

테스트 활동	테스트 도구	내 용
테스트 계획	요구사항 관리	고객 요구사항 정의 및 변경 사항 관리
테스트 분석/설계	테스트케이스 생성	테스트 기법에 따른 테스트 데이터 및 케이스 작성
	커버리지 분석	대상 시스템에 대한 테스트 완료 범위의 척도
테스트 수행	테스트 자동화	기능 테스트 등 테스트 도구를 활용하여 자동화를 통한 테스트의 효율성 제고
	정적 분석	코딩 표준, 런타임 오류 등을 검증
	동적 분석	대상 시스템 시뮬레이션을 통한 오류 검출
	성능 테스트	가상 사용자를 인위적으로 생성하여 시스템 처리 능력 측정
테스트 통제	모니터링	시스템 자원(CPU, Memory 등)의 상태 확인 및 분석 지원 도구
	형상 관리	테스트 수행에 필요한 다양한 도구 및 데이터 관리
	테스트 관리	전반적인 테스트 계획 및 활동에 대한 관리
	결함 추적/관리	테스트에서 발생한 결함 관리 및 협업 지원

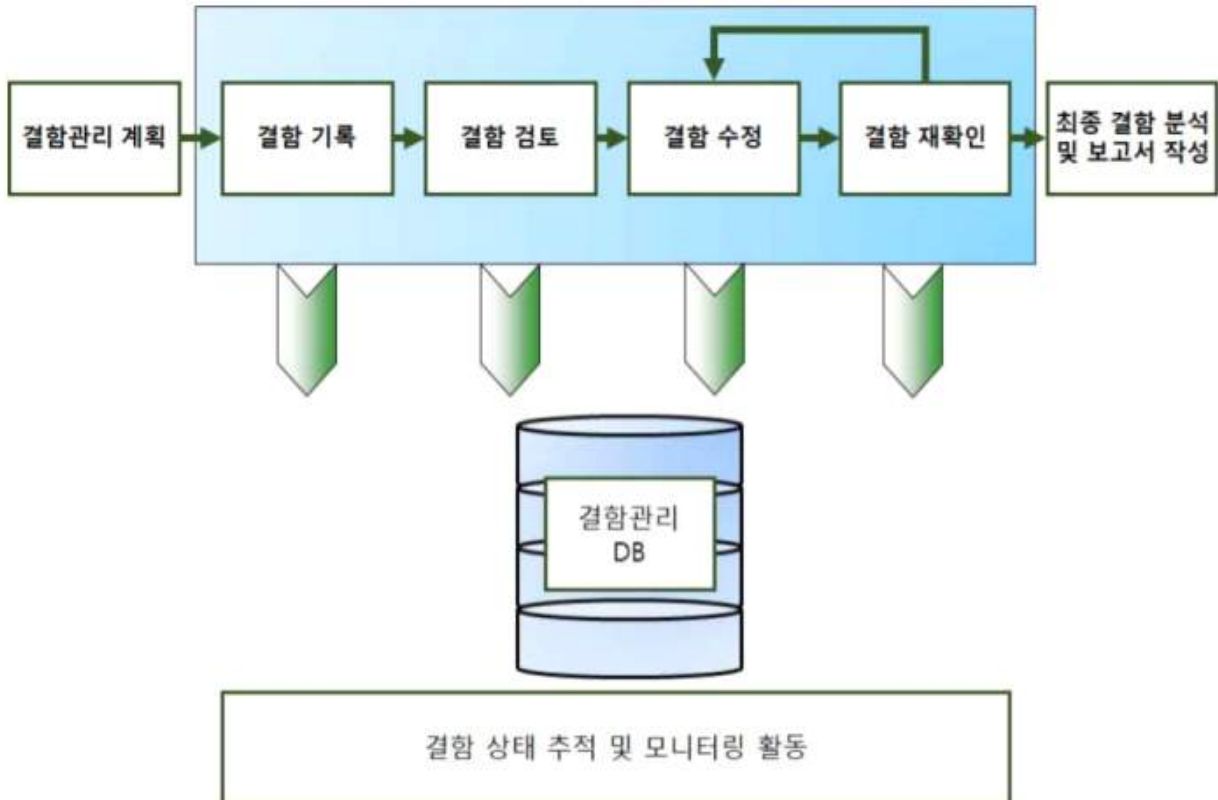
결함관리

- 애플리케이션 테스트 수행으로 발견된 결함을 유형별로 기록할 수 있다.
- 애플리케이션 테스트 수행 결과 발견된 결함에 대해서 원인을 분석하고 개선 방안을 도출할 수 있다.

[1] 결함의 정의

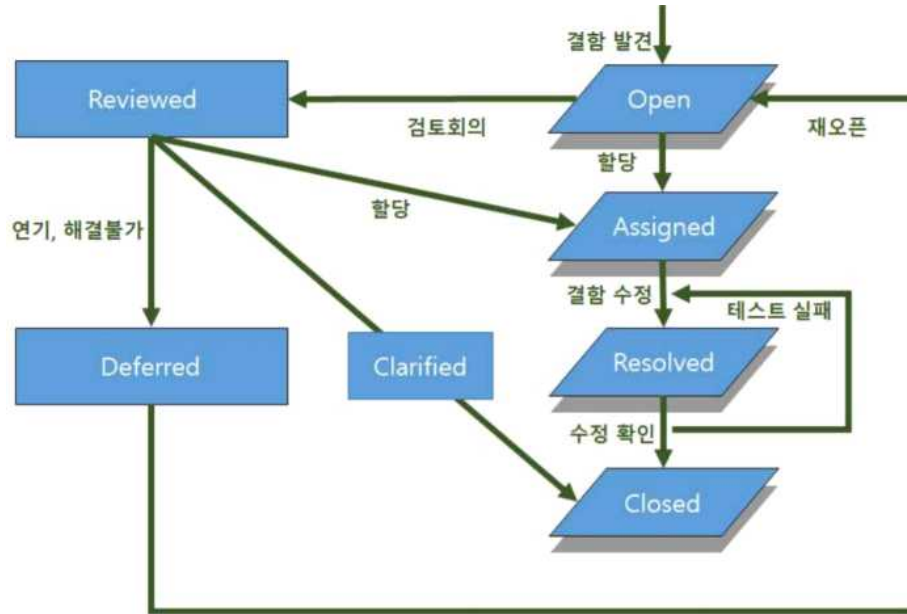
- 프로그램과 명세서 간의 차이, 업무 내용 불일치
- 기대 결과와 실제 관찰 결과 간의 차이
- 시스템이 사용자가 기대하는 타당한 기대치를 만족시키지 못할 때 변경이 필요한 모든 것

[2] 결함관리 프로세스



1. 결함관리 계획 : 전체 프로세스에서 결함관리에 대한 일정, 인력, 업무 프로세스를 확보하여 계획을 수립하는 것이다.
2. 결함 기록 : 테스터는 발견된 결함에 대한 정보를 결함관리 DB에 기록한다.
3. 결함 검토 : 등록된 결함에 있어서 주요 내용을 검토하고, 결함을 수정할 개발자에게 전달한다.
4. 결함 수정 : 개발자는 할당된 결함의 프로그램을 수정한다.
5. 결함 재확인 : 테스터는 개발자가 수정한 내용을 확인하고 다시 테스트를 수행한다.
6. 결함 상태 추적 및 모니터링 활동 : 결함관리 팀장은 결함관리 데이터베이스를 이용하여 대시보드 또는 게시판 형태의 서비스를 제공한다.
7. 최종 결함 분석 및 보고서 작성 : 발견된 결함에 대한 내용과 이해관계자들의 의견이 반영된 보고서를 작성하고 결함관리를 종료한다.

[3] 결함의 상태 및 추적



1. 결함 등록(Open)
테스터와 품질 관리(QA) 담당자에 의해 결함이 처음 발견되어 등록되었지만, 아직 분석이 되지 않은 상태.
2. 결함 검토(Reviewed)
등록된 결함을 담당 모듈 개발자, 테스터, 프로그램 리더, 품질 관리(QA) 담당자와 검토하는 상태.
3. 결함 할당(Assigned)
결함의 영향 분석 및 수정을 위해 개발자와 문제 해결 담당자에게 할당된 상태.
4. 결함 수정(Resolved)
개발자에 의해 결함의 수정이 완료된 상태.
5. 결함 조치 보류(Deferred)
수정이 필요한 결함이지만 현재 수정이 불가능해서 연기된 상태로서 우선순위, 일정 등을 고려하여 재오픈을 준비하는 상태.
6. 결함 종료(Closed)
발견된 결함이 해결되고 테스터와 품질 관리(QA) 담당자에 의해 종료 승인을 한 상태.
7. 결함 해제(Clarified)
테스터, 프로그램 리더, 품질 관리(QA) 담당자가 결함을 검토한 결과, 결함이 아니라고 판명된 경우.

[4] 결함 분류

1. 시스템 결함
비정상적인 종료/중단, 응답 시간 지연, 데이터베이스 에러 등 주로 애플리케이션 환경과 데이터베이스 처리에서 발생하는 결함을 말한다.
 - (1) 비정상적인 종료/중단
특정 기능 실행 시 응용 프로그램의 작동 정지, 종료, 시스템 다운이 되는 경우.
 - (2) 응답 시간 지연
응용 프로그램 작동 후 조회 또는 보고서 출력 시 지연되는 경우와 메모리 부족, 하드웨어와 소프트웨어의 비일관성으로 발생하는 경우.
 - (3) 데이터베이스 에러
응용 프로그램 작동 후 사용자 데이터의 등록, 수정, 삭제, 조회가 정상적으로 작동하지 않는 경우.

2. 기능 결함

사용자의 요구사항 미반영/불일치, 부정확한 비즈니스 프로세스, 스크립트 에러, 타 시스템 연동 시 오류 등 기획, 설계, 업무 시나리오 단계에서 발생한 결함을 말한다.

(1) 요구사항 미반영/불일치

요구사항에 명시된 기능이 응용 프로그램에 구현되지 않은 경우와, 다르게 구현되어 작동하는 경우.

(2) 부정확한 비즈니스 프로세스

기능 자체는 수행되나 내부 프로세스 로직의 문제로 부정확한 결과를 내는 경우.

(3) 스크립트 에러

특정 기능 실행 시 웹 브라우저에서 스크립트 오류가 발생하는 경우.

(4) 타 시스템 연동 시 오류

기존 시스템과의 연동을 통해 데이터를 주고받는 과정에서 오류가 발생하는 경우.

3. GUI 결함

GUI 결함은 응용 프로그램의 UI 비일관성, 부정확한 커서/메시지, 데이터 타입의 표시 오류 등으로 사용자 화면 설계에서 발생한 결함을 말한다.

(1) 응용 프로그램 UI 비일관성

프로젝트에서 정의한 UI 표준과 상이하게 구현된 경우.

(2) 부정확한 커서/메시지

커서의 위치가 입력 대상의 첫 번째 필드에 위치해 있지 않거나, 탭 시퀀스가 순차적으로 동작하지 않는 경우, 각 기능에서 제공하는 메시지 내용이 부정확한 내용을 보여주는 경우.

(3) 데이터 타입의 표시 오류

입력 필드에 지정된 형식과 다르게 입력해도 저장이 되는 경우와 입력 필드에 유효하지 않은 데이터(Invalid Data)를 입력했을 때 오류가 나는 경우.

4. 문서 결함

기획자, 사용자, 개발자 간의 의사소통과 기록이 원활하지 않은 경우에 발생하는 결함으로 사용자의 온라인/오프라인 매뉴얼의 불일치, 요구사항 분석서와 기능 요구사항의 불일치로 인한 불완전한 상태의 문서의 경우를 말한다.

[5] 결함 심각도

결함 심각도는 여러 개의 결함 중 전체 시스템에 결함이 미치는 영향을 레벨별로 나타내고, 우선순위를 High, Medium, Low 등으로 정하는 것을 말한다.

1. High

시스템이 중단(또는 다운)되어 더 이상 프로세스를 진행할 수 없게 만드는 결함으로 시스템의 핵심 요구사항 미 구현, 시스템 다운, 장시간 시스템 응답 지연, 시스템 복구 후 데이터 왜곡 등의 경우를 말한다.

2. Medium

시스템의 흐름에 영향을 미치는 결함으로 부정확한 기능, 부정확한 업무 프로세스, 데이터 필드 형식의 오류, 데이터베이스 에러, 보안 관련 오류 등의 경우를 말한다.

3. Low

시스템의 흐름에는 영향을 미치지 않는 결함이나 상황에 맞지 않는 용도와 화면 구성(Configuration) 결함으로, 부정확한 GUI 및 메시지, 에러 시 메시지 미출력, 화면상의 문법/철자 오류 등을 말한다.

조치 우선순위 결정

애플리케이션 테스트 수행 결과에서 발견된 결함을 식별하고 조치에 대한 우선순위를 결정하고 적용할 수 있다.

[1] 소프트웨어 테스트 기법

1. 단위 테스트 기법

JUnit 테스트, Mock 테스트

2. 통합 테스트 기법

(1) 테스트 설계 기법

전체 시스템이 통합 완료될 때까지 단위 시스템 간의 연계성 및 기능 요구사항들을 확인하고, 하드웨어와 소프트웨어 구성 요소 간의 상호 작용을 테스트.

업무 간 연계성과 상호 운영성 중심의 테스트 수행

(2) 테스트 설계 방법

분류	수행 내용
보다 작은 케이스	동등 클래스(Equivalence Class) 예) 한 자리 정수를 더하는 프로그램인 경우 1+1, 1+2...9+9까지 81가지 케이스는 모두 동등한 경우이다. 따라서 이런 경우를 모두 모아서 몇 가지 케이스만 테스트 하면 되는 경우이다.
보다 많은 버그 찾기	경계 테스트(Boundary Test) 예) 동등 클래스 중 대표 클래스를 뽑을 때 가장자리, 즉 경계값을 뽑는 경우
테스트 대상을 빠짐없이 실행하기	모델 기초(Model-based) 테스트 설계 예) 상태 전이(State-transition) 테스트 모델을 UML의 State Chart Diagram으로 표현하여 노드와 링크를 모두 대상으로 하여 테스트한다. 또는 모든 테스트 요소를 매트릭스 형태로 나열하여 테스트한다.

3. 시스템 테스트 기법

시스템 테스트 업무 진행 전체를 총괄할 수 있도록 절차 및 각 프로세스별 세부 업무를 알아야 하고 결과에 대한 분석 및 해결 방안을 제시할 수 있어야 한다.

(1) 부하 및 성능 테스트

동시접속으로 시스템에 많은 요청(업데이트, 조회 등)이 발생될 때 어떻게 가동되는지 확인하는 테스트로서, 성능 요구사항에 정의된 임계점을 넘어 부하가 계속 증가하는 상황에서 시스템이 정상적인 수준의 반응을 보이는지를 확인해야 한다.

분류	기본 공식 내용
동시 이용자 수 (Concurrent User)	TPS(Throughput)*호출간격(응답시간(Sec)+대기시간(Sec))
동시 단말 이용자 (Concurrent User)	PC 앞에 앉아서 시스템을 이용하는 이용자로서, Active User와 (가상적으로) In-active User의 합으로 정의한다.
액티브 이용자 (Active User)	동시에 같은 서비스나 업무를 실행하고 나서 응답을 기다리고 있는 이용자
처리량 (Throughput)	단위 시간당 처리하는 건수. 단위는 tps(transaction per second), tpm, tph를 사용하고, 관점에 따라 두 가지 유형이 있으며, 요청을 받는 입장에서 단위 시간당 요청 건수(Arrival rate)와 단위 시간당 처리 건수(Service Rate)로 구분되어 표현한다.
대기 시간 (Thinktime)	응답을 받은 직후부터 다음 명령 또는 호출할 때까지 사용자가 대기하는 시간

(2) 장애 복구 테스트

- 각종 장애(하드웨어 장애, 네트워크, 정전, 운영 오류 등) 상황에서 복구 및 재가동을 테스트한다.
- 애플리케이션의 장애 복구 테스트는 다양한 관점에서 테스트할 수 있으나, 서비스의 지속적인 유지라는 측면에서 발생할 수 있는 장애를 목록으로 작성하고 장애 상황을 발생시킨다.
 - 시스템(하드웨어, 네트워크, 시스템설비 등) 장애와 애플리케이션 장애로 나눌 때, 시스템 장애는 인프라 아키텍처 팀 등에서 별도로 담당하여 진행하도록 하고, 소프트웨어 아키텍트는 애플리케이션 장애 관점에서 장애 복구 테스트를 수행하도록 한다.

(3) 보안 테스트

보안성 검증은 애플리케이션의 기능적 요구사항과는 별도로 해킹이나 침투에 대비할 수 있는 최신 기술 트렌드가 반영된 검증이 필요하며, 보안 검증이 완료되면 결과를 바탕으로 취약점 분석을 한 후 개선 방안을 수립하고 반드시 이를 시스템에 반영하도록 계획을 수립해야 한다.

4. 인수 테스트 기법

최종 사용자가 요구한 기능이 제대로 반영되었는지, 인수 조건에 만족하는지를 테스트하는 기법이다. 요구 기능 만족 여부, 사용 편리성에 대하여 실제 운영 환경에서 실행되며 고객이 주도하는 테스트이다.

[2] 결함관리의 이해

1. 결함 관련 용어

(1) 에러(Error)

소프트웨어 개발 또는 유지 보수 수행 중에 발생한 부정확한 결과로, 개발자의 실수로 발생한 오타, 개발 명세서의 잘못된 이해, 서버루틴의 기능 오해 등이 있다.

(2) 오류(Fault)

프로그램 코드 상에 존재하는 것으로 비정상적인 프로그램과 정상적인 프로그램 버전 간의 차이로 인하여 발생되며, 잘못된 연산자가 사용된 경우에 프로그램이 서버루틴으로부터의 에러 리턴을 점검하는 코드가 누락된 것을 말한다.

(3) 실패(Failure)

정상적인 프로그램과 비정상적인 프로그램의 실행 결과의 차이를 의미하며, 프로그램 실행 중에 프로그램의 실제 실행 결과를 개발 명세서에 정의된 예상 결과와 비교함으로써 발견한다.

(4) 결함(Defect)

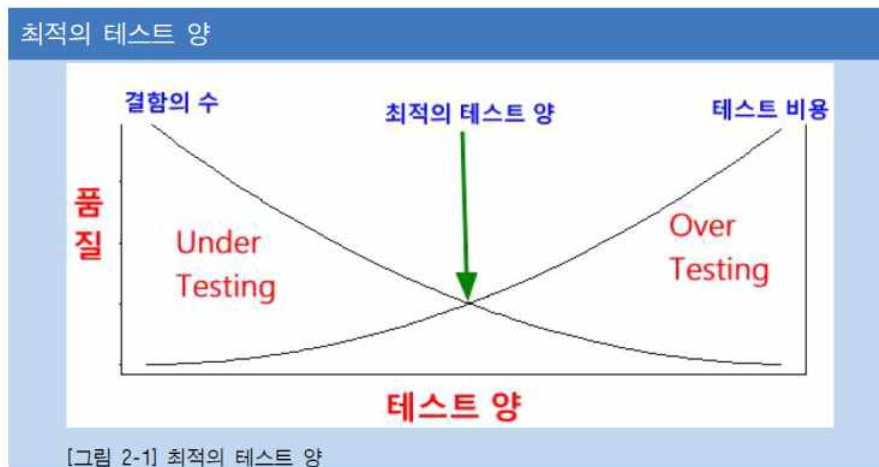
버그, 에러, 오류, 실패, 프로그램 실행에 대한 문제점, 프로그램 개선 사항등의 전체를 포괄하는 용어이다.

2. 결함의 판단 기준

- (1) 기능 명세서에 가능하다고 명시된 동작을 수행하지 않는 경우
- (2) 기능 명세서에 불가능하다고 명시된 동작을 수행하는 경우
- (3) 기능 명세서에 명시되어 있지 않은 동작을 수행하는 경우
- (4) 기능 명세서에 명시되어 있지 않지만 수행해야 할 동작을 수행하지 않는 경우
- (5) 테스터의 시각에서 볼 때 문제가 있다고 판단되는 경우

[3] 테스트 격언

1. 소프트웨어를 완벽하게 테스트하는 것은 불가능하다.



2. 소프트웨어 테스트는 위험을 수반하는 훈련이다.
3. 테스트 작업으로 결함이 존재하지 않는다는 사실을 입증할 수 없다.
4. 발견한 모든 결함을 수정할 수는 없다.

[4] 소프트웨어 테스터의 능력

1. 소프트웨어 테스터의 역할

- (1) 결함을 발견하는 것이다.
- (2) 결함을 가능한 한 빨리 발견하는 것이다.
- (3) 가능한 한 결함을 빨리 발견하여 결함을 수정 보완되었는지 확인하는 것이다.

2. 소프트웨어 테스터의 능력

- (1) 탐구심과 문제 해결력을 갖추어야 한다.
- (2) 때로는 창의력을 발휘하여 결함을 찾기 위해 새로운 접근력도 사용해야 한다.
- (3) 완벽함을 추구하지만, 가능한 한도 내에서 적당한 수준의 완벽성을 추구한다.
- (4) 테스터는 항상 나쁜 소식을 전하는 사람이다. 개발자에게 그들의 작품에 결함이 있다고 말하려면 재치와 능숙함, 그리고 설득력이 필요하다.

결함 조치 관리

- 결함이 발생한 소스를 분석하고 기존에 구현된 로직과의 연관성을 고려하여 부작용이 최소화되도록 결함을 제거할 수 있다.
- 애플리케이션 테스트 결과 결함 조치로 변경되는 소스의 버전을 관리하고 결함 조치 결과에 대한 이력을 관리할 수 있다.

[1] 프로그램 코드 검토 기법

1. 소프트웨어 인스펙션(Software Inspection)의 개요

※ inspection : 검사, 점검

Code Inspection은 개발팀에서 작성한 개발소스 코드를 분석하여 개발 표준에 위배되었거나 잘못 작성된 부분을 수정하는 작업을 말한다. 잘못된 부분이란, 코드가 중복되거나 작성규칙에 맞지 않거나 잘못 구현된 부분들을 말한다.

Code Inspection을 통해 소프트웨어의 품질을 높일 수 있고 다른 사람이 구현한 코드를 통해 여러 가지를 배울 수도 있다. 어떻게 보면 시간이 많이 걸리는 작업이기도 하나 Code Inspection은 개발 단계에서 발생할 수 있는 여러 시행 착오를 줄일 수 있는 중요한 과정중의 하나이다

- (1) 코드 인스펙션외에도 설계 및 설계 산출물까지 포괄하여 소프트웨어 인스펙션으로 부르기도 한다.
- (2) 코드 인스펙션은 매우 효과적인 테스트 방법으로서 어떠한 다른 테스트 방법으로 대체할 수 없다. 이는 상당한 시간이 필요한 작업이며, 통계에 따르면 인스펙션을 적절히 잘 수행하기만 하면 포함된 에러의 90%까지 찾아낼 수 있다.
- (3) 코드 인스펙션, 워크스루(walkthrough)와 같이 몇 시간 동안 수행되는 단위 미팅과는 구별되어야 한다. 적절한 코드 인스펙션은 여러 날이 필요하고 도구의 도움이 있어야 한다.

※ 워크스루에서는 코드를 작성한 프로그래머가 5명 내외의 프로그래머 또는 테스터 그룹에게 격식을 갖춰 발표한다. 검토자들은 검토 회의 전에 코드 사본을 전달받고 미리 분석하여 질의 사항과 코멘트를 미리 작성할 수 있도록 한다. 검토자 그룹에 적어도 한 명의 고참 프로그래머를 포함시키는 것이 중요하다. 발표자는 코드를 한 줄씩 또는 함수 단위로 읽어나가고 해당 부분이 무엇을 의미하는지 설명한다. 검토자들은 의문 사항은 무엇이든 질의하며 동료 검토에 비해 다수의 인원이 참여하는 형태이기 때문에 검토 준비와 규칙 준수는 매우 중요하다. 발표자는 검토 결과를 작성하고 발견된 버그에 대한 처리 계획을 명기하는 것이 중요하다.



- (4) 적절한 인스펙션은 소프트웨어 개발의 전체 수명 주기에 걸친 리소스 절감과 그에 따른 비용 감소 그리고 산출물의 품질을 향상시키는 것으로 나타난다.
- (5) 인스펙션을 해야 하는 비즈니스적인 이유는 아래와 같다.
 - (가) 결함을 빨리 찾을수록 수정(fix) 비용이 적게 든다.
 - (나) 인스펙션의 데이터를 통해 업무에 집중할 수 있다.
 - (다) 인스펙션을 함으로써 교차 교육(Cross-training)을 돕는다.
 - (라) 제품의 "re-engineering"이 가능한 영역을 식별하도록 돕는다.
 - (마) 소프트웨어를 개발하고 유지하는 데 적은 비용이 든다.
 - (바) 스케줄에 긍정적인 효과를 준다.
 - (사) 품질을 향상시킨다.

2. 인스펙션 중점 항목

프로젝트에서 소프트웨어 인스펙션 업무는 품질보증(QA)이 주관하여 담당 또는 부서에서 실시하나, 아키텍트가 각 단계에서 원활한 업무 수행을 위해 지원을 하여야 한다. 소프트웨어 인스펙션 중 특히 코드 인스펙션과 관련하여, 아키텍트는 코드 인스펙션의 프로세스 전반과 각 단계별 수행 업무 등을 전체적으로 이해할 필요가 있다. 아래와 같이 검토 절차에서 아키텍트는 Overview와 Rework가 잘 수행될 수 있도록 지원하는 것이 중요하다.

- (1) 자동 코드 인스펙션을 위한 환경 지원, 계획 수립 지원 활동
- (2) 체크리스트 정합성 검토 지원 활동
- (3) 인스펙션 결과 리뷰 참석
- (4) 발견된 결함을 수정하기 위한 개발자 리딩 지원 활동

3. 코드 인스펙션의 프로세스와 수행 내용

(1) 코드 인스펙션 프로세스

구분	수행 단계	주요 내용
자동 수행	1. 범위 계획 (Capacity Plan)	얼마만큼 인스펙션할 것인가? 인스펙션이 범위와 범위 선정 기준을 결정함
	2. 시작 (Overview)	자동 인스펙션 수행
준비 단계	3. 준비 (Preparation)	계획서 작성, 체크리스트 작성, 계획 공지, 대상 산출물 준비
이행 단계	4. 인스펙션 회의 (Inspection Meeting)	사전 검토 실시, 미팅 실시
시정 조치	5. 재작업 (Rework)	개발 원작자가 직접 작업함
	6. 후속처리 (Follow-up)	결과 분석서 작성 및 보고

(2) 코드 인스펙션 태스크별 수행 내용

구분	태스크	수행 내용	산출물
자동 수행	1. 자동 인스펙션 수행	- 전수검사, Quality Metrix, 결함 분석	코드 인스펙션 결과
준비 단계	2. 계획서 작성	- 일정 및 관련자, 대상 산출물 및 준비물 정의	인스펙션 계획서
	3. 체크리스트 작성	- 표준 체크리스트를 테일러링하여 인스펙션 체크리스트 작성	인스펙션 체크리스트
	4. 계획 공지	- 메일이나 공지를 통해 관련자에게 사전 공지	
	5. 대상 산출물 준비	- 산출물 작성자가 인스펙션 시 필요한 자료를 준비	
	6. 착수 회의 실시	- 진행자는 참여자에게 검토 주안점, 검토 방법, 역할 등을 교육 - 작성자는 대상 산출물 및 참조 자료에 대한 개요 소개	
이행 단계	7. 사전 검토 실시	- 산출물, 체크리스트, 사전검토서 양식 배포 - 참여자별로 자료를 개별 검토하여 발견된 부적합 사항 기록	인스펙션 결과서
	8. 미팅 실시	- 사전 검토에서 발견한 부적합 사항 검증 - 미팅에서 발견된 부적합 사항 추가 기재 - 부적합 사항 목록 정리 또는 조치 계획 수립	인스펙션 결과서
	9. 결과 정리	- 진행자는 최종 확정된 결함 내용을 인스펙션 결과서에 정리한 후 작성자에게 배포	

시정 조치	10. 보완 작업 실시	- 작성자는 각 결함에 대한 보완 작업을 실시	
	11. 시정 조치 결과 확인	- 진행자는 보완 완료 여부를 확인 (필요시 재검토 실시)	
	12. 결과 보고	- 진행자는 결과서를 작성하여 관련자에게 보고	인스펙션 결과서

(3) 코드 인스펙션 테스트별 수행 내용

(가)자동 코드 인스펙션	(나) 수동 코드 인스펙션
전체 개발된 프로그램을 대상으로 자동 인스펙션을 수행한다.	1) 자동 코드 인스펙션 코드 중 에러가 많은 경우 2) 업무 중에 복잡한 처리 로직이 있는 경우 3) 처음 투입되는 개발자의 산출물

4. 설계 인스펙션

설계 인스펙션을 위한 체크리스트
<ol style="list-style-type: none"> 1. 설계 내역들은 요구사항들로 추적 가능해야 한다. 2. 설계의 외형 부분은 고객의 요구사항 분석과 연결되어야 하고 고객 프로파일 정보를 포함해야 한다. 3. 모든 에러 메시지가 정의되어야 한다. 4. 설계 내역은 명확하게 정의되어 오해가 생기지 않도록 한다. 5. 하나 이상의 의미를 가진 용어들은 의도했던 의미가 무엇인지 분명하게 표현한다. 6. 모든 요구사항의 제약조건들이 설계에 반영되어 있어야 한다. 7. 모든 의존성이 명확히 표현되어야 한다. 8. 설치 및 데이터 이동 요구사항이 설계에 반영되어 있어야 한다. 9. (요구사항에 명시된) 성능 기준이 정제되어야 하고, 설계 내역이 그것을 만족하는지 보여야 한다. 10. 모든 설계는 최소 하나 이상의 테스트 활동에서 검증되어야 한다. 11. 에러 복구 및 백업 요구사항이 모두 설계 내역에 반영되어 있어야 한다. 12. 모든 에러 조건이 정의되어야 한다. 13. 컴포넌트와 모듈의 구조가 명확히 정의되어야 한다. 14. 모듈과 컴포넌트의 명명법 표준이 준수되어야 한다. 15. 설계에 나타난 모든 알고리즘을 명확히 표현하여 코드를 작성할 때 혼란스럽지 않게 해야 한다. 16. 산업 표준 알고리즘을 사용하는 경우에, 의도했던 알고리즘을 정확히 반영하게 해야 하며, 어떤 알고리즘의 제약사항도 설계 요구사항을 만족시켜야 한다. 17. 안전이나 보안 등의 산업계 요구사항이 설계에 반영되어야 한다. 18. 경계 값에 관련된 조건들을 식별하고, 관련된 로직을 분명하고 완전하게 정의해서 요구사항과 매핑시켜야 한다.

5. 인스펙션과 워크스루의 차이점

인스펙션은 워크스루와 달리 체크리스트를 기반으로 검토하며 발견된 모든 결함을 제거해야 한다는 특징이 있다. 완성도가 기준 이상일 때 수행함으로써 모든 결함을 없애는 데 주요 목적이 있다.

※ 인스펙션과 워크스루 비교

구분	인스펙션	워크스루
목적	결함 파악 및 제거	산출물 평가 및 개선
수행 조건	완성도가 기준 이상일 때	팀이나 관리자의 필요 시
결함 수정 여부	모든 결함은 제거되어야 함.	저자 결정
변경 사항 검증	진행자가 재작업 결과 확인	저자 결정
검토자 인원	3 ~ 6 명	2 ~ 7 명
참여자	동료	기술 전문가 및 동료
검토 인도자	교육받은 진행자	저자
검토 준비 여부	체크리스트를 이용한 검토	일반적으로 검토 준비하지 않음
검토 분량	상대적으로 적음	상대적으로 적음

검토 속도	상대적으로 느림	빠름
발표자	산출물에 의존도가 높은 사람	저자
자료 수집 여부	모든 검토자들이 기록함	하지 않음
보고서	결함 리스트 및 측정 지표	워크스루 보고서
데이터 측정 여부	필수	권장사항
체크리스트 사용 여부	사용함	사용하지 않음

[2] 형상 관리 및 구성 요소

1. 소프트웨어 형상 관리의 정의

원래 형상 관리라는 말은 소프트웨어 개발에서 사용하기 이전에 하드웨어에서 출발한 개념이다. 자동차나 비행기와 같은 제품은 수만에서 수십만 개의 부품으로 이루어져 있다. 이러한 작은 단위의 부품들을 형상 관리에서는 형상 항목(configuration item)이라 한다. 그런데 자동차나 비행기의 조립 과정에서 부품을 변경하거나 새로 개발된 부품을 장착하여 사용할 경우에 기존 부품들과의 인터페이스에 문제가 없어야 시스템이 안정적으로 작동할 수 있다.

만약 어느 한 부품이 관리되지 않고 규격이 바뀌었다면 엄청난 사고로 이어질 수 있다. 따라서 형상 관리는 바로 이러한 특정 항목의 변화에 대해 관리하면서 시스템의 통합과 일치를 보장하는 것이다.

이처럼 형상 관리는 하드웨어에서 출발했지만, 소프트웨어 개발에서 형상 관리의 중요성도 매우 크다고 생각한다. 소프트웨어 개발이라는 게 한번 시작해서 변경 없이 끝까지 가는 경우는 없기 때문이다. 대부분 사용자의 요구가 추가되는 등 지속적인 변경이 생긴다. 따라서 변경되는 것들을 체계적으로 정리하여 관리하지 않으면 어떤 개발자는 변경 전의 모듈로 개발하는 경우도 발생해 원하는 결과를 얻지 못하게 된다. 하드웨어에서 시작된 형상 관리를 소프트웨어 개발에 사용하면서 용어도 소프트웨어 형상 관리(SCM: Software Configuration Management)라고 차별화하였다.

[네이버 지식백과] 형상 관리의 개념과 절차 (쉽게 배우는 소프트웨어 공학, 2015. 11. 30, 한빛아카데미(주))

- (1) 소프트웨어 프로세스의 모든 출력물 정보
- (2) 컴퓨터 프로그램, 컴퓨터 프로그램 설명 문서, 데이터 등
- (3) 소프트웨어 프로세스 전반에 걸쳐 소프트웨어 형상의 변경 요인에 대해 소프트웨어 형상을 보호하는 활동

※ 소프트웨어 형상 관리와 소프트웨어 지원 비교 예시

소프트웨어 형상 관리	소프트웨어 지원
소프트웨어 엔지니어링 프로젝트 개시에서 소프트웨어 소멸 시점까지의 활동	소프트웨어가 고객에게 인도되고 운영되는 시점에 발생하는 소프트웨어 엔지니어링 활동

2. 기준선과 소프트웨어 형상 항목

(1) 기준선(Baseline)

변경을 통제하게 도와주는 기준선은 정식으로 검토 및 합의된 명세서나 제품 개발의 바탕으로서, 정식의 변경 통제 절차를 통해서만 변경 가능하다.

(2) 소프트웨어 형상 항목(SCI, Software Configuration Item)

소프트웨어 형상과 개발 도구의 합성으로서, SCI는 개발 단계별로 기준선을 기준으로 형상 항목을 관리한다.

개발 단계	기준선	소프트웨어 형상 항목
계획	사용자 요구사항	시스템 명세서, 개발계획서, 구성관리계획서, 품질평가계획서, 개발표준 및 절차 매뉴얼
요구분석	사용자 요구 기능이 하위 시스템 간에 어떻게 분배되는가 여부	자료흐름도, 자료사전, 자료흐름도 명세서
설계	개발 전 설계 명세	입출력명세서, 화면설계서, 초기 사용자 매뉴얼, 초기 시스템 매뉴얼, 자료 구조도, 시스템 구조도
구현	시험 계획서	원시코드, 목적코드, 실행코드, 단위시험 보고서
시스템 통합 및 시험	제품	통합시험보고서, 기능/성능/과부하 시험 보고서, 인증시험 보고서
설치 및 운영	운영	목적/실행코드, 운영자 매뉴얼, 사용자 매뉴얼

3. 형상 관리의 주요 활동

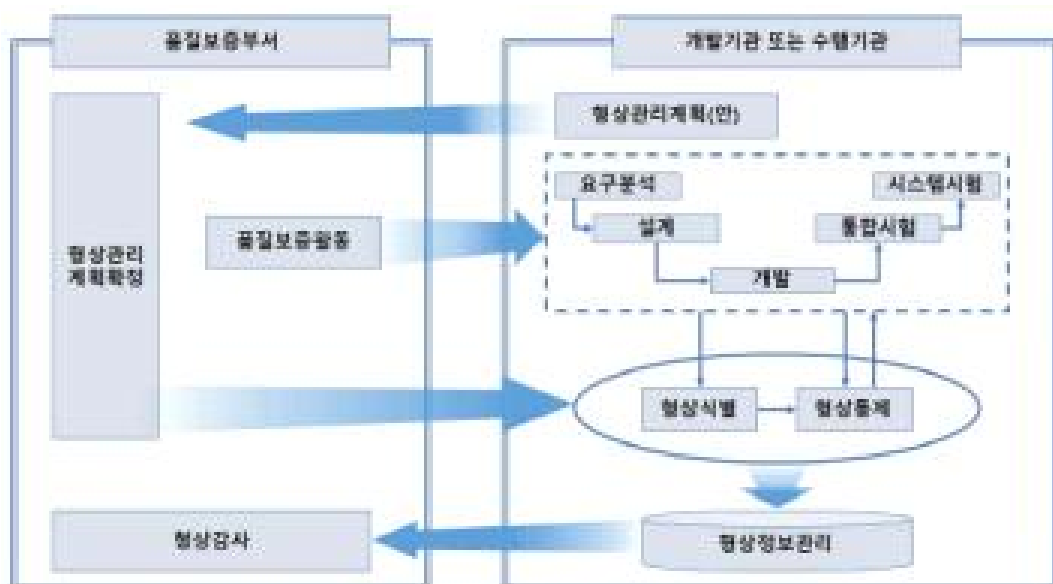
(1) 형상 관리 기능

단계	설명
형상 식별	형상 관리 대상을 구분하고 관리 목록 번호 부여
버전 관리	진화 그래프 등을 통해 SCIdml 버전 부여/갱신
변경 통제	SCI에 대한 접근 및 동기화 제어
형상 감사	SCI 무결성을 평가하여 공식적으로 승인
상태 보고	개발자와 유지 보수자에게 변경 사항을 공지

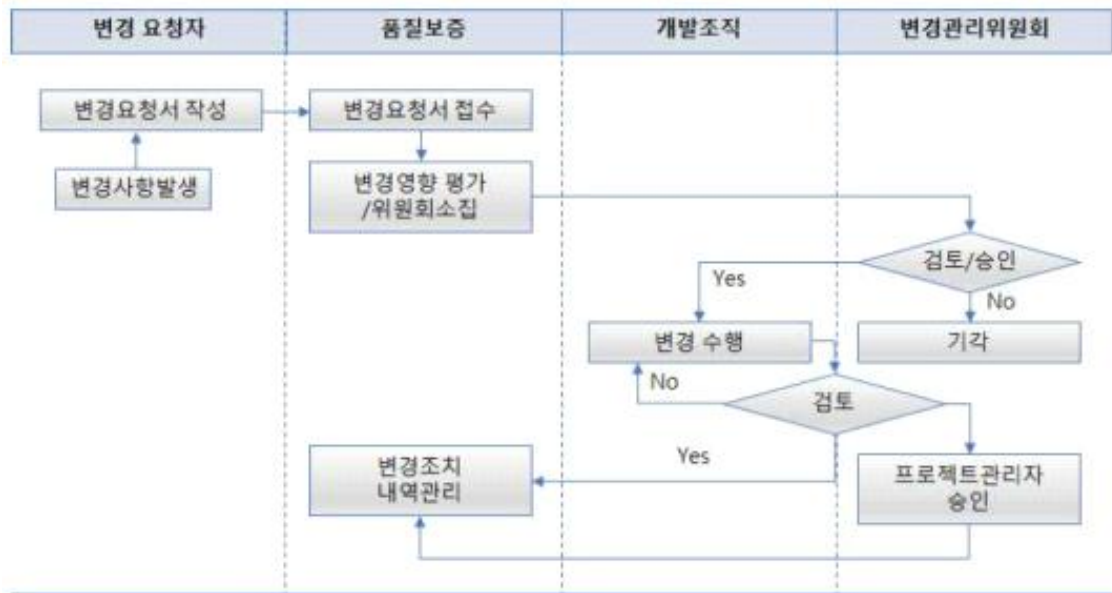
(2) 형상 식별

식별자 항목	내용
이름(name)	객체 명
서술(description)	SCI 타입, 프로젝트 식별자, 버전 정보
자원(list of resource)	제공/처리/참조/요구되는 객체
실현(realization)	기본 객체인 경우 : 단위 텍스트에 대한 포인터 실현(realization)

(3) 버전 관리



4. 변경 통제에 대한 업무별 활동



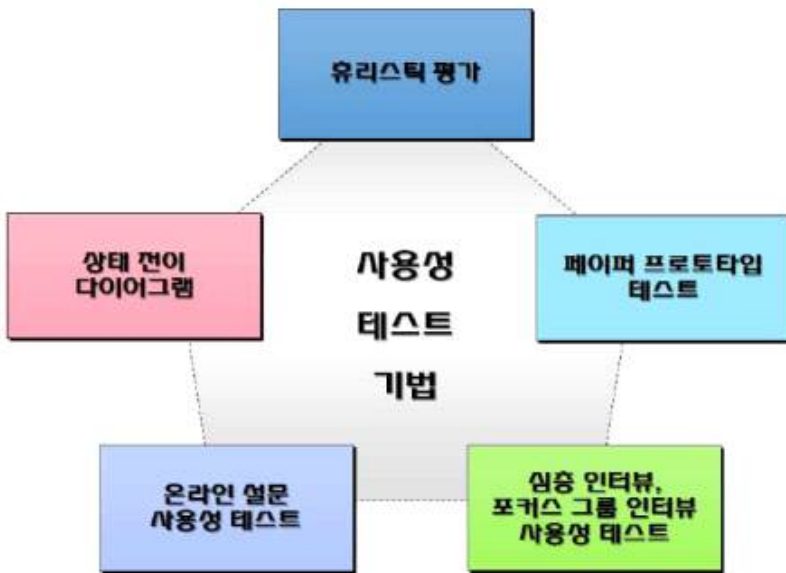
- 최근 오픈소스 형상 관리 도구로는 CVS(Concurrent Versions System), SVN(Apache Subversion), Git 등의 공개 소프트웨어를 많이 사용한다. CVS는 초창기 형상 관리 도구로 서버는 단순한 명령 구조를 가진 장점이 있지만, 텍스트 기반의 코드만 지원하는 단점이 있다. SVN은 다양한 GUI 도구가 존재하고 gzip 압축을 통해 서버의 공간을 절약하는 장점을 가지고 있지만, trunk, branch, tag가 모두 저장 위치를 점유하는 단점을 가지고 있다. Git는 앞서 나온 CVS와 SVN의 단점을 모두 보완한 장점이 있으나, 중앙 집중형인 CVS와 SVN와는 개념이 다르므로 개발자에게 학습할 시간이 필요하다
- SVN과 Git 형상 관리 기능 이외에도 다른 기능의 서드 파트 GUI환경을 지원한다. 예를 들어 개발자 게시판의 테스트 수행 시 업무 담당자와 개발자 간의 대화 채널을 관리하는 도구까지 제공하고 있으니 형상 관리 도구를 선정하기 전에 서드 파트지원 도구도 파악하고 설치하는 것이 필요하다.

7. UI 테스트(2001020779_14v1)

1-1

테스트 기법 선정

구현된 UI의 사용성을 검증하기 위하여 적합한 테스트 기법을 선정할 수 있다.



[1] 휴리스틱 평가(Heuristic Evaluation)

사용성에 대한 문제를 찾아내기 위한 사용성 공학 방법으로 전문가에 의해 이론과 경험을 근거로 하여 일련의 규칙들을 만들어 놓고 평가 대상이 그러한 규칙들을 얼마나 잘 지키고 있는가를 확인하는 평가 방법

유리스틱 평가

- **목적** : 디자인 전문가들이 사용성 원칙 또는 유리스틱 가이드라인에 비추어 평가하려는 대상의 문제점을 발견하고 디자인에 반영하는 방법.
- **결과물** : 평가보고서.
- **절차** : ① 평가계획을 수립 및 평가를 실행함.
② 발견된 이슈를 취합하고, 개선방향을 논의함.
③ 평가 결과를 정리하여 평가보고서를 작성함.

- ▶ **장점**: 상대적으로 비용이 적게 들어가며, 짧은 시간 내에 시스템의 중요한 문제점들을 발견할 수 있으며, 프로젝트 수행 과정의 전반에 테스트가 가능하여 초기에 문제점을 발견할 수 있음.
- ▶ **단점**: 구체적이고 계량적인 평가 자료를 만들기 어렵고, 사용성 전문가와 실제 시스템의 사용자가 시스템을 바라보는 시각이 다를 수 있다는 점, 그리고 사용성 전문가의 능력에 따라 평가 결과가 달라질 수 있음.

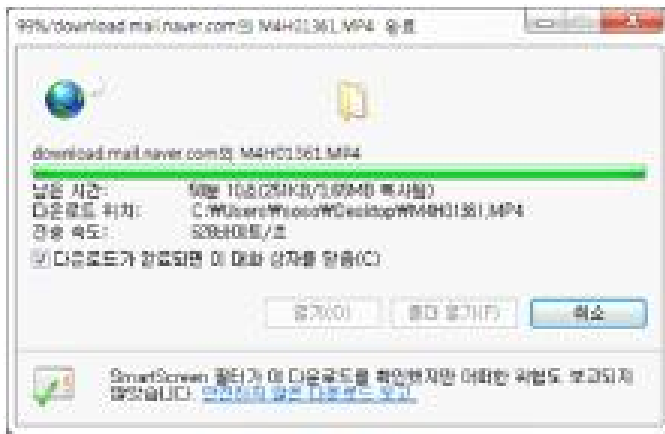
닐슨의 10계명

1) 시스템 상태의 시각화(Visibility of system status)

시스템은 끊임없이 자신의 상태를 알려주어야 한다.

반응시간에 따른 주의력 집중

- 0.1초 : 특별한 반응 없음
- 1.0초 : 데이터 흐름을 놓친다.
- 10초 : 주의집중에 참는 한계
- 더 이상 : progress bar를 이용해야한다.



2) 시스템/실세계 일치(Match between system & real world)

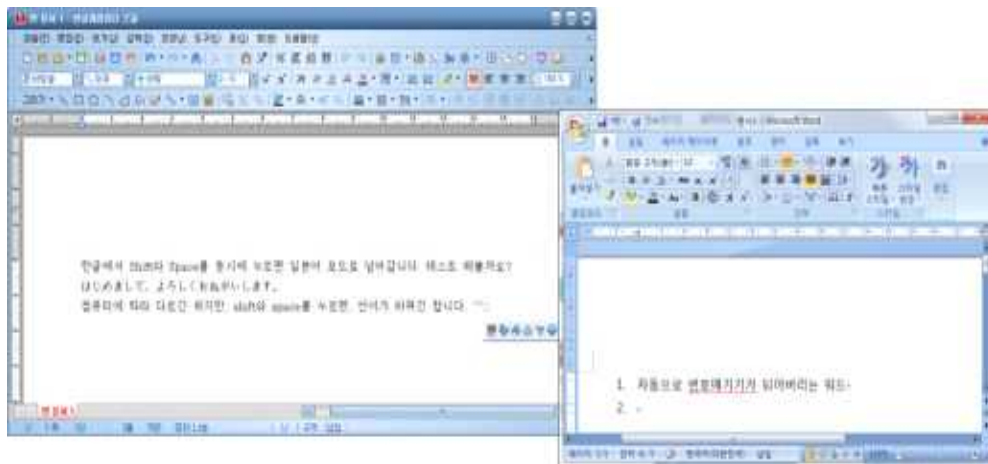
시스템은 사용자의 언어로 말해야 한다. 사용자에게 익숙한 단어, 문구, 개념을 해야 직관적 이해가 가능하다.



3) 제어의 자유(User control & freedom)

인간이 기계를 자유롭게 제어할 수 있어야 한다.

전 상태로 돌아갈 수 있는 기능이 제공되어야 한다.



4) 일치와 표준(Consistency & Standards)

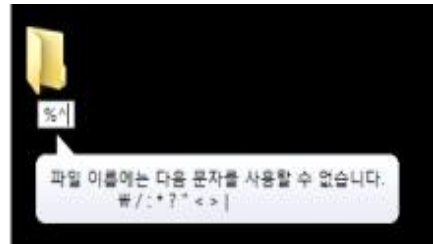
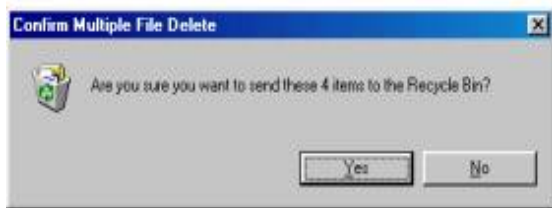
사용자는 서로 다른 용어, 상황, 행동들이 같은 것을 의미하고 있는지 고민하지 않아야 한다. 다른 시스템에서도 사용하는 GUI 표준을 따르는 것이 중요하다.

깜빡임의 시간 간격, 색상의 수, 사운드 효과, 메뉴 선택 명칭과 순서등이 산업 표준들과 일치해야 한다.



5) 에러방지(Error prevention)

에러를 어떻게 알리는가의 문제보다는 에러를 미리 방지할 수 있는가의 문제가 중요하다. 사용자가 심각한 에러를 저지르기 전에 경고할 수 있어야 하며 메뉴나 명령어의 용어가 모호하지 않아야 한다.

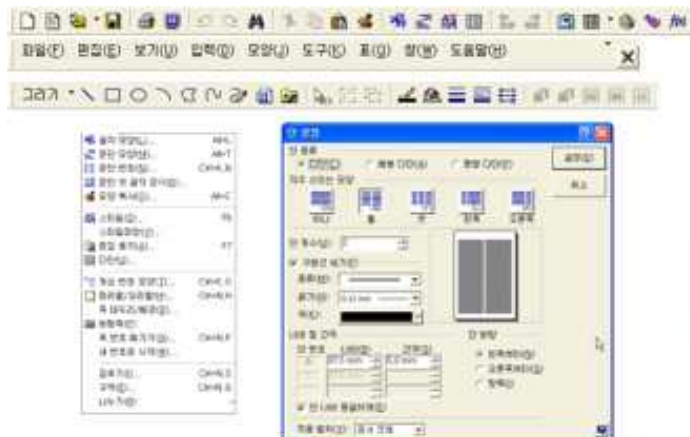


6) 기억보다 인식(Recognition rather than recall)

대상, 행동 옵션을 시각화하는 것이 중요하다.

GUI메뉴들, 아이콘들은 행위 유발성

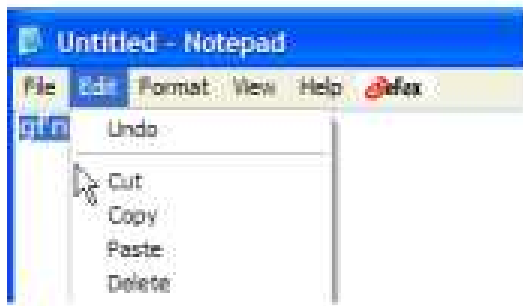
활성/비활성, 데이터 강조/비강조, 의미있는 그룹 사이이 경계선



7) 융통성과 효율성(Flexibility & efficiency of use)

초보자와 경험 있는 사용자가 모두 만족할 수 있는 시스템

하나의 명령을 수행하는 방법에도 대체 방법이 필요(메뉴와 단축키 등)



8) 간소화된 디자인(Aesthetic & minimalist design)

불필요한 정보는 중요한 정보의 시각화를 방해시키므로 제외
모든 아이콘이 중복된 느낌없이 서로 구분
배경과 분명하게 구분



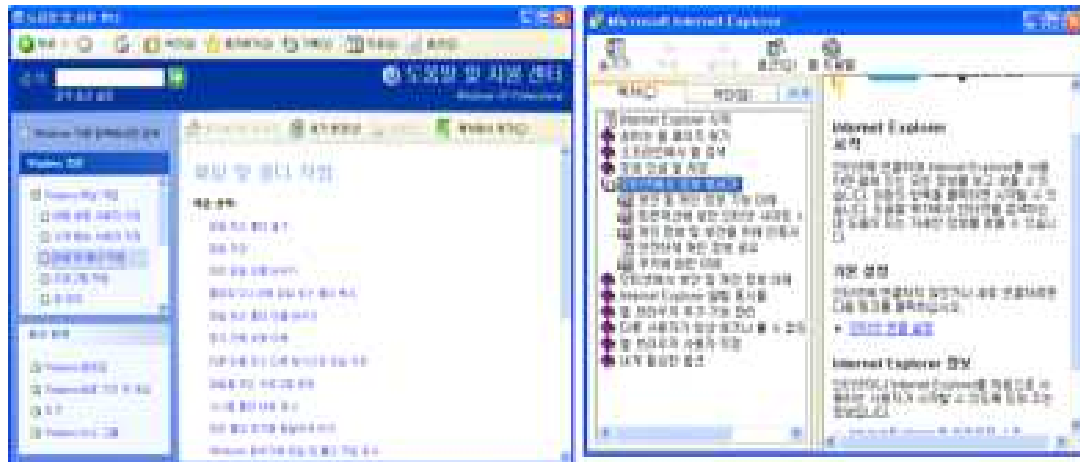
9) 에러인식, 진단, 복구(Help users recognize, diagnose & recover from errors)

에러메시지는 자연어로 된, 사용자에게 익숙한 문장이어야 하며 문제의 원인을 명시하거나 해결책을 제안할 수 있으면 더욱 좋음



10) 도움말과 문서화(Help and documentation)

복잡성이 증가하는 시스템의 경우 도움말 기능과 문서화는 필수적



[2] 페이퍼 프로토타입 (Paper Prototype) 평가

프로토타입의 가장 빠른 방법으로 제품의 전반적인 컨셉과 흐름을 잘 보여주며, 보는 사람들이 최종 제품에 대한 기대를 갖지 않고 더 자유롭게 의견을 개진하면서 발전시킬 수 있는 방법이다.

- ▶ 제품의 전반적인 컨셉과 흐름을 잘 보여주며, 보는 사람들이 최종 제품에 대한 기대를 갖지 않고 더 자유롭게 중요한 의견을 개진하면서 발전시킬 수 있는 방법이다.
- ▶ 사용자들에게 보다 현실감 있는 테스트를 제공하기 위해서 단순히 UI 스케치에 그치지 않고, 프로토타입을 구현하게 될 경우에는 테스트까지 포함해서 한달 이상의 시간이 소요될 수 있다.
- ▶ 프로토타입 작성시 포함되어야 할 중요한 내용을 빠짐없이 체크할 수 있도록 체크리스트를 준비한다.



페이퍼 프로토타입 평가

- **목 적** : 실제 출시될 제품의 디자인을 미리 경험해 볼 것으로서 수정 및 보완해야 할 부분을 발견함.
- **결과물** : 프로토타입.
- **절 차** : ① 종이 위에 제품 및 시스템 개략도를 그림.
② 페이지에 특정 순서에 따라 번호나 설명을 별도로 붙임.
③ 사용자, 디자이너들이 실제 형태를 테스트하면서 발견시킴.

[3] 선호도 (Preference) 평가

“A가 B보다 더 좋다”, “C가 D보다 더 편리하다”와 같이 제품이나 서비스에 대한 사용자의 선호도에 영향을 미치는 속성들을 파악하고, 중요도에 따른 선호도를 예측하기 위하여 사용된다. 사용자의 니즈에 대응할 수 있는 평가방법이다.

선호도 평가

- **목 적** : 사용자의 감성을 제대로 읽어내기 위해 과학적인 시점에서 객관적으로 해석함
- **결과물** : 선호도 평가.
- **내 용** : 수집되는 자료의 특성에 따라서 적절한 측정법을 적용해야 함
 - ① 점수 (Rating)
 - ② 순위 (Ordinal Ranking)
 - ③ 태도-기반 선호도 (Attitude-Based Preference)
 - ④ 속성-기반 선호도 (Attribute-Based Preference)

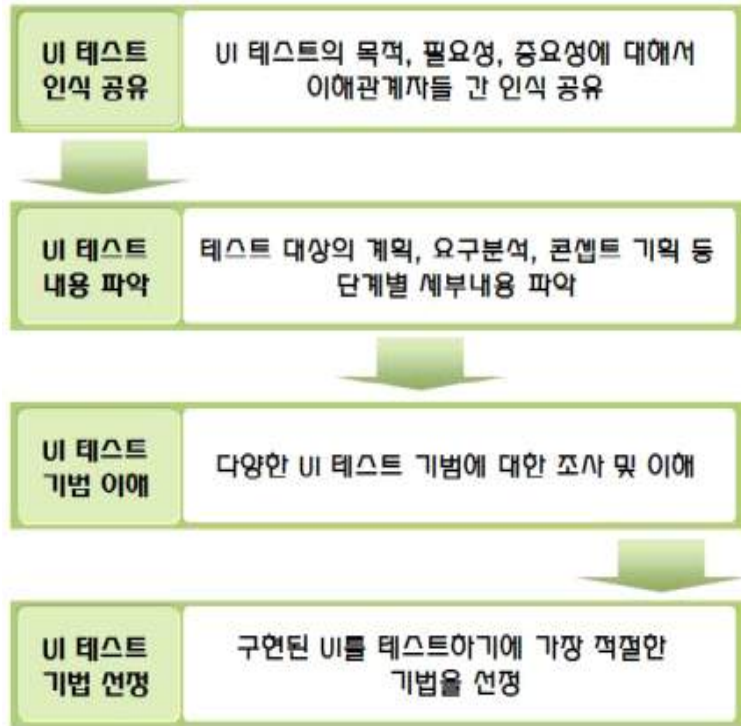
[4] 성능(Performance) 평가

사용자가 실제로 제품이나 서비스와 연관된 것을 사용해 보고 태스크(TASK) 별 학습성, 효율성, 기억용이성, 오류, 만족도 등에 대해 평가하여, 그 결과를 바탕으로 성능을 개선한다. 제품이나 서비스를 개발하는 단계에 맞춰 평가를 진행한다.

성능 평가

- **목 적** : 개발 마지막 단계에서 각 제품이나 서비스의 태스크들이 지닌 장단점을 파악하기 위해 실행함.
- **결과물** : 학습성, 효율성, 기억용이성, 오류, 만족도에 대한 평가.
- **내 용** : ① 학습성 : 쉽게 학습할 수 있는가
② 효율성 : 일단 학습하면 매번 신속하게 사용할 수 있는가
③ 기억용이성 : 사용한 기능을 능숙하게 다시 수행할 수 있는가
④ 오류 : 오류가 적고, 사용자가 상황을 쉽게 극복할 수 있는가
⑤ 만족도 : 사용하는 것이 즐겁고 만족스러운가

[5] 사용성 테스트 기법 선정하기 수행절차



1-2

테스트 환경 구축

선정된 기법에 적합한 테스트 환경을 구축할 수 있다.

[1] 사용자 중심 디자인(User Centered Design)

사용자 중심 디자인은 사용자가 원하고 필요로 하는 것에 대해 사용자의 한계 능력과 상황에 맞추어 디자인 프로세스를 통해 사용자를 중심으로 사고하여 만들어 내는 인터페이스, 서비스, 제품 디자인의 철학적 접근 방법이자 디자인 사상이다. 디자인의 각 단계에서 인간을 중심으로 하는 집중적인 관심과 연구가 이루어지며 최종적 단계에서의 결과물인 디자인은 인간사용자의 편의와 복지의 실현에 초점을 맞추게 된다.

이를 위해 각 단계의 디자인 문제 해결과정에서 디자이너는 사용자가 어떻게 사용하고 있고 어떻게 사용하게 될지에 대한 분석과 예상을 하는 디자인 리서치를 통해 디자인을 만들어 나가게 되는데 이때 실제의 세상에서 행해지는 실제 사용자의 행태와 환경을 이해하는 것이 필수적이다.

다른 디자인 사상과 많은 차이점을 보이는 것은 사용자로 하여금 시스템이나 기능에 맞추어 적응하게 하거나 학습하도록 유도하기보다는 특히 사용자가 무엇을 할 수 있고, 무엇을 원하고, 무엇을 할 필요가 있는지에 따른 사용자 인터페이스의 최적화를 통해 디자인이상을 현실화하는 노력에 있다.

[2] 인터랙션 디자인(Interaction Design)

인터랙션이란 입출력 장치를 매개로 디지털 시스템과 사람이 주고 받는 일련의 의사소통 과정으로, 인터랙션 디자인이란 사람의 행동과 이에 반응하는 시스템의 절차를 설계하는 것이다. HCI(Human-Computer Interface)의 인터랙션은 사용자에게 최적의 경험을 제공하는 기본 단위로서, 인터페이스, 인터랙션, 경험까지 포괄하는 전반적인 상호작용을 말하며, 시스템 디자인에서 인터랙션은

시스템의 행동적인 측면에 초점을 맞추는 상대적으로 작은 개념을 말한다.

[3] 프로토타입(Prototype)

원래의 형태 또는 전형적인 예, 기초 또는 표준을 말하는 것으로 정보시스템의 미완성 버전 또는 중요한 기능들이 포함되어 있는 시스템의 초기모델을 의미한다. 프로토타입은 사용자의 모든 요구사항이 정확하게 반영될 때까지 계속해서 개선되고 보완되는 것이 일반적이다.

[4] 와이어프레임 (Wireframe)

최종 화면에 표시될 콘텐츠를 간단히 요약하여 보여주는 것으로서, 색상, 타이포그래피, 이미지를 생략하는 경우가 많다. 도식, 청사진, 또는 프로토타입이라고 부르기도 한다.

[5] 시장현황보고서

기업이 참여하고 있는 사업 분야의 시장 현황 및 특성을 파악하여, 시장 매력 요인을 도출하고 성공가능성을 예측하기 위한 보고서이다. 시장 특성, 기회요인, 위협요인을 파악하여 브랜드 컨셉, 차별화 요인, 신제품의 컨셉 도출, 디자인 개발 방향을 수립하는데 활용된다.

[6] 기술현황보고서

기업이 참여하고 있는 사업 분야의 시장 현황 및 특성을 파악하여, 시장 매력 요인을 도출하고 성공가능성을 예측하기 위한 보고서이다. 시장 특성, 기회요인, 위협요인을 파악하여 브랜드 컨셉, 차별화 요인, 신제품의 컨셉 도출, 디자인 개발 방향을 수립하는데 활용된다.

[7] 사용성 테스트 환경 구축하기 수행 절차



[8] 사용성 테스트 과제 설정

- ▶ 사용성 테스트의 과제를 온라인 쇼핑몰을 예로 들면, 어버이날에 부모님께 선물할 제품을 실제로 구입하는 과정이 쉽게 구현이 되어 있어서 사용자가 어려움을 겪지 않고 간단하게 제품을 구입할 수 있는지를 확인하는 것이 과제가 될 수 있다.
- ▶ 사용성 테스트의 과제를 그룹웨어로 예로 들면, 관계사와 회의를 실시하는 기안을 작성하여, 비용 등을 입력하고, 상사에게 결재를 올릴 수 있는지 확인하는 것이 과제가 될 수 있다.
- ▶ 사용성 테스트의 과제를 공공기관을 예로 들면, 사용자가 퇴근 후에 주민등록등본 등의 개인 증빙 자료를 발급 받으러 가려고 하는데, 주민센터 운영시간, 방문하지 않고 온라인 발급 여부, 어떻게 하면 되는지에 대한 절차 등이 상세하게 설명되어 있는지 찾는 것이 과제가 될 수 있다.

1-3

사용성 테스트 계획서 작성

사용성 테스트를 수행하기 위하여 계획서를 작성할 수 있다.

[1] 사용성



<Jakob Nielsen>

Jakob Nielsen은 사용성을 학습용이성, 사용효율성, 기억용이성, 최소한의 에러, 주관적 기쁨의 요건을 충족시킴으로써 얻어지는 시스템의 특성이라고 주장하였다.

UPA(Usability Professionals' Association, 現 UXPA)에서는 사용성을 “회사가 비용을 절감시키고 사용자의 요구와 부합되는 제품을 만들어 내기 위해 개발사이클 전체에 걸쳐 사용자의 피드백을 받는 제품 개발 방식”이라고 정의하였다.

[2] 사용성 테스트(Usability Test)

- 소프트웨어의 품질 측면에서 사용성은 HCI(Human Computer Interface)를 구성하고 있는 핵심 원리
- 부가적인 고려사항이 아닌 필수적으로 갖추어야 할 시스템의 핵심 요소
- 사용성 평가가 기존의 소프트웨어 테스트와 다른 점은 테스트의 중심이 사용자

[3] 테스트 계획서

테스트 계획을 평가하는 데는 기능의 실현방법과 품질의 평가방법에 관해 고려할 필요가 있다. 효과적인 경험치가 지표화되어 표에 기재된 경험치는 항상 동일하게 중요한 것은 아니다. 상황에 의해 적용할 수 없는 것도 있지만 일반적인 해설과 간단한 근거를 나타내고 있다. 근거 항목에는 적용을 판단할 때에 도움이 되는 정보를 서술한다.

[4] 테스트 계획서 항목

테스트 계획서 항목에는 테스트 계획 ID, 서문, 테스트 항목, 테스트 대상 기능, 테스트 대상외의 기능, 테스트 방법, 합격/불합격 기준, 테스트 시작 조건과 중단 기준, 재개 요건, 성과물, 테스트 작업, 환경, 책임자, 인원 및 교육, 스케줄, 리스크, 승인 등이 있다.

[5] 인터페이스(Interface)

컴퓨터와 같은 디지털 시스템의 입출력 장치, 사물 간 또는 사물과 인간간의 의사소통이 가능하도록 일시적 혹은 영속적인 접근을 목적으로 만들어진 물리적, 가상적인 매개체.

[6] 그래픽 사용자 인터페이스(Graphical User Interface)

컴퓨터를 사용하면서, 그림으로 된 화면 위의 물체나 특성, 색상과 같은 그래픽 요소들을 어떠한 기능과 용도를 나타내기 위해 고안된 사용자를 위한 컴퓨터 인터페이스.

[7] UI디자인(User Interface Design)

- 인터페이스, 즉 정보기기나 소프트웨어의 화면 등 사람과 접하는 면을 설계하고 디자인하는 것
- 스마트기기의 대중화로 스마트폰, 태블릿 PC 등의 모바일 UI 디자인이 주목을 받고 있으며, IA(Information Architecture) 설계와 인터랙션 설계, GUI 디자인을 포함한다.

[8] UX 디자인(User Experience Design)

제품과 시스템, 서비스 등을 사용자가 직간접적으로 사용하고 경험하면서 느끼고 생각하는 총체적 경험을 의미하며, ISO 정의에 따르면 User Experience(UX)란 사용자가 제품, 시스템, 혹은 서비스의 사용 혹은 기대되는 사용 결과에서 오는 인식과 반응(ISO 9241-2120(2010))이다.

[9] 테스트 설정

사용성 테스트를 통해서 테스트하고자 하는 제품의 모든 기능을 평가하는 것은 사실상 불가능하다. 따라서 사용성 테스트에서는 대상 제품의 중요한 기능이나 반드시 테스트를 통해 검증하고 싶은 기능을 선별하여 사용성 테스트를 통해 수행하는 태스크로 정의한다. 그리고 태스크는 자연스러운 흐름에 맞는 시나리오를 통해 수행되어야 한다.

[10] 사용성 테스트의 절차



[11] 사용성 테스트 계획서의 구성안

사용성 테스트 개요

- 사용성 테스트에 대한 개요에 대해서 작성함
 - ex) 주요 고객들이 균등하게 제품을 이용할 수 있는지 이해하기 원한다.
 - ex) 사용설명서가 인터페이스상의 알려진 문제들을 모두 상쇄하는지 여부

사용성 테스트 목적

- 과제에 포함되어야 하는 주요 이슈를 정리하여 작성함
 - ex) 한 줄에 20개의 문자가 표현되는 것이 의사소통하기에 적절한가?
 - ex) 제어판에 있는 모든 버튼들은 설명서 없이 정확하게 사용 가능한가?
 - ex) 사용설명서가 인터페이스상의 알려진 문제들을 모두 상쇄하는지 여부

사용자 프로파일

- 어떠한 사용자를 사용성 테스트에 참여시킬지에 대해서 작성함
 - ex) 타겟시장, 나이, 인터넷 사용능력, 사이트 접속횟수, 제품구매 횟수, 특정 기능에 대한 학습능력 등

사용성 테스트 디자인

- 사용성 테스트를 어떻게 실시할 것인지에 대한 방법론을 작성함
 - ex) 테스트 방법, 참가자 수, 테스트 장소, 세션의 소요시간, 테스트의 절차, 작성이 요구되는 문서 등

사용성 테스트 진행 순서

- 오리엔테이션 => 배경(테스트 사이트) => 과제 시뮬 =>
테스트 게임 설문조사 => 참가자 면담

결과 요약

- 어떻게 결과를 취합 할 것인지, 어떠한 결과 데이터를 수집할 것인지 기술

개선안 보고

- 개선안을 무엇을 기초로 작성할지, 어떠한 내용들을 포함할 것인지 기술

[12] 사용자경험 허니콤 모형



- * Useful : 유용한
- * Useable : 사용할 수 있는
- * Findable : 발견할 수 있는
- * Credible : 신뢰할 수 있는
- * Accessible : 접근할 수 있는(장애인, 노약자)
- * Desirable : 매력적인
- * Valuable : 가치있는

Useful. 얼마나 유용한가?

전문가로서 우리는 관리자들이 그려놓은 선 안을 색칠하는 것에만 만족해서는 안 된다. 우리가 만드는 제품과 시스템이 유용한지 지속적으로 재고하고, 좀더 유용하고 혁신적인 솔루션을 만들기 위해 기술과 매체에 대한 우리의 모든 지혜와 깊은 지식을 용감하게, 창의적으로 적용해야 한다.

Usable. 사용하기 쉬운가?

사용하기 쉬운 사용하기 쉽다는 것이 매우 중요하긴 하지만, 인터페이스를 중심으로 인간-컴퓨터의 상호작용을 파악하는 방식이나 시각은 웹 디자인의 모든 차원을 설명해 주지는 못한다. 간단히 말해, 사용성은 필요 조건이지 충분 조건은 아니다.

Desirable. 매력적인 서비스인가?

매력적인 효율성의 추구도 중요하지만 이미지나 아이덴티티, 브랜드, 기타 감성적인 디자인 요소들의 강렬함이나 가치도 그만큼 중요하므로 적절한 조율이 필요하다.

Findable. 길을 잃지 않고 잘 찾을 수 있나?

사용자들이 필요로 하는 것을 쉽게 찾을 수 있도록, 웹사이트는 네비게이션이 가능하게, 개체는 위치 파악이 가능하게 만들어야 한다.

Accessible. 모든 사람이 접근할 수 있는가?

건물에 엘리베이터와 휠체어용 경사로가 있듯이 웹사이트도 장애가 있는 사람들(전체 인구의 10% 이상)이 접근할 수 있도록 만들어져야 한다. 오늘날에는 그것이 좋은 사업 기회이기도 하고 윤리에 맞는 일이기도 하다. 언젠가는 이것이 법에 의해 강제성을 갖게 될 것이다. 디자인을 할 때 처음부터 표준을 준수하면 모바일 기기에서도 접속하기 쉬워진다.

Credible. 믿을 만한가?

어떤 디자인적 요소들이 사용자의 신뢰도의 영향을 미치는지 알 수 있다.

Valuable. 투자 가치가 있는가?

마지막 특성은 단지 사용자에게 관한 것만은 아니다. 사이트는 광고주에게 가치를 전달해야 한다. 비영리 사이트의 경우 사용자 경험은 사이트의 목표에 이바지해야 한다. 영리 사이트의 경우 수익을 창출해야 하고 고객 만족도를 높여야 한다.

2-1

사용성 테스트 수행

선정된 테스트 기법에 맞는 도구를 활용하여 테스트를 수행할 수 있다.

[1] 파일럿 테스트

파일럿 테스트란 주로 컴퓨터 프로그램 등의 최신 기술을 개발하여, 실제 상황에서 실현하기전에 소규모로 시험 작동 해보는 것을 의미한다. 대규모 프로젝트를 실행하거나 플랜트를 본격적으로 가동하기 전에, 발생할 수 있는 여러 가지 변인들에 미리 파악해서 수정 보완하기위해, 모의로 시행해 보는 것을 말하기도 한다.

[2] 심층 인터뷰 (In Depth Interview)

1명의 응답자와 일대일 면접을 통해 소비자의 심리를 파악하는 조사법으로, 어떤 주제에 대해 응답자의 생각이나 느낌을 자유롭게 이야기함으로써 응답자의 내면 깊숙이 자리잡고 있는 욕구, 태도, 감정 등을 발견하는 면접조사이다. 심층 인터뷰의 경우, 조사원의 면접 및 분석능력에 따라, 조사 결과의 신뢰성과 타당성이 크게 변할 수 있으므로 철저한 사전 준비와 시간이 필요하다.

심층 인터뷰

목 적 : 소수의 개별 사용자에게 대한 깊이 있는 정보를 수집함

결과물 : 사용자의 동기나 태도, 의견에 대한 다양하고 심층적인 정보

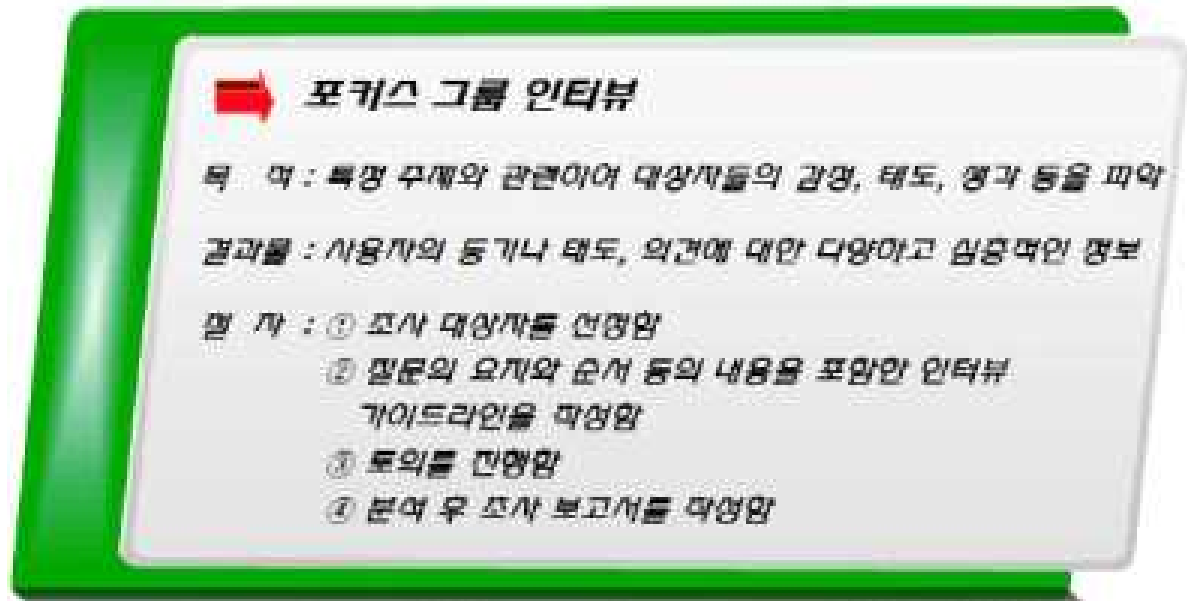
절 차 : ① 조사 대상자 섭외를 위한 리크루팅을 진행함

② 일대일 개별 인터뷰(녹음/녹화)를 진행함

③ 인터뷰 결과 분석 후 보고서를 작성함

[3] 포커스 그룹 인터뷰

표적시장으로 예상되는 소비자를 일정한 자격 기준에 따라 6~12명 정도 선발하여, 한 장소에 모이게 한 후, 면접자의 진행 아래 조사 목적과 관련된 토론을 함으로써 자료를 수집하는 방법이다.

A green-bordered box with a red arrow pointing right. Inside, the text describes the purpose, results, and steps of a Focus Group Interview.

→ 포커스 그룹 인터뷰

목 적 : 특정 주제와 관련하여 대상자들의 감정, 태도, 생각 등을 파악

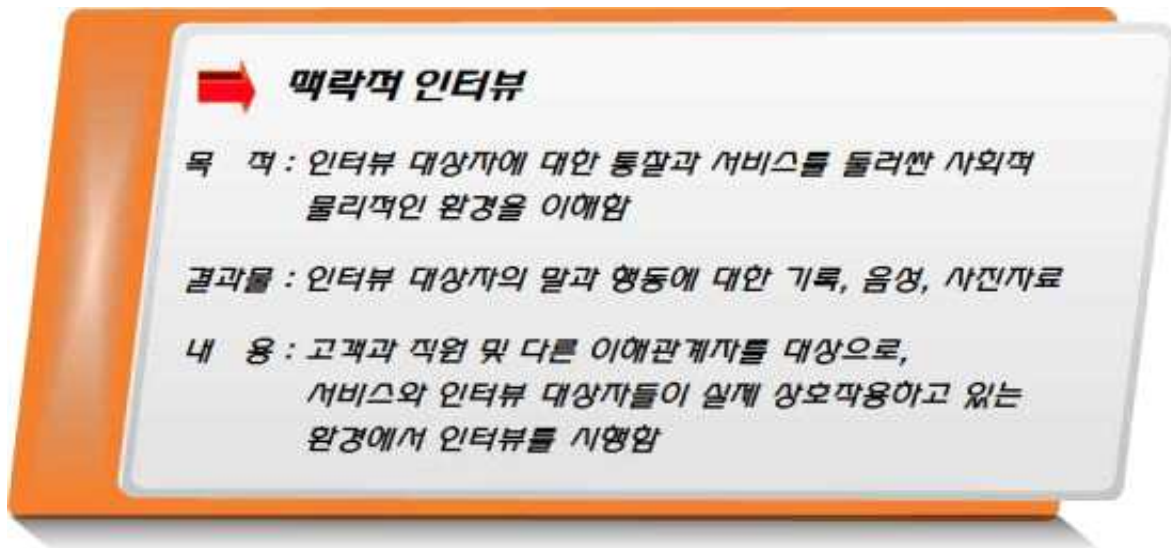
결과물 : 사용자의 동기나 태도, 의견에 대한 다양하고 심층적인 정보

절 차 : ① 조사 대상자를 선정함
② 질문의 요지와 순서 등의 내용을 포함한 인터뷰 가이드라인을 작성함
③ 토의를 진행함
④ 본역 후 조사 보고서를 작성함

[4] 맥락적 인터뷰 (제인구달의 침팬지연구)

맥락적 인터뷰는 서비스 과정 가운데 특정 상황이나 맥락에서 이루어지며, 인터뷰를 진행하면서 리서치는 민족지학적 기법으로 특정 행동을 관찰하고 조사할 수 있는 방법이다.

※ 문화 기술지 연구라고도 하는 민족지학적 연구는 한 집단의 의사소통적 행위를 포함한 문화적 행위를 기술하고 해석하기 위해 그 집단의 생활 양상을 현장에서 조사, 수집, 기록하고 분석하는 연구 방법이다.

An orange-bordered box with a red arrow pointing right. Inside, the text describes the purpose, results, and content of a Contextual Interview.

→ 맥락적 인터뷰

목 적 : 인터뷰 대상자에 대한 통찰과 서비스를 둘러싼 사회적 물리적인 환경을 이해함

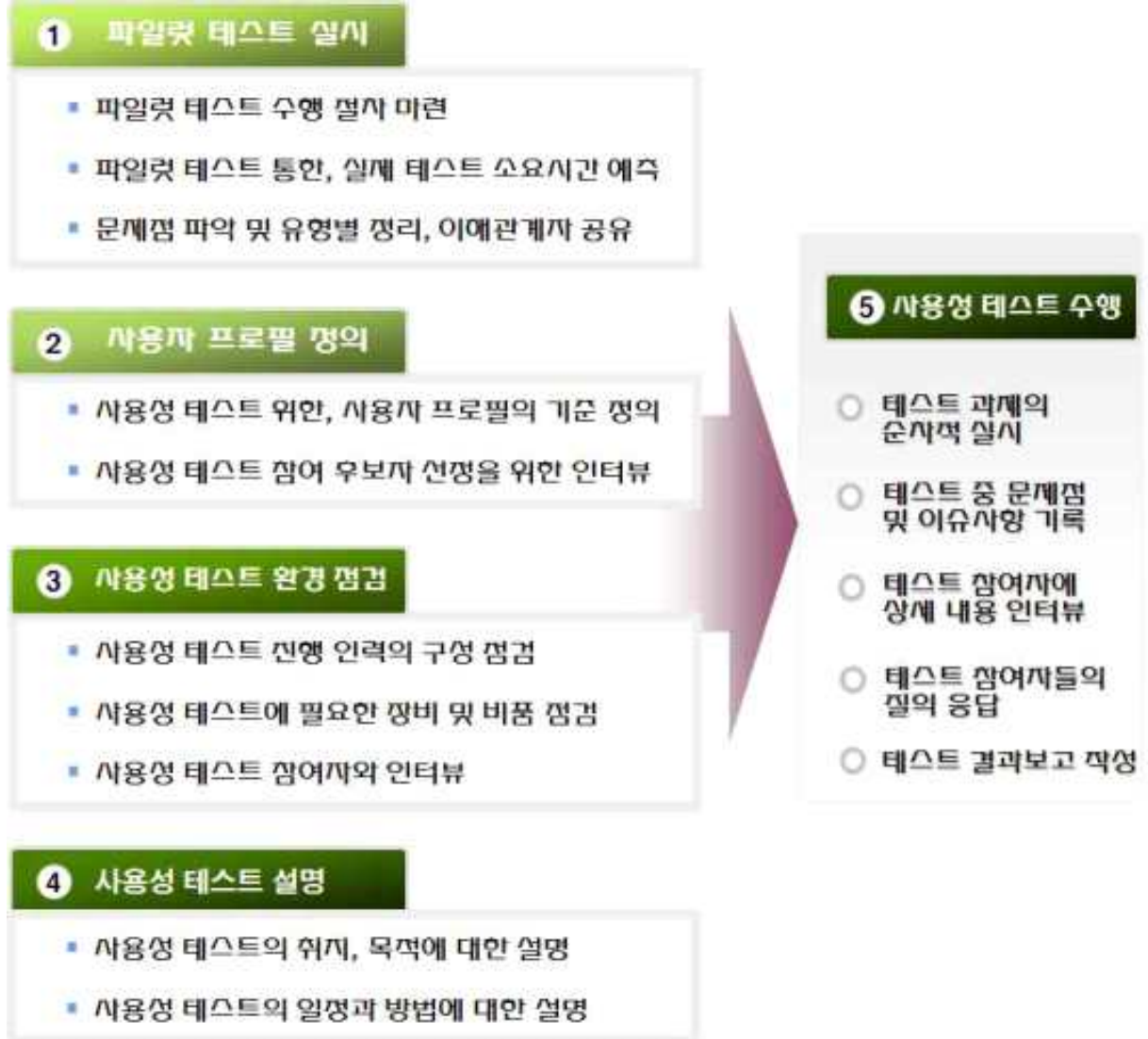
결과물 : 인터뷰 대상자의 말과 행동에 대한 기록, 음성, 사진자료

내 용 : 고객과 직원 및 다른 이해관계자를 대상으로, 서비스와 인터뷰 대상자들이 실제 상호작용하고 있는 환경에서 인터뷰를 시행함

[5] 사용성 테스트를 위한 사용자 프로파일(예시)

항목	내용
개인이력	<ul style="list-style-type: none"> • 나이 • 성별 • 컴퓨터에 대한 이해 정도 • 오른손/왼손잡이 여부(마우스 사용에 영향을 줄 수 있는 요인) • 학습스타일(직접 경험을 통해 배우는 것을 선호하는지, 독서를 통해 간접적으로 배우는 것을 좋아하는지 등...)
학력	<ul style="list-style-type: none"> • 최종 학력 • 수강 과목 • 전공
컴퓨터 경험	<ul style="list-style-type: none"> • 사용시간 • 사용빈도 • 사용기종 및 장치 • 사용 OS(Operating System) • 사용한 스크린 인터랙션 종류(GUI, DOS)
제품 경험	<ul style="list-style-type: none"> • 사용빈도 • 수행한 과업의 종류 및 빈도 • 회사 제품의 사용자 여부
직업이력	<ul style="list-style-type: none"> • 현재 및 이전 직종 • 담당 업무 • 이수한 직업교육 • 현재 직장 복무 기간

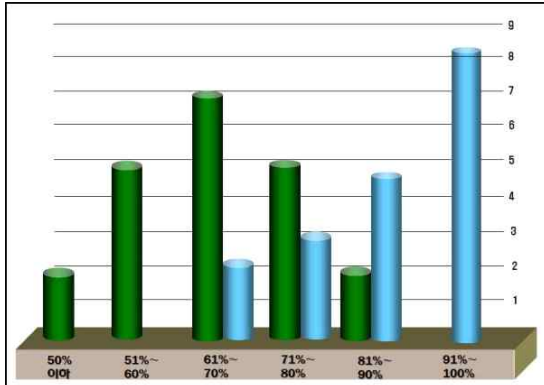
[6] 사용성 테스트 수행 절차



평가 분석서 작성 및 이슈 도출

- 테스트 자료를 수집하여 사용성 평가 분석서를 작성할 수 있다.
- 사용성 평가 분석서를 토대로 이슈 사항을 도출할 수 있다.

1] 태스크 성공 매트릭스



태스크(업무)가 포함되어 있는 사용성 테스트 연구에서 태스크 성공 매트릭스는 가장 일반적이고 보편적인 방법이다. 개선 이전의 UI/UX 디자인과 개선 이후의 UI/UX 디자인의 태스크 성공 여부(이진성공율)의 빈도 분포로 표현하여 비교한다.

[2] 에러 매트릭스

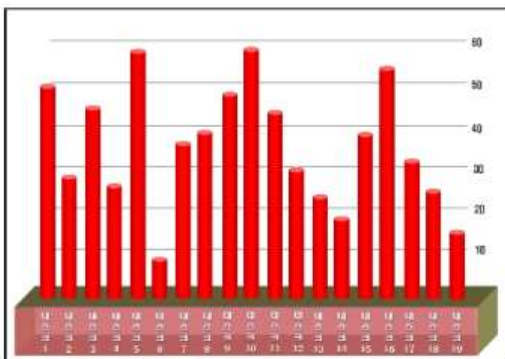
태스크를 수행하는 동안 사용자가 범한 실수를 수집하여 분석하는 기법으로, 태스크 진행 중 얼마나 많은 실수가 있었는지, 이러한 실수가 제품의 어느 부분에서 발생했는지, 디자인에 따라 어떤 종류의 에러가 얼마나 자주 일어나는지, 일반적으로 무엇이 어떻게 유용한지에 대해서 말해줄 수 있다.

예를 들어서, 에러로 인해 데이터가 손실되는 등 효율성에서 중대한 손실을 가져올 때, 에러로 인해 고객센터로 걸려오는 통화량이 증가되는 등 많은 비용을 발생시킬 때, 에러가 태스크의 실패를 야기할 때 등이다.

[3] 효율성 매트릭스

효율성 매트릭스는 태스크를 완료하기 위해서 걸리는 시간 뿐 아니라, 인지적 노력과 물리적 노력의 양을 중요하게 여길 때 효과적이다. 효율성은 흔히 태스크를 완료하는데 필요한 단계나 행동의 개수에 의해 측정되고, 또는 태스크마다 걸린 평균 시간과 태스크 성공률의 비율에 따라 측정되기도 한다.

[4] 시간 기반 태스크 매트릭스



시간 기반 태스크는 어떤 제품에서건 효율성을 측정할 수 있는 방법이며, 태스크 완료 시간 또는 단순히 태스크 시간이라고 하기도 한다. 대부분 사용자가 태스크를 빠르게 완료하는 것을 더 나은 경험으로 규정하고 있으며, 사용자의 기대보다 짧은 시간 내에 태스크를 완료하는 일은 많지 않다. 테스트 진행자가 스톱워치로 기록하거나, 자동화 툴(Ego Browser, Data Logger, Bailey의 Usability Testing Environment 등)을 사용하는 방법, 사용자가 시계 끄고 켜기 방법 등으로 데이터를 수집한다.

[5] 데이터 이력기록기 (Data Logger)



온도·유량·압력 등 공정변수의 아날로그 입력을 디지털 수로 변환하여 자동적으로 기록하는 장치를 말한다. 계측 끝에서부터의 아날로그 신호를 AD변환기를 통해서 디지털 신호로 변환하고, 이것으로 자동 타이프라이터를 작동시켜 일정한 시간주기로 계측값을 시트 위에 나타낸다. 또 관리 한계를 설정하는 핀셋이 등을 갖추고, 계측값이 한계를 넘으면 적자로 인쇄하거나 자동적으로 경보를 내는 것이 많다. 데이터 이력기록의 본체는 간단한 계산도 할 수 있는 구조가 많으며, 고급 모델은 완전한 디지털 컴퓨터의 기능을 가진다.

이 경우에는 전산 자동이력기록기로서 구별하는 경우도 있다. 공업 플랜트에서의 계측은 관리실의 패널 위에 다수의 계기를 설치하여 집중적으로 감시하는 방법이 취해지나, 데이터 이력기록기는 이 다수의 계기를 1대의 자동 타이프라이터로 대체해 놓은 것으로서 감시를 더욱 수월하게 하고 있다.

[6] 옵저베이션 (Observation) → 관측

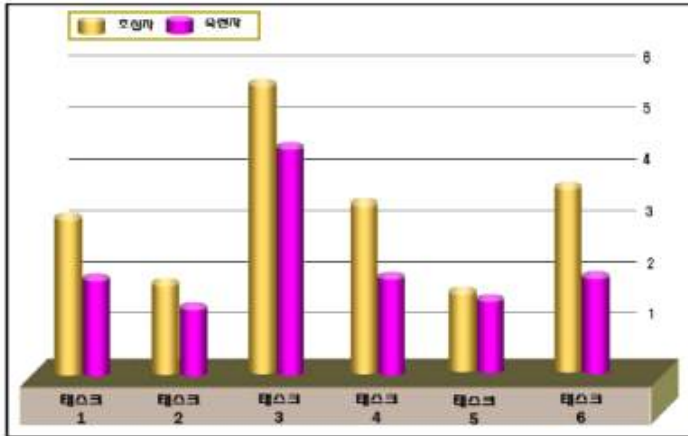
어떤 대상이나 어떤 과정이 어떻게 구성이 되어 있으며, 어떻게 해서 생기는가의 사실을 있는 그대로 확인하는 것으로, 넓은 의미에서는 실험을 포함시킬 수 있지만, 일반적으로 실험처럼 대상이나 과정에 인위적인 간섭을 가하지 않는 것을 의미한다.

[7] 테스트 신뢰성

테스트의 신뢰성이란, 테스트의 결과 또는 테스트를 통해 측정된 측정치, 측정기준을 신뢰할 수 있을 정도로 정확성과 안정성, 그리고 일관성을 가지고 있는가를 의미하는 것이다. 테스트의 설계 및 절차에 기본적으로 오류가 있다면, 테스트를 아무리 잘 수행하였다 하더라도, 해당 테스트의 결과는 신뢰성에 의심을 받게 되며, 테스트 신뢰성에 문제가 있는 테스트 결과는 아무런 의미도 없고 활용할 수도 없게 된다.

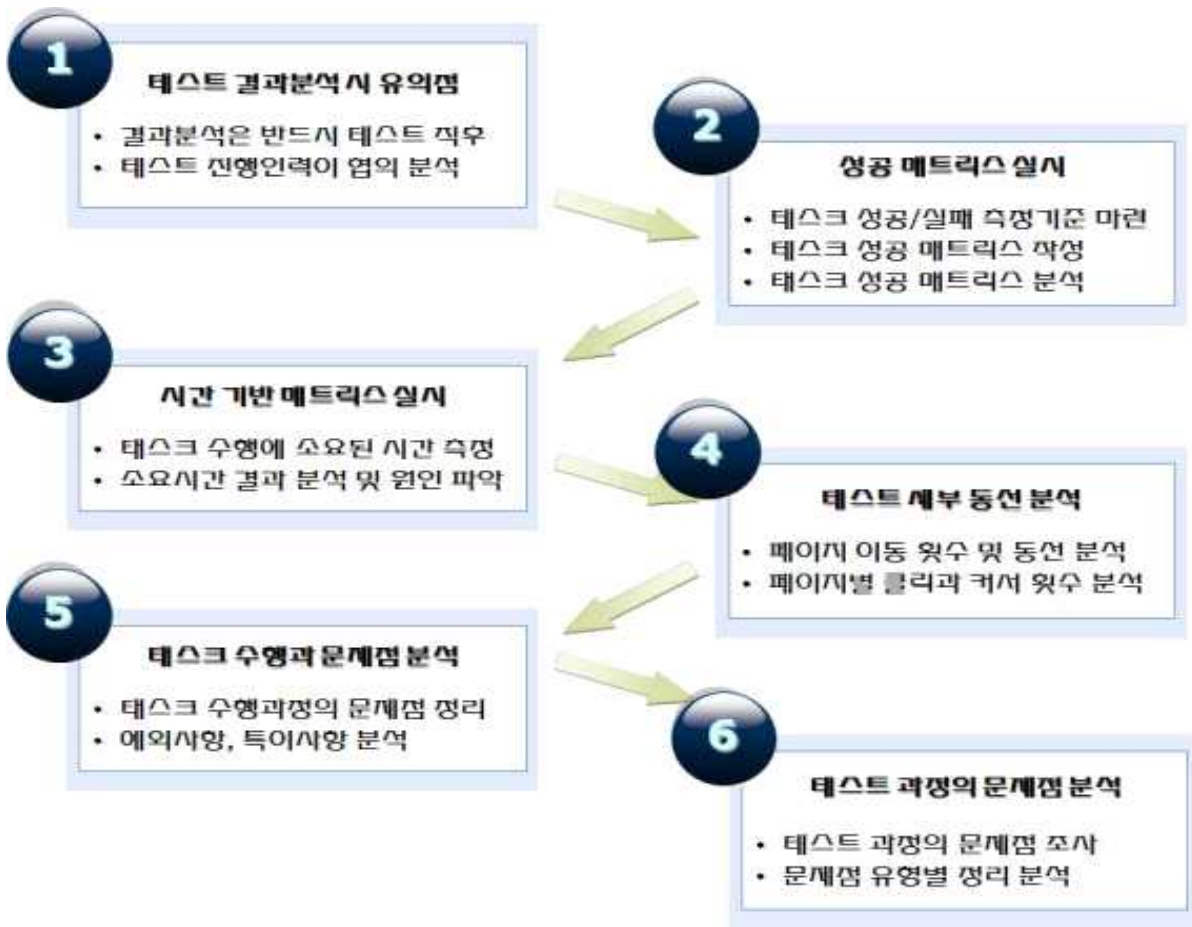
또한 테스트를 진행하는 사람이 중립적인 시각을 가지지 않고 있거나, 제품에 대해서 상세히 잘 알고 있는 경우에도 사용성 테스트의 신뢰성에는 문제가 발생할 수 있다. 사용성 테스트 신뢰성에 문제가 있는 테스트 결과를 기반으로 제품에 반영이 되고 최종 출시가 되면 기업에는 막대한 피해를 가져올 수 있으므로, 테스트 신뢰성 확보에는 많은 노력을 기울여야 한다.

[8] NEM 기법 (Novice Export ratio Method)



대상 제품을 처음으로 사용해 보는 초심자 (Novice)와 대상 제품의 설계 및 개발에 참여한 숙련자(Expert)를 사용성 테스트에 참여시켜서 대상 제품의 태스크 수행시간을 비교하여 문제점을 객관적으로 제시하는 사용성 테스트 기법이다.

[9] 사용성 평가 분석서 작성하기 수행절차



[10] ISO/IEC 9126 품질 모델 4개 계층

- 1) 제 1계층 : 사용자 관점에서 소프트웨어의 품질 목표를 정의한다.
- 2) 제 2계층 : 품질 목표를 달성할 수 있는 광범위한 품질특성이다.
- 3) 제 3계층 : 상위 특성을 구성하는 구체적인 품질 부특성이다.
- 4) 제 4계층 : 소프트웨어 특성을 측정하기 위한 메트릭이나 품질인자가 위치한다.

[11] ISO/IEC 9126 소프트웨어 품질특성

1) 기능성

특정 조건에서 사용될 때 명시된 요구와 내재된 요구를 만족하는 기능을 제공하는 소프트웨어 제품의 능력

2) 신뢰성

명시된 조건에서 사용될 때, 성능 수준을 유지할 수 있는 소프트웨어 제품의 능력.
성숙성, 결함 허용성, 회복성, 준수성 등이 있다.

3) 사용성

명시된 조건에서 사용될 경우, 사용자에게 의해 이해 및 학습되고 사용되고 선호될 수 있는 소프트웨어 제품의 능력

4) 효율성

명시된 조건에서 사용되는 자원의 양에 따라 요구된 성능을 제공하는 소프트웨어 제품 능력.
효율성의 품질부특성은 시간반응성, 자원효율성, 준수성 등이 있다.

5) 이식성

한 환경에서 다른 환경으로 전이될 수 있는 소프트웨어 제품의 능력.
이식성의 품질부특성은 적응성, 설치성, 공존성, 대체성, 준수성 등이 있다.

6) 유지보수성

소프트웨어 제품의 변경되는 능력. 환경과 요구사항 및 기능적 명세에 따른 소프트웨어의 수정, 개선, 혹은 개작 등이 포함된다. 유지보수성의 품질부특성은 분석성, 변경성, 안정성, 시험성, 준수성 등이 있다.

8. 정보처리 산업기사 외부평가 필기시험[2018.09.08.]

※ 애플리케이션 배포

1. 다음 중 형상관리 도구가 아닌 것은?

- ① Maven ② CVS ③ SVN ④ Git

2. 형상관리 도구 중 무엇에 관한 설명인지 고르면?

공개 SW 커뮤니티를 중심으로 사용 저번이 많이 확대되고 있다. 중앙 집중형 방식이 아닌 분산형 방식으로 각 로컬 PC에 스스로 완전한 저장소가 구성되며, 필요에 따라 중앙 집중형 방식으로 운영할 수 있다.

- ① CVS ② Git ③ SVN ④ Cloud

3. 다음 용어를 쓰시오. (한글 2글자)

소스코드 파일들을 컴퓨터에서 실행할 수 있는 소프트웨어로 변환하는 일련의 과정을 말한다.

()

4. 형상관리도구중 Git 명령어로 옳지 않은 것은?

- ① 버전 확인 : git -version
② 커밋되지 않은 변경사항 조회 : git status
③ 스테이징에 있는 파일 커밋 : git commit
④ 브랜치명의 브랜치를 현재 브랜치로 합침 : git branch

※ 프로그래밍 언어 활용

5. 다음은 자바의 기본 자료형의 분류와 이름이다. 적절한 것끼리 바르게 연결하시오.

- ① 정수형 • • ㉠ boolean
② 논리형 • • ㉡ short
③ 문자형 • • ㉢ char

6. 자바의 주석 처리 방법 중 코드의 특정한 부분을 사용하고 싶지 않을 때나 여러 줄에 걸쳐 주석을 달고 싶을 때 사용하는 표기법을 쓰시오.

()

7. 다음에서 설명하는 것은 무엇인지 쓰시오.

웹 애플리케이션과 서버 환경을 만들어 동작시키는 기능을 제공하는 소프트웨어 프레임워크이다. 인터넷상에서 HTTP를 통해 사용자 컴퓨터나 장치에 애플리케이션을 수행해주는 미들웨어(소프트웨어 엔진)으로 볼 수 있다.

()

8. 다음 중 무반반복이 실행되지 않는 것은?

- ① while(100) ② while(-1)
③ while(0) ④ for(;;)

9. 다음 중 자바의 기본 타입이 아닌 것은?

- ① String ② byte
③ char ④ boolean

10. 다음 실행문의 출력결과에 오류가 없으면 O, 오류가 있으면 X로 표기하시오.

```
int i = 21;
if(i > 20){
    int k = 7;
}
System.out.print(k);
```

()

11. 자바 프로그램의 종류 중 웹 서버 프로그램을 작성하기 위한 프로그램은?

- ① 애플릿(applet)
② 서블릿(servlet)
③ 안드로이드(android)
④ 자바스크립트

12. 증감연산자의 연산식을 보고 결과를 옳게 연결한 것은?

```
a=10;  
b = ++a + 10;
```

- ① $a = 11$ $b = 21$
 ② $a = 11$ $b = 20$
 ③ $a = 9$ $b = 19$
 ④ $a = 9$ $b = 20$

※ 응용SW기초기술활용

13. 다음 FTP에 대한 설명이 옳으면 O표, 틀리면 X표를 하시오.

TCP/IP 네트워크 상에서 컴퓨터들이 파일을 교환하기 위해 1971년에 최초로 공개된 통신 규약이다. 네트워크에 연결된 컴퓨터끼리 데이터를 원활하게 교환하기 위한 목적으로 개발되었다.

()

14. 리눅스/유닉스 기본 명령어를 쓰시오(단답형)

파일의 모든 내용을 한 번에 화면에 출력시키는 명령어

()

15. 다음은 단답형 문항입니다. 다음 메모리관리 기법에서 설명하는 것은 무엇인지 쓰시오




주 기억장치를 몇 개의 영역으로 고정 분할시키고, 각 작업에게 가장 적당한 하나의 분할 영역을 할당해 주는 방법이었다. 이 기법에서 분할된 영역의 개수 및 각 영역의 크기는 해당 운영체제를 생성할 때 설정되어 그 운영체제가 사용되는 동안에는 그대로 유지된다.

()

16. 다음 중 시스템 프로그램이 아닌 것은?

- | | |
|-------|----------|
| ① OS | ② DBMS |
| ③ JVM | ④ Editor |

17. ER 다이어그램 표준 기호를 연결하시오.

- | | | | | |
|------|---|---|---|---|
| ① 개체 | • | • | ㉠ |  |
| ② 속성 | • | • | ㉡ |  |
| ③ 관계 | • | • | ㉢ |  |
| ④ 연결 | • | • | ㉣ | |

18. 네트워크 명령어를 쓰시오. (단답형)

원격지 호스트와의 연결 상태를 확인하는 명령어

()

19. 메모리관리기법에 대한 내용입니다. 빈칸을 채우시오

페이징 기법에서는 가상메모리를 같은 크기의 단위로 분할했으나, [] 기법에서는 가상메모리를 서로 다른 논리적 []로 분할하고 메모리를 할당하며 주소 변환을 하게 됩니다.

()

20. 프로세스 스케줄링 기법 중 선점형 스케줄링에 속하는 것은?

- ① 라운드로빈
- ② 선착순 처리(FCFS)
- ③ 최단작업우선(SJF)
- ④ 우선순위 스케줄링





21. DBMS의 종류를 고르시오.

마이크로소프트사에서 개발한 상업용 데이터베이스로 다른 운영체제도 사용가능하지만 윈도우즈에 특히 특화되어 있다. 비공개 소스로 폐쇄적인 정책을 사용하고 있다. 리눅스버전은 오픈소스이다. 비교적 중소기업에서 주로 사용한다.

- ① 오라클 ② MY SQL
③ MS SQL ④ DB2

22. ER다이아그램의 기호를 고르시오.

개체(엔티티)는 분리된 물체 하나를 표현한다.
엔티티는 명사 하나에 해당한다고 생각하면 된다.

- ①  ② 
- ③  ④ 

※ 화면구현

23. 몸짓, 말소리만으로 IT 기기를 사용할 수 있는 사용자 인터페이스를 고르면?

- ① CLI ② GUI
③ NUI ④ OUI

24. HTML의 레이아웃 관련 태그 중 독립적인 콘텐츠 항목에 대한 영역을 지정할 때 사용하는 것은?

()

25. 다음은 화면 설계에 대한 설명이다. 옳은 것은?

기획 단계에서 페이지 레이아웃이나 화면 이동, 구성요소에 대한 내용을 기술한 문서이다.

- ① 목업 ② 스토리보드
③ 프로토타입 ④ 와이어프레임

26. 유스케이스에 명세서에 포함되는 내용으로 옳지 않은 것은?

	유스케이스 명	작성자
①	개요	유스케이스의 전체적인 개요가 되는 내용 기술
②	관련 액터	유스케이스 간의 관계 (예. 수강신청)
③	우선순위	해당 유스케이스의 중요도를 표시
④	흐름(시나리오)	[기본] 오류가 없는 정상적인 흐름을 기록

27. 웹의 3 요소 중 나머지 하나를 쓰시오.

1.
2. 웹 접근성(Web Accessibility)
3. 웹 호환성(Cress Browsing)

()

※ SQL활용

28. DML에 대한 연결을 옳게 연결하시오.

- | | | | |
|----------|---|---|------------------------|
| ① Grant | • | • | ㉠ 데이터베이스 사용
자 권한 부여 |
| ② Update | • | • | ㉡ 데이터 수정 |
| ③ Create | • | • | ㉢ 데이터베이스 오브
젝트 생성 |

29. Dept는 학과 컬럼이다. 기계과, 전산과, 전자과에 학생 20명씩 재학 중이다. 다음 쿼리문을 실행했을 때 튜플의 개수는?

```
select distinct dept from student
```

()

30. 다음을 설명을 보고 용어를 쓰시오.

DCL의 조작 대상 중 동시에 다수의 작업을 독립적으로 안전하게 처리하기 위한 상호 작용 단위이다.

()

31. 다음 SQL 명령문의 의미로 가장 적절한 것은?

DROP TABLE 학과 CASCADE;

- ① 학과 테이블을 제거하십시오.
- ② 학과 필드를 제거하십시오.
- ③ 학과 테이블과 이 테이블을 참조하는 다른 테이블도 함께 제거하십시오.
- ④ 학과 테이블이 다른 테이블에 참조 중이면 제거하지 마십시오.

32. 다음 중 DDL에 해당하는 것이 아닌 것은?

- ① CREATE ② ALTER
③ DROP ④ SELECT

※ 애플리케이션테스트수행

33. 결합은 크게 4가지 유형으로 분류된다. 다음 설명에 해당하는 결합을 고르면?

비정상적인 종료/중단, 응답 시간 지연, 데이터베이스 여러 등 주로 애플리케이션 환경과 데이터베이스 처리에서 발생하는 결함을 말한다.

- ① 시스템결합 ② 기능결합
③ GUI결합 ④ 문서결합

34. 단위 테스트 방법 중 프로그램 내부 구조 및 복잡도를 검증하는 테스트는?

- ① 알파테스트 ② 베타테스트
③ 화이트박스 테스트 ④ 블랙박스 테스트

35. 다음 설명을 읽고 프로젝트 수행 단계에 따른 테스트 종류를 쓰시오.

개발 프로젝트 차원에서 정의된 전체 시스템의 동작과 관련되어 있다. 환경 제한적 장애 관련 리스크를 최소화하기 위하여 실제의 최종 사용자 환경과 유사하게 시스템 성능, 관련된 고객의 기능, 비기능적인 요구사항 등이 완벽하게 수행되는지를 테스트하며, 이때 요구사항 명세서, 비즈니스 절차, 유스케이스 리스크 분석 결과 등을 이용한다.

()

36. 소프트웨어 생명주기의 V모델에서 테스트 매니저의 순서가 옳게 연결된 것은?

- ① 단위-통합-시스템-인수
- ② 통합-단위-인수-시스템
- ③ 인수-시스템-통합-단위
- ④ 시스템-인수-단위-통합

37. 시스템 테스트 기법 중 동시접속으로 시스템에 많은 요청이 발생될 때 어떻게 작동되는지 확인하는 테스트는?

- ① 부하 및 성능 테스트
- ② 장애 복구 테스트
- ③ 보안 테스트
- ④ 인수테스트

※ UI 테스트

38. 다음 설명을 읽고 해당되는 테스트 기법을 고르면?

사용성에 대한 문제를 찾아내기 위한 사용성 공학 방법으로 전문가에 의해 이론과 경험을 근거로 하여 일련의 규칙을 만들어 놓고 평가 대상이 그러한 규칙들을 얼마나 잘 지키고 있는가를 확인하는 평가 방법이다.

- ① 휴리스틱 평가 ② 페이퍼 프로토타입
③ 선호도평가 ④ 성능평가

39. 다음에 해당하는 용어를 쓰시오.

사용자에게 최적의 경험을 제공하는 기본 단위로서, 인터페이스, 인터랙션, 경험까지 포괄하는 전반적인 상호작용을 말하며, 시스템 디자인에서 인터랙션은 시스템의 행동적인 측면에 초점을 맞추는 상대적으로 작은 개념을 말한다.

()

40. 사용성 평가가 기존의 소프트웨어 테스트와 다른 점은?

- ① 요구사항 정의에 기반한다.
- ② 기능 위주의 보완에 초점을 맞춘다.
- ③ 개발자에 기반한 시나리오를 구성한다.
- ④ 정황요소를 고려한 태스크 기반의 개선을 다룬다.

9. 정보처리 산업기사 외부평가 필기시험(2018.10.13.)

※ 애플리케이션 배포

1. 소프트웨어 빌드 도구인 Maven에 대한 특징으

로 옳지 않은 것은?

- ① 원격저장소 사용
- ② 틀에 대한 이식성과 통합
- ③ 디펜던시(dependency) 관리
- ④ 빌드 로직의 전역적인 재사용 어려움

2. git의 저장소 3가지가 아닌 것은?

- ① 커밋한 소스가 보관되는 저장소
- ② 현재 프로젝트 파일들이 있는 작업트리
- ③ 저장소와 작업트리 사이의 버퍼영역
- ④ 실패한 파일들이 보관되는 임시저장소

3. 다음 용어를 쓰시오.

소프트웨어 설계에서 기능단위로 분해하고 추상화하여 재사용 및 공유 가능한 수준으로 만들어진 단위

()

4. git명령어 중 현재 디렉토리에 git 저장소를 생성하는 명령어를 쓰시오(단답형)

()

※ 프로그래밍 언어 활용

5. 데드락이 걸리는 조건으로 옳지 않은 것은?

- ① 상호배제(Mutual Exclusion)
- ② 점유와 대기(Hold and Wait)
- ③ 선점(preemption)
- ④ 환형 대기(Circular Wait)

6. 자바의 변수 이름 작성 규칙으로 옳지 않은 것은?

- ① 대소문자를 구분한다
- ② 언더바(_)는 사용할 수 있다.
- ③ 첫 문자로 숫자는 올 수 있다.
- ④ 자바 예약어는 사용할 수 없다.

7. 자바 멤버변수의 접근 권한을 연결하십시오.

- | | | | |
|-------------|---|---|----------------|
| ① private | • | ㉠ | 자신 클래스 안에서만 사용 |
| ② protected | • | ㉡ | 모든 클래스에서 사용 |
| ③ public | • | ㉢ | 하위 클래스에서 접근 가능 |

8. 다음 용어를 쓰시오.

고급 언어로 쓰인 프로그램이 컴퓨터에서 수행되기 위해서는 컴퓨터가 직접 이해할 수 있는 언어로 바꾸어 주어야 한다.

()

9. OX로 표기하십시오

객체는 인스턴스라 불리고 객체를 실체화한다.

()

10. 다음 실행문의 결과를 쓰시오.

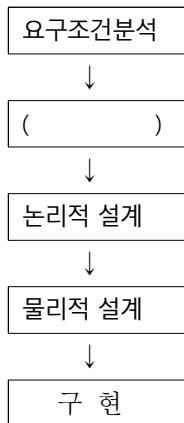
```
int a = 10;
if(a%2==0)
    System.out.println("짝수");
else
    System.out.println("홀수");
```

()

21. OSI 7계층 참조 모델에서 인접 개방형 시스템 간의 데이터 전송, 에러 검출, 오류 회복 등을 취급하는 계층은?

- ① 물리적 계층 ② 데이터 링크 계층
③ 응용계층 ④ 세션 계층

22. 데이터베이스 설계 순서의 빈칸을 채우시오.



※ 화면구현

23. 웹 페이지 전체의 일관성을 유지할 수 있도록 스타일을 미리 저장해둔 시트는?

()

24. UI구현 지침에 관한 내용이다. 빈칸을 채우시오.

월드 와이드 웹을 위한 표준을 개발하고 장려하는 국제적인 컨소시엄 조직으로, 설립 취지는 웹의 지속적인 성장을 도모하고 프로토콜과 가이드라인을 개발하여 월드 와이드 웹의 모든 잠재력을 이끌어 낸다는 목적에서 찾을 수 있다.

()

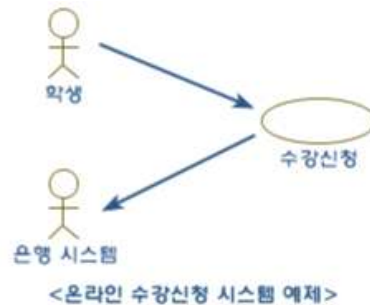
25. HTML의 이벤트 처리 중 해당 페이지와 해당 페이지에 연결된 외부 내용들이 완전히 로딩되었을 때 구동되는 것은?

- ① onclick ② onmousedown
③ onchange ④ onload

26. HTML이 기본태그 중 사용자에게 보여주는 실제 내용이 구현되는 부분은?

- ① `<html>...</html>`
- ② `<head>...</head>`
- ③ `<title>...</title>`
- ④ `<body>...</body>`

27. 다음은 유스케이스 다이어그램의 예이다. 위
그림에서 액터를 고르면?



- ① 학생
- ② 은행시스템
- ③ 학생과 은행시스템
- ④ 수강신청

※ SQL활용

28. 다음 중 테이블 생성 과정에 적용할 수 있는 제약조건이 아닌 것은?

- ① INDEX ② NOT NULL
③ PRIMARY KEY ④ FOREIGN KEY

29. 다음 중 내부조인이 아닌 것은?

- ① 동등 조인 ② 자연 조인
③ 교차 조인 ④ 왼쪽 외부 조인

30. 다음 SQL 검색문의 의미로 가장 적절한 것은?

```
SELECT DISTINCT 학과명 FROM 학생;
```

- ① 학생 테이블의 학과명을 모두 검색하라.
- ② 학생 테이블의 학과명을 중복되지 않게 모두 검색하라.
- ③ 학과 테이블의 학과명 중에서 중복된 학과명을 모두 검색하라.
- ④ 학생 테이블을 학과명 구별하지 말고 모두 검색하라.

31. 테이블 생성 과정에서 사용할 수 있는 제약 조건 중 테이블 내에서 동일한 값을 가져서는 안 되는 항목에 지정하는 것은?
()

32. 다음 SQL 명령문의 의미로 가장 적절한 것은?

DROP TABLE 성적 CASCADE;

- ① 성적 테이블과 이 테이블을 참조하는 다른 테이블도 함께 제거하시오.
- ② 성적 테이블이 다른 테이블에 의해 참조 중이면 제거하지 마시오.
- ③ 성적 테이블만 제거하시오.
- ④ 성적 테이블의 인덱스만 제거하시오.

※ 애플리케이션테스트수행

33. 인수 테스트 종류 중 개발하는 조직 내 잠재 고객에 의해 테스트를 수행하는 방법은?
()

34. 소프트웨어 생명주기의 V모델에서 테스트 매니저의 순서를 쓰시오.
()

35. 결함관련 용어 중 다음 설명으로 옳은 것은?

소프트웨어 개발 또는 유지 보수 수행 중에 발생한 부정확한 결과로, 개발자의 실수로 발생한 오타, 개발 명세서의 잘못된 이해, 서브루틴의 기능 오해등이 있다.

- ① 에러(Error) ② 오류(Fault)
- ③ 실패(Failure) ④ 결함(Defect)

36. 결함은 크게 4가지로 구분된다. 시스템결함, (), GUI 결함, 문서결함이 다. 빈칸에 들어갈 값으로 옳은 것은?

37. 부하 및 성능 테스트에서 동시 이용자 수를 계산하는 공식으로 옳은 것은?
① $T P S * \text{호출간격}(\text{응답시간} + \text{대기시간})$
② $T P S * \text{응답시간}$
③ $T P S + \text{대기시간}$
④ $T P S + \text{호출간격}(\text{응답시간} * \text{대기시간})$

※ UI 테스트

38. 다음 설명을 읽고 무엇에 대한 설명인지 쓰시오.

사용자 측면에서의 요구사항이며 주로 기능 개선에 대한 내용으로, 사용자가 원하는 목표를 위하여 시스템에서 수행해야 하는 내용을 기술

()

39. HTML 구조 중 Style과 Script 태그가 위치해야 할 위치는?

- ① `<!DOCTYPE html>`
- ② `<head>...</head>`
- ③ `<title>...</title>`
- ④ `<body>...</body>`

40. 다음 CSS 선택자 중 클래스 선택자를 활용하면 특정 HTML 요소들을 그룹화하여 스타일을 지정하는 것이 가능한 것은?

- ① 공용 선택자
- ② 태그 선택자
- ③ 클래스 선택자
- ④ 아이디 선택자

10. 정보처리 산업기사 외부평가 구술시험(1,2차)

※ 애플리케이션 배포

1. 형상관리를 왜 하는가?

※ 프로그래밍 언어 활용

2. 개발도구를 어느데로 말하시오.
3. 알고리즘을 쓰는 이유를 설명하시오.
4. JSP의 뜻을 말하시오.
5. 디버깅종류를 설명하시오.

※ SQL 활용(4문제)

6. DCL 명령어를 어느데로 말하시오.
7. 테이블 설계 시 규칙이나 방법을 설명하시오.
8. 데이터베이스 정규화에 대해서 설명하시오.
9. 무결성 제약조건에 대해서 설명하시오.
10. 데이터베이스 구조를 설명하시오.

※ 애플리케이션테스트수행

11. 인스펙션에 대해서 설명하시오.
12. 워크스루에 대해서 설명하시오.
13. 디버그할 때 비파괴형식과 파괴형식의 차이점을 설명하시오.
14. 알파테스트와 베타 테스트 차이

※ UI 테스트(3문제)

15. 심층인터뷰와 포커스 인터뷰에 대해서 설명하시오.
16. UX디자인

11. 정보처리 산업기사 외부평가 실기시험(2018.09.09.)

가. 과제 개요

- 1) 본 과제는 학생 성적 조회 사이트를 만드는 프로그램이다. 본 과제에서는 학생 조회, 학생 성적 확인 및 학생의 성적 기입 등이 가능하도록 프로그램을 제작하시오.
- 2) 프로그램 개발을 위해 개발 환경을 확인하고, 필요한 설정을 수행하시오.

나. 업무 요건

데이터베이스에 교과목과 강사 테이블을 생성하여야 한다. 테이블의 데이터는 샘플 데이터를 참조하여 테스트를 실시하여야 한다.

다. 프로젝트 준비

- 1) 접속에 사용할 오라클 계정은 'system'이고, 암호는 '1234'이다.
- 2) 이클립스(eclipse)의 작업영역(workspace)은 'c:\Wexamkorea'를 사용한다.
- 3) 프로젝트를 생성하기 전에 java, jsp, html, css, text 파일의 기본 인코딩을 'UTF-8'로 지정한다.
(이클립스 Window - Preference 메뉴)
- 4) 이클립스에서 톱캣을 연동하여 실행하기 위한 설정을 수행해야 한다.
- 5) 오라클 관리를 위해 8080 포트를 사용하고 있기 때문에, 톱캣 서버는 8090 포트를 사용하도록 권장한다.
- 6) 프로젝트 유형은 'Dynamic Web Project'을 생성하고, 프로젝트 이름은 'HRD_비번호'를 사용한다.
(비번호는 수험자가 부여받은 번호를 사용한다.)
- 7) 시험 후 이클립스 작업영역(workspace), 즉 'c:\Wexamkorea' 디렉토리를 반드시 'sw_비번.zip'으로 압축해서 제출해야 한다.

[참고] 아래 소스는 Oracle DataBase Connection 함수이다.

```
package DBPKG;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public static Connection getConnection() throws Exception{
    Class.forName("oracle.jdbc.OracleDriver");
    Connection con = DriverManager.getConnection
        ("jdbc:oracle:thin:@//localhost:1521/xes","system","1234");
    return con;
}
```

라. 수행작업

1) 메인화면은 다음과 같은 디자인을 참고하여 작성하시오.

학생 성적 조회 프로그램 ver 1.0				<header>
학생조회	학생성적입력	학생성적조회	<u>홈으로</u>	<nav>
<p>학생성적 관리 프로그램</p> <p>학생과 학생 성적에 관한 데이터베이스를 구축하고 관리하기 위한 프로그램이다.</p> <p>1. 학생정보 및 학생성적 테이블을 생성하고 샘플데이터를 추가한다. 2. 학생조회, 학생성적 입력, 학생성적 조회 프로그램을 작성한다. 3. 올바르게 구현되었는지 확인한다.</p>				<section>
Copyright @ 2018 All right reserved Hanyang High School				<footer>

2) 데이터 입출력 요건에 맞게 테이블을 생성하시오.

가) 학생 테이블 (테이블 명 : member_tbl)

순서	컬럼ID	컬럼명	형태	길이	NULL	비고
1	id	학번	varchar2	5	NOT NULL	Primary Key
2	name	이름	varchar2	20		
3	gender	성별	Varchar2	3		

나) 학생 점수 테이블(테이블 명 : score_tbl)

순서	컬럼ID	컬럼명	형태	길이	NULL	비고
1	id	학번	Varchar2	5	NOT NULL	Primary Key
2	korean	국어	number	3		
3	english	영어	number	3		
4	math	수학	number	3		
5	history	역사	number	3		

3) 샘플 데이터를 SQL 문장을 사용하여 각각 생성된 테이블에 추가 입력 하시오.

가) 학생정보 샘플 데이터]

학번	이름	성별
10301	권기현	남
10302	구지우	여
10303	권태준	남
10304	금기연	여
10305	김영진	남
10401	김주현	남
10402	김태민	여
10403	임수민	여
10404	남상도	남
10405	조은호	여

나) 학생 점수 데이터

번호	국어	영어	수학	역사
10301	70	80	75	95
10302	80	73	80	85
10303	45	75	45	75
10304	65	73	65	65
10305	90	100	68	95
10401	95	75	84	60
10402	65	70	50	94
10403	80	60	100	80
10404	64	50	40	65
10405	55	70	80	75

4) 화면별 업무 요구사항 및 화면 구성 요건에 맞게 화면을 구현하시오.

[참고사항]

- 화면의 구성요소는 필수 사항이다.
- 화면의 색깔, 폰트 등 스타일 구성요소는 선택사항이다.

가) 시작화면(index.jsp)

- ① 시작화면은 헤더, 메뉴, 섹션, 푸터로 구성된다. 메뉴는 ‘학생조회’, ‘학생성적입력’, ‘학생성적조회’, ‘홈으로’로 구성된다.
- ② 푸터(footer)는 저작권 관련 정보로 구성된다.

학생 성적 조회 프로그램 ver 1.0

학생조회 학생성적등록 학생성적조회 홈으로

학생성적 관리 프로그램

학생과 학생 성적에 관한 데이터베이스를 구축하고 관리하기 위한 프로그램이다.

1. 학생정보 및 학생성적 테이블을 생성하고 샘플데이터를 추가한다.
2. 학생조회, 학생성적 입력, 학생성적 조회 프로그램을 작성한다.
3. 올바르게 구현되었는지 확인한다.

Copyright @ 2018 All right reserved Hanyang High School

나) 학생 조회 화면

- ① 학생 조회를 클릭 했을 때의 화면이다.
- ② 데이터베이스를 연동하여 학번, 이름, 성별을 불러와서 테이블로 만든다.
- ③ 단. 학생에 따른 학년, 반, 번호도 같이 테이블에 기재하여야 한다.

학생 성적 조회 프로그램 ver 1.0

학생조회 학생성적등록 학생성적조회 홈으로

학생 조회

학번	이름	학년	반	번호	성별
10301	권기현	1	03	01	남
10302	구지우	1	03	02	여
10303	권태준	1	03	03	남
10304	금기연	1	03	04	여
10305	김영진	1	03	05	남
10401	김주현	1	04	01	남
10402	김태민	1	04	02	여
10403	임수민	1	04	03	여
10404	남상도	1	04	04	남
10405	조은호	1	04	05	여

Copyright @ 2018 All right reserved Hanyang High School

다) 학생성적등록 화면

- ① 학생성적 입력을 클릭 했을 때의 화면이다.
- ② 학번, 국어, 수학, 영어, 역사 항목을 각각 입력한 뒤 등록 버튼을 누르면 학생 점수 테이블 데이터베이스에 입력되어야 한다.

학생 성적 조회 프로그램 ver 1.0

[학생조회](#) [학생성적등록](#) [학생성적조회](#) [홈으로](#)

학생 성적 등록

학번

국어

수학

영어

역사

Copyright © 2018 All right reserved Hanyang High School

라) 학생성적조회 화면

- ① 다음과 같이 데이터베이스를 연동하여 학번, 이름, 학년, 반, 번호, 성별, 국어, 수학, 영어, 역사, 합계 및 평균을 구한 뒤 총 합계, 총 평균을 구하여 테이블로 만들어 화면에 출력하라.

학생 성적 조회 프로그램 ver 1.0

[학생조회](#) [학생성적입력](#) [학생성적조회](#) [홈으로](#)

학생성적 조회

학번	이름	학년	반	번호	성별	국어	수학	영어	역사	합계	평균	순위
10404	남상도	1	04	04	남	64	40	50	65	219	54.75	1
10303	권태준	1	03	03	남	45	45	75	75	240	60	2
10304	금기연	1	03	04	여	65	65	73	65	268	67	3
10402	김태민	1	04	02	여	65	50	70	94	279	69.75	4
10405	조은호	1	04	05	여	55	80	70	75	280	70	5
10401	김주현	1	04	01	남	95	84	75	60	314	78.5	6
10302	구지우	1	03	02	여	80	80	73	85	318	79.5	7
10301	권기현	1	03	01	남	70	75	80	95	320	80	8
10403	임수민	1	04	03	여	80	100	60	80	320	80	9
10305	김영진	1	03	05	남	90	68	100	95	353	88.25	10
						국어 총합	수학 총합	영어 총합	역사 총합	총합	총평균	-----
						709	687	726	789	2911.0	291.1	-----

Copyright © 2018 All right reserved Hanyang High School

12. 정보처리 산업기사 외부평가 실기시험(2018.10.14.)

가. 과제 개요

- 1) 본 과제는 매장별 커피 판매현황을 관리하기 위한 프로그램이다. 본 과제에서는 판매등록, 판매현황, 매장별 판매액, 상품별 판매액 확인 및 등록이 가능하도록 프로그램을 제작하시오.
- 2) 프로그램 개발을 위해 개발 환경을 확인하고, 필요한 설정을 수행하시오.

나. 업무 요건

- 1) 데이터베이스에 테이블을 생성하여야 한다. 테이블의 데이터는 샘플 데이터를 참조하여 테스트를 실시하여야 한다.

다. 프로젝트 준비

- 1) 접속에 사용할 오라클 계정은 'system'이고, 암호는 '1234'이다.
- 2) 이클립스(eclipse)의 작업영역(workspace)은 'c:\WcoffeeManage'를 사용한다.
- 3) 프로젝트를 생성하기 전에 java, jsp, html, css, text 파일의 기본 인코딩을 'UTF-8'로 지정한다. (이클립스 Window - Preference 메뉴)
- 4) 이클립스에서 톱캣을 연동하여 실행하기 위한 설정을 수행해야 한다.
- 5) 오라클 관리를 위해 8080 포트를 사용하고 있기 때문에, 톱캣 서버는 8090 포트를 사용하도록 권장한다.
- 6) 프로젝트 유형은 'Dynamic Web Project'을 생성하고, 프로젝트 이름은 'Coffee_비번호'를 사용한다. (비번호는 수험자가 부여받은 번호를 사용한다.)
- 7) 시험 후 이클립스 작업영역(workspace), 즉 'c:\WcoffeeManage' 디렉토리를 반드시 'sw_비번.zip'으로 압축해서 제출해야 한다.

[참고] 아래 소스는 Oracle DataBase Connection 함수이다.

```
package DBPKG;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public static Connection getConnection() throws Exception{
    Class.forName("oracle.jdbc.OracleDriver");
    Connection con = DriverManager.getConnection
        ("jdbc:oracle:thin:@//localhost:1521/x", "system", "1234");
    return con;
}
```

라. 수행작업

1) 메인화면은 다음과 같은 디자인을 참고하여 작성하시오.

매장별 커피 판매관리 ver 1.0					<header>
판매등록	판매현황	매장별판매액	상품별판매액	<u>홈으로</u>	<nav>
<p style="text-align: center;">매장별 커피 판매관리 프로그램</p> <p>매장별 커피 판매를 관리하기 위한 프로그램이다.</p> <p>1. 상품테이블, 매장테이블, 판매테이블을 추가한다.</p> <p>2. 판매등록, 판매현황, 매장별판매액, 상품별판매액 페이지를 작성한다.</p> <p>3. 올바르게 구현되었는지 확인한다.</p>					<section>
Copyright @ 2018 All right reserved Hanyang High School					<footer>

2) 데이터 입출력 요건에 맞게 테이블을 생성하시오.

가) 상품 테이블 (테이블 명 : tbl_product_01)

순서	컬럼ID	컬럼명	형태	길이	NULL	비고
1	pcode	상품코드	varchar2	10	NOT NULL	Primary Key
2	name	상품명	varchar2	20		
3	cost	금액	number	10		

나) 매장 테이블(테이블 명 : tbl_shop_01)

순서	컬럼ID	컬럼명	형태	길이	NULL	비고
1	scode	매장코드	varchar2	10	NOT NULL	Primary key
2	sname	매장이름	varchar2	20		

다) 판매 테이블(테이블 명 : tbl_salelist_01)

순서	컬럼ID	컬럼명	형태	길이	NULL	비고
1	saleno	판매번호	number	10	NOT NULL	Primary key
2	pcode	판매코드	varchar2	10	NOT NULL	
3	saledate	판매일	date			
4	scode	매장코드	varchar2	10	NOT NULL	
5	amount	수량	number	10		

3) 샘플 데이터를 SQL 문장을 사용하여 각각 생성된 테이블에 추가 입력 하시오.

가) 상품정보 데이터

상품코드	이름	금액
AA01	아메리카노	3000
AA02	에스프레소	3500
AA03	카페라떼	4000
AA04	카라멜마끼	4500
AA05	카푸치노	5000
AA06	초코롤케익	6000
AA07	녹차롤케익	6500
AA08	망고쥬스	7000
AA09	핫초코	2500

나) 매장 데이터

매장코드	매장이름
S001	강남점
S002	강서점
S003	강동점
S004	강북점
S005	동대문점
S006	인천점

다) 판매데이터

판매번호	판매코드	판매일	매장코드	수량
100001	AA01	20180902	S001	50
100002	AA03	20180902	S002	40
100003	AA04	20180902	S002	20
100004	AA04	20180902	S001	30
100005	AA05	20180902	S004	40
100006	AA03	20180902	S004	30
100007	AA01	20180902	S003	40
100008	AA04	20180902	S004	10
100009	AA01	20180902	S003	20
100010	A005	20180902	S003	30
100011	AA01	20180902	S001	40
100012	AA03	20180902	S002	50
100013	AA04	20180902	S002	50
100014	AA05	20180902	S004	20
100015	AA01	20180902	S003	30

4) 화면별 업무 요구사항 및 화면 구성 요건에 맞게 화면을 구현하시오.

[참고사항]

- 화면의 구성요소는 필수 사항이다.
- 화면의 색깔, 폰트 등 스타일 구성요소는 선택사항이다.

가) 시작화면(index.jsp)

- ① 시작화면은 헤더, 메뉴, 섹션, 푸터로 구성된다. 메뉴는 '판매등록', '판매현황', '매장별판매액', '상품별판매액', '홈으로'로 구성된다.
- ② 푸터(footer)는 저작권 관련 정보로 구성된다.

매장별 커피 판매관리 ver 1.0				
판매등록	판매현황	매장별판매액	상품별판매액	<u>홈으로</u>
<p>매장별 커피 판매관리 프로그램</p> <p>매장별 커피 판매를 관리하기 위한 프로그램이다.</p> <ol style="list-style-type: none"> 1. 상품테이블, 매장테이블, 판매테이블을 추가한다. 2. 판매등록, 판매현황, 매장별판매액, 상품별판매액 페이지를 작성한다. 3. 올바르게 구현되었는지 확인한다. 				
Copyright © 2018 All right reserved Hanyang High School				

나) 판매등록 화면

매장별 커피 판매관리 ver 1.0																
<u>판매등록</u>	매출액확인	매장별총판매액	상품별총판매액	<u>홈으로</u>												
<p>판매등록</p> <div style="display: flex; justify-content: center; align-items: center;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td style="padding: 5px;">비번호</td><td style="width: 150px; height: 25px;"></td></tr> <tr><td style="padding: 5px;">상품코드</td><td style="width: 150px; height: 25px;"></td></tr> <tr><td style="padding: 5px;">판매날짜</td><td style="width: 150px; height: 25px;"></td></tr> <tr><td style="padding: 5px;">매장코드</td><td style="width: 150px; height: 25px;"></td></tr> <tr><td style="padding: 5px;">판매수량</td><td style="width: 150px; height: 25px;"></td></tr> <tr> <td style="padding: 5px; width: 50px;"> <div style="display: flex; justify-content: space-between; padding: 0 5px;"> 등록 다시쓰기 </div> </td> <td style="width: 150px; height: 25px;"></td> </tr> </table> </div>					비번호		상품코드		판매날짜		매장코드		판매수량		<div style="display: flex; justify-content: space-between; padding: 0 5px;"> 등록 다시쓰기 </div>	
비번호																
상품코드																
판매날짜																
매장코드																
판매수량																
<div style="display: flex; justify-content: space-between; padding: 0 5px;"> 등록 다시쓰기 </div>																
Copyright © 2018 All right reserved Hanyang High School																

다) 판매현황 화면

매장별 커피 판매관리 ver 1.0

판매등록 판매현황 매장별총판매액 상품별총판매액 홈으로

판매현황

비번호	상품코드	판매날짜	매장코드	상품명	판매수량	총판매액
100001	AA01	2018-09-02	S001	아메리카노	50	150,000
100002	AA03	2018-09-02	S002	카페라떼	40	160,000
100003	AA04	2018-09-02	S002	카라멜마끼	20	90,000
100004	AA04	2018-09-02	S001	카라멜마끼	30	135,000
100005	AA05	2018-09-02	S004	카푸치노	40	200,000
100006	AA03	2018-09-02	S004	카페라떼	30	120,000
100007	AA01	2018-09-02	S003	아메리카노	40	120,000
100008	AA04	2018-09-02	S004	카라멜마끼	10	45,000
100009	AA01	2018-09-02	S003	아메리카노	20	60,000
100010	AA05	2018-09-02	S003	카푸치노	30	150,000
100011	AA01	2018-09-02	S001	아메리카노	40	120,000
100012	AA03	2018-09-02	S002	카페라떼	50	200,000
100013	AA04	2018-09-02	S002	카라멜마끼	50	225,000
100014	AA05	2018-09-02	S004	카푸치노	20	100,000
100015	AA01	2018-09-02	S003	아메리카노	30	90,000

Copyright @ 2018 All right reserved Hanyang High School

라) 매장별판매액 화면

매장별 커피 판매관리 ver 1.0		
판매등록	판매현황	매장별판매액
상품별총판매액		
홈으로		
매장별 판매액		
매장코드	매장명	매장별 판매액
S001	강남점	405,000
S002	강서점	675,000
S003	강동점	420,000
S004	강북점	465,000
Copyright @ 2018 All right reserved Hanyang High School		

마) 상품별판매액 화면

매장별 커피 판매관리 ver 1.0

판매등록 판매현황 매장별판매액 상품별판매액 홈으로

상품별 판매액

상품코드	상품명	상품별 판매액
AA01	아메리카노	540,000
AA03	카페라떼	480,000
AA04	카라멜마끼	495,000
AA05	카푸치노	450,000

Copyright @ 2018 All right reserved Hanyang High School