

(디지털컨버전스)  
공공데이터 융합 웹기반  
SW개발자 양성과정  
학 습 안 내 서

〈 훈련과정 국가직무능력표준 분류 〉

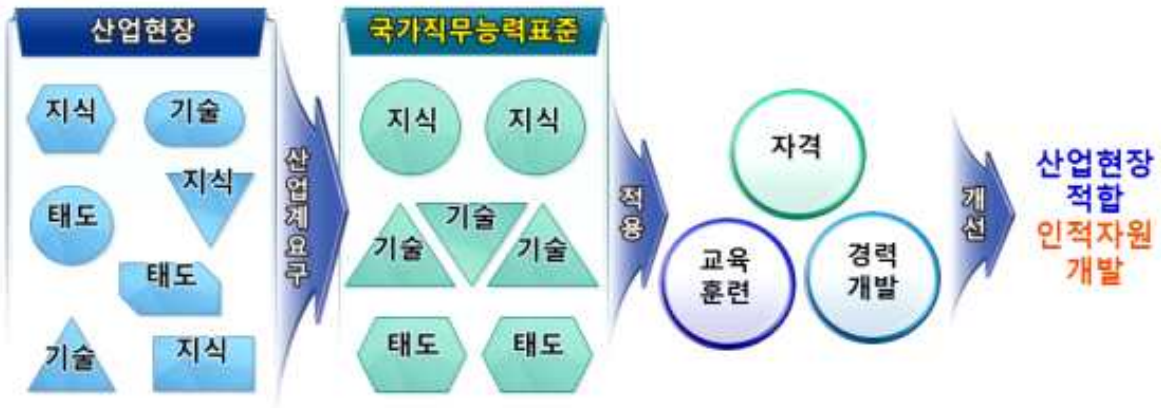
대분류	중분류	소분류	세분류
20.정보통신	01.정보기술	02.정보기술개발	02.응용SW엔지니어링
20.정보통신	01.정보기술	02.정보기술개발	04.DB엔지니어링
20.정보통신	01.정보기술	02.정보기술개발	05.NW엔지니어링
20.정보통신	01.정보기술	02.정보기술개발	07.UI/UX엔지니어링

KH정보교육원 강남지원

# NCS과정 안내

## 1. NCS의 정의

- “NCS란” 산업현장에서 직무를 수행하기 위해 요구되는 능력(지식, 기술, 태도)을 국가가 산업부문별, 수준별로 체계화한 것(자격기본법 제2조)



## 2. NCS능력단위

- 능력단위는 NCS의 세분류를 구성하는 기본단위  
 - 능력단위는 능력단위분류번호, 능력단위정의, 능력단위요소(수행준거, 지식/기술/태도), 적용범위 및 작업상황, 평가지침, 직업기초능력으로 구성

## 3. 기존훈련기준과 NCS 기반 훈련기준의 차이점

구분	기존 훈련	NCS기반 훈련
교과내용	입직자 수준의 훈련기준만 제시	1~8수준으로 설계되어 있는 능력단위를 활용하여 입직자 및 근로자 수준에 따라 필요한 교육훈련 실시
	350시간 이상의 장기과정에 대한 훈련기준만 제시	필요한 능력단위를 선택하여 다양한 훈련과정 편성 가능
훈련교과목	훈련교과목별 훈련목표 및 세부훈련내용 미 제시 - 훈련단위명만 제시	능력단위별 훈련목표 및 능력단위요소별로 수행준거 (~할 수 있다)를 통해 훈련할 내용을 구체적으로 제시
	훈련직종에 필요한 교과내용만 명시	훈련이수체계를 통해 연관직종 교과내용(능력단위)을 함께 제시
훈련방법	강의 중심의 주입식교육 - 교수중심	수행준거 중심의 자기주도형 훈련 - 훈련생 중심

# 훈 련 과 정 개 요

## 1. 훈련과정명

- (디지털컨버전스) 공공데이터 융합 웹기반 SW개발자 양성과정

## 2. 훈련기간(시간)

- 8개월(888시간)

## 3. 훈련 수준

- 5수준

## 4. 훈련대상자

- 정보통신분야로 취업을 희망하는 자

## 5. 훈련 목표

1. 공공데이터를 분석, 활용하여 웹 융합 소프트웨어를 구현할 수 있다.
2. 프로그래밍 언어의 기초문법을 적용하고 언어의 특징과 라이브러리를 활용하여 기본 응용소프트웨어를 구현할 수 있다.
3. 네트워크 프로그램을 구현하기 위한 네트워크 개발환경을 분석하고, 단계별로 요구되는 기능을 구현하고 프로그램 오류를 디버깅하고 최적화할 수 있다.
4. 사용자의 요구사항을 파악하고 기획된 아이디어를 시각화하여 테스트할 수 있고, UI 설계 산출물과 GUI 디자인 가이드를 바탕으로 UI 구현 표준 수립과 제작을 할 수 있다.
5. 응용소프트웨어가 다루어야 하는 데이터 및 이들 간의 연관성, 제약조건을 식별하여 논리적으로 조직화 하고, 데이터가 저장될 최적화된 물리적 공간을 구성하고 데이터 조작 언어(DDL, DML, TCL, PL/SQL, JDBC)를 이용하여 구현할 수 있다.
6. 화면에 대한 요구사항을 정의된 구현지침, 제시된 UI 표준과 지침에 따라 화면을 설계할 수 있다.
7. UI 설계 산출물과 GUI 디자인 가이드를 바탕으로 UI 구현 표준을 수립하고 UI를 제작할 수 있다.
8. 애플리케이션 설계를 기반으로 개발에 필요한 환경을 구성하고, 프로그래밍 언어와 도구를 활용하여 공통모듈, 업무프로그램과 배치 프로그램을 구현할 수 있다.
9. 개발도구와 프로그래밍 언어를 사용하여 애플리케이션 모듈과 인터페이스를 구현할 수 있고, 버전 관리를 할 수 있다.
10. 업무 분석가가 수집·분석·정의한 요구사항과 현행 시스템에 대한 분석 등 요구사항 확인을 통한 상세 분석 결과와 소프트웨어 아키텍처 가이드라인 및 산출물에 의거하여

공통모듈 설계, 타 시스템 연동에 대해 상세 설계할 수 있다.

11. 모듈간의 분산이 이루어진 경우를 포함하여 단위 모듈간의 데이터 관계를 분석하고 이를 기반으로 한 메커니즘을 통해 모듈간의 효율적인 연계를 구현하고 검증할 수 있다.

12. 요구사항대로 응용소프트웨어가 구현되었는지를 검증하기 위해서 분석된 테스트케이스를 따라 테스트를 수행하여 결함을 조치할 수 있다.

## 6. 교육과정 총괄표

### ○ NCS 전공교과(총 834시간)

교과목명	능력단위명	능력단위요소	훈련시간
프로그래밍 언어 활용	프로그래밍 언어 활용 (71)	구조적 프로그래밍 언어 활용하기	24
		객체지향 프로그래밍 언어 활용하기	24
		스크립트 활용하기	23
프로그래밍 언어 응용	프로그래밍 언어 응용 (72)	언어특성 활용하기	36
		라이브러리 활용하기	36
네트워크 프로그래밍 구현	네트워크 프로그래밍 구현 (54)	개발환경 분석하기	27
		기능 구현하기	27
데이터베이스 구현	데이터베이스 구현 (30)	DBMS 설치하기	10
		데이터베이스 생성하기	10
		데이터베이스 오브젝트 생성하기	10
SQL활용	SQL활용 (30)	기본 SQL 작성하기	15
		고급 SQL 작성하기	15
SQL응용	SQL응용 (36)	절차형 SQL 작성하기	18
		응용 SQL 작성하기	18
UI 디자인	UI 디자인 (72)	GUI 디자인 콘셉트 도출하기	36
		GUI 디자인 요소 선정하기	36
UI 구현	UI 구현 (72)	UI 설계 검토하기	36
		UI 구현 표준 검토하기	36
요구사항 확인	요구사항 확인 (40)	현행 시스템 분석하기	14
		요구사항 확인하기	14
		분석모델 확인하기	12

서버프로그램 구현	서버프로그램 구현 (72)	개발환경 구축하기	18
		공통 모듈 구현하기	18
		서버 프로그램 구현하기	18
		배치 프로그램 구현하기	18
애플리케이션 설계	애플리케이션 설계 (60)	공통 모듈 설계하기	30
		타 시스템 연동설계하기	30
인터페이스 구현	인터페이스 구현 (60)	인터페이스 설계서 확인하기	20
		인터페이스 기능 구현하기	20
		인터페이스 구현 검증하기	20
통합 구현	통합 구현 (60)	연계 데이터 구성하기	20
		연계 메커니즘 구성하기	20
		내외부 연계 모듈 구현하기	20
애플리케이션 배포	애플리케이션 배포 (45)	애플리케이션 배포 환경 구성하기	12
		애플리케이션 소스 검증하기	12
		애플리케이션 빌드하기	12
		애플리케이션 배포하기	9
애플리케이션 테스트 관리	애플리케이션 테스트 관리 (60)	애플리케이션 테스트케이스 설계하기	20
		애플리케이션 통합 테스트하기	20
		애플리케이션 성능 개선하기	20

○ 비NCS 교과(실기) (총 48시간)

교과목명	단원명	세부내용	훈련시간
공공데이터 활용하기	공공데이터 이해하기	공공데이터 개요	3
		공공데이터 개방	3
	공공데이터 분석하기	서비스신청하기	3
		개발문서분석하기	3
	공공데이터 적용하기	공공데이터 웹애플리케이션 적용하기	12
		지오통합활용한 지도API활용하기	12
		공공데이터활용사례 연구를 통한 웹애플리케이션 적용하기	12

○ 비NCS 교과(이론) (총 6시간)

교과목명	단원명	세부내용	훈련시간
재량교과	재량교과 (6)	<p>[입학OT]</p> <ul style="list-style-type: none"> <li>- 교육원 및 교육과정 소개</li> <li>- 성희롱 예방 교육, 장애인 인식 개선 교육, 부정훈련 안내</li> <li>- 평가기준, 평가방법, 평가일 안내</li> <li>- 고용노동부 규정 안내 (수료기준, 출석 인정가능 사유, 훈련장려금 지급, 제적사유)</li> <li>- 강의장 내 교육용 PC 사용 안내</li> <li>- 소프트웨어 및 저작권 안내</li> <li>- 소방안전 (소화기 위치 및 사용방법, 화재 및 지진 발생 시 대피 방법, 강의장 피난 안내도 안내)</li> <li>- 강의장, 휴게공간 에티켓 안내</li> </ul>	1
		<p>[취업특강 및 면접 트레이닝]</p> <ul style="list-style-type: none"> <li>- 취업지원 프로세스 안내</li> <li>- 입사지원서 항목 설명</li> <li>- 직무설명회 (직무별 업무 내용, 직무별 로드맵, 신입사원에게 기대하는 역량, 기업별 평균 연봉)</li> <li>- 이력서 작성 요령 및 클리닉</li> <li>- 실전대비 면접 트레이닝</li> </ul>	4
		<p>[수료식]</p> <ul style="list-style-type: none"> <li>- 훈련생 인터뷰 및 단체 사진</li> <li>- 우수자 포상 및 수료증 수여</li> <li>- 사후 취업지원 프로세스 안내</li> <li>- 취업 시 제출서류 안내</li> <li>- 수료 후 한달 이내 HRD수강평 작성 안내</li> </ul>	1

## 7. 훈련이수체계

8수준	임원				
7수준	부장/ 차장				네트워크 환경 분석 네트워크 프로젝트 관리
		애플리케이션 요구사항 분석	UI/UX 계획 수립	개념데이터 모델링	
6수준	과장/ 대리	애플리케이션 리팩토링 인터페이스 설계 기능 모델링 애플리케이션 설계 정적모델 설계 동적모델 설계 소프트웨어공학 활용 소프트웨어개발 방법론 활용	UI/UX 환경 분석 사용자 리서치	데이터베이스 요구사항 분석 논리 데이터베이스 설계 물리 데이터베이스 설계 데이터베이스 성능확보 데이터 전환 설계	네트워크 토폴로지 설계 네트워크 소프트웨어 아키텍처 수립 네트워크 품질 평가 계획
5수준	사원	요구사항 확인 데이터 입출력 구현 통합 구현 정보시스템 이행 제품소프트웨어 패키징 서버프로그램 구현 인터페이스 구현 화면 설계 애플리케이션 테스트 관리	UI/UX 요구 분석 UI/UX 콘셉트 기획 UI 아키텍처 설계	데이터 전환 SQL응용	네트워크 자원관리 설계 네트워크 소프트웨어 개발 방법 수립 네트워크 프로토콜 분석 네트워크 QoS 수준 정의
4수준	사원		UI 디자인 UI/UX 가이드 제작	데이터베이스 구현	네트워크 QoS 제어 설계 네트워크 프로토콜 설계 네트워크 자원 분석 네트워크 프로그램 최적화
3수준	사원	애플리케이션 배포 화면 구현 애플리케이션 테스트 수행 프로그래밍 언어 응용 응용SW 기초 기술 활용	UI 구현	SQL활용	네트워크 프로그래밍 구현 네트워크 품질 평가 실행 프로토콜 패킷 분석
2수준	사원	프로그래밍 언어 활용 개발자 환경 구축 개발 환경 운영 지원	UI 테스트 GUI 상세디자인 UI 제작	SQL작성	
-					
수준 직종		응용SW엔지니어링	UI/UX엔지니어링	DB엔지니어링	NW엔지니어링

## 8. 훈련과정 교과목 운영 로드맵

교과구분	차수(주/월/분기/학기)							
	1월차 (120시간)	2월차 (120시간)	3월차 (120시간)	4월차 (120시간)	5월차 (120시간)	6월차 (120시간)	7월차 (120시간)	8월차 (48시간)
NCS 전공 교과	<ul style="list-style-type: none"> <li>프로그래밍 언어 활용(71)</li> <li>프로그래밍 언어 응용(48)</li> </ul>	<ul style="list-style-type: none"> <li>프로그래밍 언어 응용(24)*</li> <li>네트워크 프로그래밍 구현(54)</li> <li>데이터베이스 구현(30)</li> <li>SQL 활용(12)</li> </ul>	<ul style="list-style-type: none"> <li>SQL 활용(18)*</li> <li>SQL 응용(36)</li> <li>UI 디자인(66)</li> </ul>	<ul style="list-style-type: none"> <li>UI 디자인(6)*</li> <li>UI 구현(72)</li> <li>요구사항 확인(40)</li> <li>서버프로그램 구현(2)</li> </ul>	<ul style="list-style-type: none"> <li>서버프로그램 구현(70)*</li> <li>애플리케이션 설계(50)</li> </ul>	<ul style="list-style-type: none"> <li>애플리케이션 설계(10)*</li> <li>인터페이스 구현(60)</li> <li>통합 구현(50)</li> </ul>	<ul style="list-style-type: none"> <li>통합 구현(10)*</li> <li>공공데이터 활용하기(48)</li> <li>애플리케이션 배포(45)</li> <li>애플리케이션 테스트 관리(13)</li> </ul>	<ul style="list-style-type: none"> <li>애플리케이션 테스트 관리(47)*</li> </ul>
비 NCS (이 론)	<ul style="list-style-type: none"> <li>재량교과(1)</li> </ul>						<ul style="list-style-type: none"> <li>재량교과(4)</li> </ul>	<ul style="list-style-type: none"> <li>재량교과(1)</li> </ul>

## 9. 훈련교재

- SQL 응용(공공데이터 21년상반기\_ver.2)
- SQL 활용(공공데이터 21년상반기\_ver.2)
- UI 구현(공공데이터 21년상반기\_ver.2)
- 공공데이터 활용하기(공공데이터 21년상반기\_ver.2)
- 네트워크 프로그래밍 구현(21년상반기)
- 데이터베이스 구현(공공데이터 21년상반기\_ver.2)
- 서버프로그램 구현(공공데이터 21년상반기\_ver.2)
- 애플리케이션 배포(21년상반기)
- 애플리케이션 설계(21년상반기)
- 애플리케이션 테스트 관리(21년상반기)
- 요구사항 확인(21년상반기)
- 인터페이스 구현(21년상반기)
- 통합 구현(21년상반기)
- 프로그래밍 언어 응용A(공공데이터 21년상반기\_ver.2)
- 프로그래밍 언어 활용(공공데이터 21년상반기\_ver.2)
- SQL응용 학습모듈(공공데이터 21년상반기)
- SQL활용 학습모듈(공공데이터 21년상반기)
- UI 구현 학습모듈(공공데이터 21년상반기)
- UI 디자인 학습모듈(공공데이터 21년상반기)



- 네트워크 프로그래밍 구현 학습모듈(21년상반기)
- 데이터베이스 구현 학습모듈(공공데이터 21년상반기)
- 서버프로그램 구현 학습모듈(21년상반기)
- 애플리케이션 배포 학습모듈(21년상반기)
- 애플리케이션 설계 학습모듈(21년상반기)
- 애플리케이션 테스트관리 학습모듈(21년상반기)
- 요구사항 확인 학습모듈(21년상반기)
- 인터페이스 구현 학습모듈(21년상반기)
- 통합 구현 학습모듈(21년상반기)

## 10. 사용장비

- LED모니터, 모니터, 조립컴퓨터(본체), windows10, windows7, 빔프로젝터, Open JDK Java SE 11, eclipse 2020-09(4.15), draw.io Diagrams, Oracle 11g XE, Sql Developer 19.4, ERD cloud, GITHUB, Visual Studio Code 1.51, Chrome Developer Tool, Oven(카카오 Prototyping tool), Tomcat 9, Spring Framework 5, Spring Tool Suite 3, MyBatis 3.5.5, Junit Test 5, MAVEN 3.6.3, pixlr(픽슬러), Apache OpenOffice 4.1.7, 공공데이터, 카카오맵API

## 11. 특이사항

- 교육원 이용사항 : 교육원 내 음식물 반입 및 취식 금지.
- 안전준수사항 : 수업 종료 후 PC 및 모니터 전원 OFF, 케이블선 원위치, 강의실 이동 및 계단에서 뛰지 않기, 엘리베이터 이용 시 정원준수.
- 평가일자 : 능력단위 또는 교과목이 종료되는 시점에 평가를 실시하는 것을 원칙으로 하며, 재평가 및 결석자는 익일에 실시(부득이한 경우 훈련생들의 사전 동의를 통해 변경 가능)
- 평가방법 : NCS능력단위별 정해진 평가법으로 평가(2가지 이상)
- 환류체계 : 능력단위별 평가결과에 근거하여 과락자 대상 보충학습/개별첨삭지도 진행 후 재평가 실시
- 프로젝트실(자습실) 개방시간 : 평일 9시 ~ 21시 20분까지 선착순으로 이용가능 (공휴일 및 교육원 행사시 제외)

## 프로그래밍 언어 활용

### ● 훈련목표

- 프로그래밍 언어 활용이랑 응용소프트웨어 개발에 사용되는 프로그래밍 언어의 기본문법을 활용하여 기본 응용소프트웨어를 구현하는 능력을 함양.

### - 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
구조적 프로그래밍 언어 활용하기	1.1 응용소프트웨어 개발을 위하여 프로그램 설계서를 확인할 수 있다. 1.2 구조적 프로그래밍 언어를 활용하여 애플리케이션을 작성할 수 있다. 1.3 작성된 애플리케이션의 오류를 식별하고 수정할 수 있다.	<ul style="list-style-type: none"> <li>프로그래밍 기본 문법 활용에 대한 기본 소양 점검</li> <li>언어 특성 활용 능력 점검</li> <li>언어 사용을 위한 기본 능력 및 사고 점검</li> <li>능력단위요소에 적절한 도구를 활용할 수 있는 지를 점검</li> </ul>
객체지향 프로그래밍 언어 활용하기	2.1 응용소프트웨어 개발을 위하여 프로그램 설계서를 확인할 수 있다. 2.2 객체지향 언어를 활용하여 애플리케이션을 작성할 수 있다. 2.3 작성된 애플리케이션의 오류를 식별하고 수정할 수 있다.	
스크립트 활용하기	3.1 응용소프트웨어 개발을 위하여 프로그램 설계서를 확인할 수 있다. 3.2 스크립트 언어를 활용하여 애플리케이션을 작성할 수 있다. 3.3 작성된 애플리케이션의 오류를 식별하고 수정할 수 있다.	

### - 지식 . 기술 . 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> <li>객체지향 프로그래밍 언어 기본 문법</li> <li>객체지향 프로그래밍 언어 디버깅 기법</li> <li>객체지향 프로그래밍 언어 문법 오류</li> <li>객체지향의 개념</li> <li>구조적 프로그래밍 언어 기본 문법</li> <li>구조적 프로그래밍 언어 디버깅 기법</li> <li>구조적 프로그래밍 언어 문법 오류</li> <li>순차, 선택, 반복 제어구조</li> <li>스크립트 언어 디버깅 기법</li> <li>스크립트 언어 선택 능력</li> <li>스크립트 언어 테스트 기법</li> <li>자료구조</li> <li>클래스(Class), 객체(Object), 메서드(Method), aptlw(sage) 구성</li> <li>하향식 설계 개념</li> </ul>
기 술	<ul style="list-style-type: none"> <li>IDE(Integrated Development Environment) 도구 활용 능력</li> <li>객체지향 프로그래밍 언어 디버깅 능력</li> <li>구조적 프로그래밍언어 디버깅 기법</li> <li>동적 바인딩 기술</li> <li>스크립트 언어 디버깅 능력</li> <li>스크립트 언어 선택 능력</li> </ul>

구 분	주 요 내 용
	<ul style="list-style-type: none"> <li>• 스크립트 언어 코드 검토 기술</li> <li>• 컴파일러 사용 능력</li> <li>• 프로그래밍언어 코드 검토 기술</li> </ul>
태 도	<ul style="list-style-type: none"> <li>• 기 작성된 코드를 분석하기 위한 치밀한 태도</li> <li>• 기본 문법을 이해하고 활용하려는 적극적인 자세</li> <li>• 문법 오류를 이해하고 수정• 테스트케이스 작성 방법</li> <li>• 테스트 수행에 필요한 준비사항</li> <li>• 애플리케이션 통합 테스트 수행</li> <li>• 결함에 대한 개선 조치사항</li> <li>• 성능 저하 요인</li> <li>• 애플리케이션 성능 개선</li> <li>• 소스코드 품질 분석 도구 활용하려는 적극적인 태도</li> <li>• 스크립트 언어 오류를 이해하고 수정하려는 적극적인 태도</li> <li>• 스크립트 언어를 이해하고 활용하려는 적극적인 태도</li> <li>• 스크립트 언어의 특징을 분석하기 위한 치밀한 태도</li> <li>• 자신이 수행한 작업에 대한 평가의 객관성</li> <li>• 코드 검토 결과를 수용하고 반영하려는 태도</li> <li>• 프로그래밍 언어의 규칙과 관습을 준수하려는 태도</li> </ul>

## 프로그래밍 언어 응용

### ● 훈련목표

- 프로그래밍 언어 응용이란 응용소프트웨어 개발에 사용되는 프로그래밍 언어의 특징과 라이브러리를 활용하여 기본 응용소프트웨어를 구현하는 능력을 함양.

### - 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
언어특성 활용하기	1.1 프로그래밍 언어별 특성을 파악하고 설명할 수 있다. 1.2 파악된 프로그래밍 언어의 특성을 적용하여 애플리케이션을 구현할 수 있다. 1.3 애플리케이션을 최적화하기 위해 프로그래밍 언어의 특성을 활용할 수 있다.	<ul style="list-style-type: none"> <li>프로그래밍 언어 활용에 대한 기본 소양 점검</li> <li>프로그래밍 언어 특성 활용 능력 점검</li> <li>프로그래밍 언어 라이브러리 적용을 위한 기본 능력 및 사고 점검</li> <li>능력단위요소에 적절한 도구를 활용할 수 있는지를 점검</li> </ul>
라이브러리 활용하기	2.1 애플리케이션에 필요한 라이브러리를 선정할 수 있다. 2.2 애플리케이션 구현을 위해 선택한 라이브러리를 프로그래밍 언어 특성에 맞게 구성할 수 있다. 2.3 선택한 라이브러리를 사용하여 애플리케이션 구현에 적용할 수 있다.	

### - 지식 . 기술 . 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> <li>객체지향 프로그래밍 언어</li> <li>디버깅 기법</li> <li>라이브러리 적용 방법</li> <li>라이브러리 기능 이해</li> <li>선언형 언어</li> <li>스크립트 언어</li> <li>애플리케이션 요구사항 이해</li> <li>절차적 프로그래밍 언어</li> <li>코드 검토 기법</li> </ul>
기 술	<ul style="list-style-type: none"> <li>라이브러리 선정 능력</li> <li>라이브러리 설치 능력</li> <li>라이브러리 적용 능력</li> <li>컴파일러 사용 능력</li> <li>코드 검토 능력</li> <li>프로그래밍 언어 및 도구 활용 능력</li> <li>프로그램 디버깅 능력</li> <li>프로그램 코드 검토 기술</li> <li>프로토타입 작성 능력</li> </ul>
태 도	<ul style="list-style-type: none"> <li>라이브러리를 모듈화하려는 의지</li> <li>라이브러리를 활용하는 적극적인 태도</li> </ul>

구 분	주 요 내 용
	<ul style="list-style-type: none"> <li>• 라이브러리의 영향범위를 최소화하려는 태도</li> <li>• 라이브러리의 특성을 이해하려는 의지</li> <li>• 문법 오류를 이해하고 수정하려는 적극적인 태도</li> <li>• 복잡한 구조 및 절차를 단순화하려는 의지</li> <li>• 애플리케이션 구현의 방법과 절차를 문서화 하려는 의지</li> <li>• 요구사항을 준수하려는 태도</li> <li>• 프로그래밍 언어의 규칙과 관습을 준수하려는 태도</li> <li>• 프로그래밍 언어의 특성을 적용하려는 태도</li> </ul>

# 네트워크 프로그래밍 구현

## ● 훈련목표

- 네트워크 프로그래밍 구현이란 네트워크 프로그램을 구현하기 위한 네트워크 개발환경을 분석하고, 각 단계별로 요구되는 기능을 구현(코딩)하고, 테스트를 하는 능력과 프로그래밍 개발과정에서 발생할 수 있는 오류를 디버깅하여 프로그램을 최적화하는 능력을 함양.

## - 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
개발환경 분석하기	1.1 개발방법 기준에 따라서 네트워크 프로그래밍 구현을 위한 H/W(PC, workstation, server 등) 및 S/W(Unix, windows, IOS 등) 개발환경을 구축할 수 있다. 1.2 네트워크 개발환경의 프로그래밍 구현을 위하여 표준 개발 도구(commands, .net framework control, eclipse 등)를 사용할 수 있다. 1.3 개발 방법론에 따라서 네트워크 프로그래밍 구현을 위한 타겟시스템 형상(Configuration)을 수정하여 보완할 수 있다.	<ul style="list-style-type: none"> <li>요구사항 분석 능력</li> <li>응용 프로그램 개발 종류 및 개발 방법에 대한 이해 능력</li> <li>운영체제에 대한 구조 이해 능력</li> <li>오픈소스에 대한 이해 능력</li> <li>개발 가능한 프로그램 언어에 대한 이해 능력</li> <li>객관적인 시각에서 개발기준을 설정하였는지 여부</li> </ul>
기능 구현하기	2.1 개발방법 기준에 따라서 네트워크 프로그래밍 응용프로그램을 구현할 수 있다. 2.2 프로그래밍 방법론에 따라서 설계내용을 바탕으로 네트워크 프로토콜을 구현할 수 있다. 2.3 구축 계획에 따라 자원관리를 위하여 데이터베이스를 구현할 수 있다. 2.4 효율적인 자원관리를 위하여 에이전트(Agent)를 구현할 수 있다. 2.5 효과적인 트래픽 분석을 위하여 네트워크 QoS 제공방안을 구현할 수 있다.	<ul style="list-style-type: none"> <li>개발기준은 일관성을 유지하고 있는지 여부</li> <li>개발 공정의 흐름에 맞도록 개발기준이 수립되었는지 여부</li> <li>다양한 조건을 기반으로 최적의 개발방법을 선정하였는지 여부</li> <li>개발범위 및 기능의 명확화</li> <li>다양한 방법으로 요구사항 수집과 품질 요구수준을 조사하고 분석하였는지 여부</li> </ul>

- 지식, 기술, 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> <li>• OSI 7 Layer, TCP/IP</li> <li>• Web Protocol, Mark-up Language</li> <li>• Web Service 구조</li> <li>• 네트워크 운영 환경</li> <li>• 네트워크 토폴로지 개념</li> <li>• 네트워크 프로토콜 구성</li> <li>• 소프트웨어 개발 플랫폼</li> <li>• 운영체제 구조</li> <li>• 인터넷 서비스 구조 이해</li> <li>• 프로그래밍 언어 이해</li> </ul>
기 술	<ul style="list-style-type: none"> <li>• 개발 도구 사용 기술</li> <li>• 개발 플랫폼 선정 능력</li> <li>• 개발 환경 분석 기술</li> <li>• 네트워크 설계 능력</li> <li>• 네트워크 아키텍처 수립 능력</li> <li>• 네트워크 품질 평가 능력</li> <li>• 동적 테스트 수행 능력</li> <li>• 이슈 관리 능력</li> <li>• 프로젝트 자원 관리 능력</li> </ul>
태 도	<ul style="list-style-type: none"> <li>• 관련부서 및 조직원 간 원활하게 의사소통을 하려는 태도</li> <li>• 기능설계에 필요한 원칙적이고 논리적인 태도</li> <li>• 기법 및 도구에 대해 적극적으로 수용하려는 태도</li> <li>• 높은 문제해결 의지를 가지는 자세</li> <li>• 다양하고 복잡한 정보에 대한 분석적인 태도</li> <li>• 다양한 검증을 진행하려는 적극적인 자세</li> <li>• 다양한 기법의 장점 수용</li> <li>• 문제해결을 위한 긍정적 자세</li> <li>• 새로운 영역에 대한 적극적인 학습 태도</li> <li>• 생각을 논리적으로 표현할 수 있는 능력</li> <li>• 작업에 필요한 사전 정보나 기술을 준비하는 자세</li> <li>• 적극적, 창의적인 전략 개발 의지</li> <li>• 절차 중심으로 정확한 작업을 추진하는 태도</li> <li>• 체계적이고 분석적으로 문제를 접근하는 태도</li> </ul>

# 데이터베이스 구현

## ● 훈련목표

- 관계형 데이터베이스에서 SQL을 사용하여 목적에 적합한 데이터를 정의하고, 조작하며, 제어하는 능력을 함양

## - 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
DBMS 설치하기	1.1 데이터베이스 요구사항에 따라 하드웨어와 운영체제에 적합한 DBMS를 선정하고 설치계획을 수립할 수 있다. 1.2 DBMS 설치계획에 따라 DBMS를 설치하고 저장소 연결, 환경설정, 클라이언트접속 작업을 수행 할 수 있다. 1.3 설치된 DBMS를 검증 절차에 따라 데이터베이스 요구사항을 확인 하고 DBMS 설치 완료보고서를 작성할 수 있다.	<ul style="list-style-type: none"> <li>• 하드웨어 자원 분석능력</li> <li>• 데이터베이스 설계문서의 분석능력</li> <li>• 데이터베이스 생성 계획 문서 작성 능력</li> <li>• 데이터베이스 SQL 작성 능력</li> </ul>
데이터베이스 생성하기	2.1 설계에 따른 테이블과 인덱스 등의 오브젝트에 기반 하여 데이터 파일, 컨트롤파일, 로그파일에 필요한 용량을 산정하고 형식을 정의할 수 있다. 2.2 관리자 도구 혹은 명령어를 이용하여 데이터베이스 블록사이즈, 로그 관리, 환경설정을 적용하고 확인 할 수 있다. 2.3 데이터베이스에 대한 관리자, 사용자 계정을 이해하고 데이터베이스 사용규칙에 따라 계정별 권한 설정을 적용할 수 있다. 2.4 생성된 데이터베이스의 용량, 권한, 환경설정을 점검하고 데이터베이스 명세적 검증을 통하여 데이터베이스 완료보고서를 작성할 수 있다.	<ul style="list-style-type: none"> <li>• DBMS 설치 계획 문서 작성 능력</li> <li>• DBMS의 종류</li> <li>• DBMS 운용 방법</li> <li>• DBMS 모니터링 프로그램 활용 능력</li> <li>• 설치 프로그램과 DBMS 클라이언트 툴 활용능력</li> <li>• 테이블의 생성 및 관리</li> <li>• 제약조건, 데이터베이스 트리거와 같은 무결성 보장을 위한 기술 적용 능력</li> </ul>
데이터베이스 오브젝트 생성하기	3.1 테이블, 인덱스, 제약조건을 이해하고 데이터베이스 설계에 따라 설치된 DBMS에 적합한 오브젝트 생성 계획을 수립할 수 있다. 3.2 물리 데이터베이스 설계에 따라 데이터베이스 오브젝트를 생성하기 위한 DDL(Data Definition Language)을 작성할 수 있고 생성된 오브젝트에 대한 유효성 여부를 검사할 수 있다. 3.3 생성된 데이터베이스 오브젝트의 적정성과 무결성을 판단하고 데이터베이스 오브젝트 명세를 포함한 완료보고서를 작성할 수 있다.	<ul style="list-style-type: none"> <li>• 각종 구성, 환경 및 절차에 관한 지식</li> <li>• 관련부서와 계획 협의를 위한 의사소통 능력</li> <li>• 작업 시 사고 사례 및 대처방안에 대한 내용</li> <li>• 항목별 체크리스트 작성</li> </ul>

## - 지식, 기술, 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> <li>• DBMS 구성방법</li> <li>• DBMS 환경설정 방법</li> <li>• DBMS 환경설정 요소</li> </ul>



구 분	주 요 내 용
	<ul style="list-style-type: none"> <li>• SQL 문법</li> <li>• SQL 사용 방법</li> <li>• 관계형 데이터베이스 개념</li> <li>• 데이터 제약조건 규칙</li> <li>• 데이터베이스 구조</li> <li>• 데이터베이스 보안 적용 방법</li> <li>• 데이터베이스 오브젝트 특성</li> <li>• 데이터베이스 용량 설정 및 확장 방법</li> <li>• 데이터베이스 테이블 규칙</li> <li>• 데이터베이스 환경설정 방법</li> <li>• 뷰 개념 및 활용방법</li> <li>• 운영체제 특성</li> <li>• 인덱스 개념</li> <li>• 하드웨어 특성</li> </ul>
기 술	<ul style="list-style-type: none"> <li>• DBMS 설치프로그램 사용 기술</li> <li>• DBMS 운용 검증 능력</li> <li>• DBMS 환경 변수 설정 기술</li> <li>• 데이터베이스 계정관리 기술</li> <li>• 데이터베이스 물리 구조 구현 기술</li> <li>• 데이터베이스 복구 기술 적용 능력</li> <li>• 데이터베이스 생성 프로그램 작성 기술</li> <li>• 데이터베이스 설계서 분석능력</li> <li>• 뷰 생성 및 삭제 SQL 작성 능력</li> <li>• 시스템 사양 검토 능력</li> <li>• 오브젝트 상태 확인 기술</li> <li>• 운영체제 설정 변경 능력</li> <li>• 인덱스 생성 및 관리 능력</li> <li>• 제약조건 생성 및 관리 능력</li> <li>• 테이블 생성 및 관리 능력</li> <li>• 하드웨어 스펙 분석 기술</li> </ul>
태 도	<ul style="list-style-type: none"> <li>• DBMS, 하드웨어, 운영체제와의 호환성을 분석하고 적용하려는 의지</li> <li>• SQL(ANSI-SQL) 언어에 대한 이해와 적용 노력</li> <li>• 기술적 위험에 적극적으로 대비하려는 노력</li> <li>• 기술적 위험에 적극적으로 대비하려는 태도</li> <li>• 데이터 조작 시 제약조건을 준수하려는 태도</li> <li>• 데이터베이스 기술 동향에 대한 이해와 관련 제품들에 대해 파악하기 위한 노력</li> <li>• 데이터의 무결성 준수하려는 노력</li> <li>• 물리적 데이터베이스 설계 내용을 준수하려는 태도</li> <li>• 사용자의 요구 사항을 적극적으로 청취하고 이해하려는 노력</li> <li>• 세밀한 작업지시서 및 규격서 검토 노력</li> <li>• 세밀한 작업지시서 작성 및 규격서 검토</li> <li>• 운영, 관리 단계에서 요구되는 데이터베이스 기술에 전반적인 이해를 위한 노력</li> <li>• 적합한 계획 수립을 위한 책임감</li> </ul>

## SQL활용

### ● 훈련목표

- SQL활용이란 관계형 데이터베이스에서 SQL을 사용하여 목적에 적합한 데이터를 정의하고, 조작하며, 제어하는 능력을 함양.

### - 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
기본 SQL 작성하기	1.1 생성된 테이블의 목록, 테이블의 구조와 제약조건을 파악하기 위해 데이터 사전을 조회하는 명령문을 작성할 수 있다. 1.2 조인, 서브쿼리, 집합연산자를 사용하여 두 개 이상의 테이블로부터 데이터를 조회하는 DML(Data Manipulation Language)명령문을 작성할 수 있다. 1.3 테이블의 구조와 제약조건을 생성, 삭제하고 수정하는 DDL(Data Definition Language) 명령문을 작성할 수 있다. 1.4 업무단위인 트랜잭션의 완료와 취소를 위한 DCL(Data Control Language) 명령문을 작성할 수 있다.	<ul style="list-style-type: none"> <li>• 단일 테이블을 생성, 삭제하고 수정하는 명령문 작성기술</li> <li>• 제약조건, 뷰와 인덱스를 정의하는 명령문 작성기술</li> <li>• 데이터를 삽입, 수정, 삭제, 조회하는 명령문 작성기술</li> <li>• 단일행 함수와 그룹별 집계 함수의 사용기술</li> </ul>
고급 SQL 작성하기	2.1 생성된 테이블의 목록, 테이블의 인덱스와 뷰를 파악하기 위해 데이터 사전을 조회하는 명령문을 작성할 수 있다. 2.2 테이블 조회 시간을 단축하기 사용하는 인덱스의 개념을 이해하고 인덱스를 생성하는 DDL(Data Definition Language)명령문을 작성할 수 있다. 2.3 먼저 생성된 테이블들을 이용하여 새로운 테이블과 뷰를 생성하는 DDL(Data Definition Language)명령문을 작성할 수 있다.	<ul style="list-style-type: none"> <li>• 조인을 사용하는 명령문 작성기술</li> <li>• 트랜잭션의 제어, 사용자 권한관리 명령문 사용기술</li> <li>• 매개변수를 사용하는 프로시저, 사용자 정의 함수, 트리거 작성과 사용기술</li> </ul>

- 지식, 기술, 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> <li>• DBMS 개념</li> <li>• SQL 함수 사용 방법</li> <li>• SQL 문법</li> <li>• SQL 실행 원리</li> <li>• 관계형 데이터 모델 특성</li> <li>• 데이터사전 개념</li> <li>• 메타데이터 개념</li> <li>• 뷰 개념 및 활용방법</li> <li>• 서브쿼리 사용 방법</li> <li>• 인덱스 개념</li> <li>• 조인 종류와 특성</li> <li>• 집합연산자 종류 및 특성</li> <li>• 트랜잭션의 특성</li> </ul>
기 술	<ul style="list-style-type: none"> <li>• DBMS 내장 함수 사용 능력</li> <li>• DDL(Data Definition Language) 작성 기술</li> <li>• DML(Data Manipulation Language) 작성 기술</li> <li>• Select 명령문 작성 기술</li> <li>• 데이터 사전 조회 SQL 사용 능력</li> <li>• 뷰 생성 및 삭제 SQL 작성 능력</li> <li>• 서브쿼리 작성 능력</li> <li>• 인덱스 생성 및 삭제 SQL 작성 능력</li> <li>• 조인 구문 사용 능력</li> <li>• 집합연산자 사용 능력</li> <li>• 트랜잭션 명령문 작성 기술</li> </ul>
태 도	<ul style="list-style-type: none"> <li>• 데이터 보안에 대한 책임감</li> <li>• 데이터 요구사항에 긍정적인 태도</li> <li>• 데이터 조작 시 제약조건에 대한 인식</li> <li>• 데이터 처리 결과의 일관성을 유지하려는 노력</li> <li>• 데이터 표준을 준수하려는 노력</li> <li>• 데이터의 무결성을 유지하려는 태도</li> </ul>

# SQL응용

## ● 훈련목표

- SQL응용이란 관계형 데이터베이스에서 SQL을 사용하여 응용시스템의 요구기능에 적합한 데이터를 정의하고, 조작하며, 제어하는 능력을 함양.

## - 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
절차형 SQL 작성하기	1.1 반복적으로 사용하는 특정 기능을 수행하기 위해 여러 개의 SQL명령문을 포함하는 프로시저를 작성하고 프로시저 호출문을 작성할 수 있다. 1.2 일련의 연산처리 결과가 단일 값으로 반환되는 사용자 정의함수를 작성하고 사용자 정의함수를 호출하는 쿼리를 작성할 수 있다. 1.3 하나의 이벤트가 발생하면 관련성이 있는 몇 개의 테이블 간에 연속적으로 데이터 삽입, 삭제, 수정을 할 수 있는 트리거를 작성할 수 있다.	<ul style="list-style-type: none"> <li>• 단일 테이블을 생성, 삭제하고 수정하는 명령문 작성기술</li> <li>• 제약조건, 뷰와 인덱스를 정의하는 명령문 작성기술</li> <li>• 데이터를 삽입, 수정, 삭제, 조회하는 명령문 작성기술</li> <li>• 단일행 함수와 그룹별 집계 함수의 사용기술</li> <li>• 조인을 사용하는 명령문 작성기술</li> <li>• 트랜잭션의 제어, 사용자 권한관리 명령문 사용기술</li> <li>• 매개변수를 사용하는 프로시저, 사용자 정의 함수, 트리거 작성과 사용기술</li> </ul>
응용 SQL 작성하기	2.1 윈도우함수와 그룹함수를 사용하여 순위와 소계, 중계, 총합계를 산출하는 DML(Data Manipulation Language)명령문을 작성할 수 있다. 2.2 응용시스템에서 사용하는 특정 기능을 수행하기 위한 SQL문을 작성할 수 있다. 2.3 사용자의 그룹을 정의하고 사용자를 생성 또는 변경할 수 있고 사용자의 권한 부여와 회수를 위한 DCL(Data Control Language)명령문을 작성할 수 있다.	

- 지식, 기술, 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> <li>• SQL 문법</li> <li>• SQL 실행 원리</li> <li>• 매개변수 및 제어문 사용 방법</li> <li>• 사용자 생성, 권한부여와 회수방법</li> <li>• 사용자 정의함수 사용 방법</li> <li>• 오류 발생과 처리규칙</li> <li>• 절차형 SQL 문법</li> <li>• 테이블 간 업무처리 규칙</li> <li>• 트리거 및 이벤트 개념</li> <li>• 트리거 사용 방법</li> <li>• 프로시저 사용 방법</li> <li>• 함수 종류 및 특성</li> </ul>
기 술	<ul style="list-style-type: none"> <li>• DBMS 내장 함수 사용 능력</li> <li>• DCL(Data Control Language) 작성 기술</li> <li>• 뷰 생성 능력</li> <li>• 사용자 정의함수 사용 능력</li> <li>• 사용자 정의함수 작성 기술</li> <li>• 삭제 SQL 작성 능력</li> <li>• 서브쿼리 작성 능력</li> <li>• 인덱스 생성 및 삭제 SQL 작성 능력</li> <li>• 조인 구문 사용 능력</li> <li>• 집합연산자 사용 능력</li> <li>• 트리거 사용 능력</li> <li>• 트리거 작성 기술</li> <li>• 프로시저 사용 능력</li> <li>• 프로시저 작성 기술</li> </ul>
태 도	<ul style="list-style-type: none"> <li>• 데이터 처리 결과의 일관성을 유지하려는 노력</li> <li>• 데이터 처리의 효율성 지향</li> <li>• 데이터의 무결성을 유지하려는 태도</li> <li>• 발생 가능한 오류를 모두 예측하여 처리하려는 노력</li> <li>• 절차형 SQL의 데이터베이스 제품별 차이에 대한 인식</li> <li>• 테이블간의 연속적인 데이터 조작을 자동화하려는 노력</li> </ul>

# UI 디자인

## ● 훈련목표

- UI 디자인이란 UI 아키텍처 설계 후 시각적인 요소를 디자인하기 위해 비주얼 디자인 콘셉트를 도출하며 이 콘셉트에 근거하여 GUI 디자인 요소를 선정하는 능력을 함양.

## - 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
GUI 디자인 콘셉트 도출하기	1.1 UI/UX 콘셉트 기획안에 따라서 GUI 디자인 요소와 시각적인 콘셉트를 도출 할 수 있다. 1.2 사용자 리서치에서 도출된 페르소나 정의에 따라서 사용자의 라이프 스타일이 반영된 비주얼 무드보드를 작업할 수 있다. 1.3 도출된 비주얼 무드보드에 의하여 GUI 상세 디자인을 하기 위한 콘셉트 기획서를 작성할 수 있다.	<ul style="list-style-type: none"> <li>• 디자인 트렌드 분석 능력</li> <li>• 사용자 요구사항의 이해</li> <li>• 아이디어를 구체화시키는 능력</li> <li>• 아이디어 자료 분석과 기획력</li> <li>• 디자인 콘셉트를 도출할 수 있는 능력</li> <li>• 디자인 기획서를 작성하는 능력</li> <li>• 자료를 수집하는 능력</li> </ul>
GUI 디자인 요소 선정하기	2.1 UI 설계에 따라서 기능별 대표 화면을 선정할 수 있다. 2.2 선정된 대표 화면에 대하여 화면 레이아웃을 선정할 수 있다. 2.3 GUI 콘셉트 기획서에 따라서 GUI 디자인 요소별 컬러를 선정할 수 있다. 2.4 GUI 콘셉트 기획서에 따라서 레이블과 콘텐츠에 대한 폰트를 선정할 수 있다.	<ul style="list-style-type: none"> <li>• 아이디어를 시각화시키는 능력</li> <li>• 디자인 도구의 활용 능력</li> <li>• 흐름도 작성도구 활용 능력</li> <li>• 화면을 구성하는 능력</li> <li>• 사용자의 성향을 파악할 수 있는 능력</li> <li>• 디자인 시안을 만드는 능력 (러프 스케치, 러프 모형 제작을 위해)</li> <li>• 프레젠테이션 능력</li> </ul>

- 지식, 기술, 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> <li>• GUI 디자인 요소 이해</li> <li>• HCI(Human Computer Interaction)</li> <li>• UI/UX 디자인 동향 분석 방법</li> <li>• 그래픽 메타포(Graphic Metaphor)</li> <li>• 그리드 시스템(Grid System)</li> <li>• 디자인 구성 방법</li> <li>• 반응형 웹</li> <li>• 사용자 요구사항 추출기법</li> <li>• 시각적인 그래픽 요소에 대한 사용성</li> <li>• 페르소나(Persona) 이해</li> <li>• 플랫폼별 특성</li> </ul>
기 술	<ul style="list-style-type: none"> <li>• GUI 디자인 능력</li> <li>• GUI 디자인 콘셉트 도출 능력</li> <li>• GUI 템플릿 작성 능력</li> <li>• 그리드 시스템 활용 기술</li> <li>• 디자인 동향 이해</li> <li>• 디자인 전략 수립 능력</li> <li>• 레이블 및 콘텐츠의 폰트 활용 능력</li> <li>• 메뉴구조 템플릿 구성 기술</li> <li>• 브레인스토밍(Brainstorming) 능력</li> <li>• 비주얼 디자인 콘셉트 기획서 작성 기술</li> <li>• 비주얼 전략 무드보드(Moodboard) 작성 기술</li> <li>• 페르소나 기법 활용 능력</li> </ul>
태 도	<ul style="list-style-type: none"> <li>• GUI 콘셉트 기획서를 논리적으로 작성 하려는 의지</li> <li>• 다양성과 개방성에 대한 열린 태도</li> <li>• 도출한 아이디어를 구체화시키고 시각화 하는 태도</li> <li>• 문제 해결을 위한 전략적 사고방식 및 태도</li> <li>• 비주얼 디자인 콘셉트 도출 의지</li> <li>• 사용자 중심의 GUI 디자인 하려는 태도</li> <li>• 상대방의 의견을 수렴할 수 있는 개방적 태도</li> <li>• 새롭고 시장성 있는 GUI 디자인 창출 의지</li> <li>• 수집된 요구사항에 대한 분석적 태도</li> <li>• 일관성 있고 통일성 있는 GUI 디자인을 하려는 태도</li> <li>• 차별화된 GUI 디자인 접근방법으로 추진하려는 태도</li> <li>• 최신 UI/UX와 GUI트렌드를 조사하고 적극 활용하는 태도</li> </ul>

# UI 구현

## ● 훈련목표

- UI 구현이란 UI 설계 산출물과 GUI 디자인 가이드를 바탕으로 UI 설계와 UI 구현 표준을 검토하는 능력을 함양.

## - 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
UI 설계 검토하기	1.1 UI 제작을 위하여 GUI 디자인 가이드를 이해하고 이를 기반으로 구현 가능성 여부를 검토할 수 있다. 1.2 구현 환경에 따라서 구체적인 GUI 프로세스의 이해와 설계 변경 여부를 파악할 수 있다. 1.3 UI 구현 표준 수립을 위하여 UI 검토 의견서를 작성할 수 있다.	<ul style="list-style-type: none"> <li>멀티미디어에 대한 이해 및 활용 능력</li> <li>웹 표준에 대한 이해 및 표현하는 능력</li> <li>웹퍼블리싱하여 웹 상에 표현하는 능력</li> <li>디자인 도구 활용하여 시각화 하는 능력</li> <li>아날로그 프로토타입 제작 능력</li> <li>디지털 프로토타입 제작 능력</li> <li>UI 제작물을 포트폴리오화 하는 능력</li> </ul>
UI 구현 표준 검토하기	2.1 고객 요구사항, 접근성 기준, 플랫폼에 대한 UI 표준 관련 기준을 파악할 수 있다. 2.2 실제 페이지 제작을 위하여 아이콘, 레이아웃, 화면 개발 환경에 적합한 표준을 검토할 수 있다. 2.3 UI 구현 표준을 이해관계자와 검토하여 최종 표준안에 반영할 수 있다.	

## - 지식, 기술, 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> <li>UI 개발 프로세스</li> <li>UI/UX 개발 프로세스</li> <li>디지털 컨트롤(Digital Control)</li> <li>사용자 중심 인터랙션 디자인(Interaction Design)</li> <li>사용자 중심 인터랙션 디자인(Interaction Design) 설계</li> <li>성공한 모델의 사용자 경험</li> <li>웹 접근성 이해</li> <li>웹 표준 이해</li> <li>인터페이스 디자인</li> <li>인포메이션 아키텍처(Information Architecture)</li> <li>컨테이너(Container) 및 코덱(Codec)의 이해</li> <li>퍼블리싱 기법</li> <li>프로토타입 이해</li> </ul>
기 술	<ul style="list-style-type: none"> <li>ISO 9421 국제 표준</li> <li>UI 구현 도구 활용 기술</li> <li>UI 구현 환경 분석 기술</li> <li>UI 표준 가이드라인 파악 능력</li> </ul>



구 분	주 요 내 용
	<ul style="list-style-type: none"> <li>• UI/UX 기술 동향 분석 기술</li> <li>• 기능 설계 분석 능력</li> <li>• 아키텍처 뷰 작성 기술</li> <li>• 아키텍처 스타일 분석 능력</li> <li>• 정보 아키텍처 설계 분석 능력</li> <li>• 프로토타입 제작 기술</li> </ul>
태 도	<ul style="list-style-type: none"> <li>• UI 구현 환경에 대해 이해하려는 태도</li> <li>• UI 설계 산출물의 지속적인 관리 태도</li> <li>• 과제 리딩 태도 및 의사결정하려는 태도</li> <li>• 논리적으로 사고하려는 태도</li> <li>• 문제를 제시하고 조율하려는 적극적인 태도</li> <li>• 사용자의 사용편의성(usability)을 고려하여 설계하려는 태도</li> <li>• 설계된 산출물의 재사용성을 고려하는 태도</li> <li>• 적극적으로 사고하는 태도</li> <li>• 지속적으로 사용성을 평가하는 태도</li> <li>• 지속적으로 새로운 기술에 관심을 가지는 태도</li> <li>• 창의적으로 사고하는 태도</li> <li>• 채택된 내용을 전파하고자하는 태도</li> <li>• 타인에 대한 이해와 대화하는 태도</li> <li>• 프로토타입 개발에 할당된 시간과 리소스를 고려하는 태도</li> </ul>

## 요구사항 확인

### ● 훈련목표

- 요구사항 확인이란 업무 분석가가 수집·분석·정의한 요구사항과 이에 따른 분석모델에 대해서 확인과 현행 시스템에 대해 분석하는 능력을 함양.

### - 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
현행 시스템 분석하기	1.1 개발하고자 하는 응용소프트웨어에 대한 이해를 높이기 위해, 현행 시스템의 적용현황을 파악함으로써 개발범위와 향후 개발될 시스템으로의 이행방향성을 분석할 수 있다. 1.2 개발하고자 하는 응용소프트웨어와 관련된 운영체제, 데이터베이스관리시스템, 미들웨어 등의 요구사항을 식별할 수 있다. 1.3 현행 시스템을 분석하여, 개발하고자 하는 응용소프트웨어가 이후 적용될 목표시스템을 명확하고 구체적으로 기술할 수 있다.	<ul style="list-style-type: none"> <li>• 산업 분야에 대한 분석</li> <li>• 요구사항 명세화의 정 확성 여부</li> <li>• 서버 하드웨어 용량 산 정 방법 숙지 및 수행 내역</li> <li>• 운영체제의 주요 기능, 구조, 특성에 대한 분석</li> <li>• 미들웨어(DBMS/WAS)의 주요 기능, 구조, 특성에 대한 분석</li> </ul>
요구사항 확인하기	2.1 소프트웨어 공학기술의 요구사항 분석 기법을 활용하여 업무 분석가가 정의한 응용소프트웨어의 요구사항을 확인할 수 있다. 2.2 업무 분석가가 분석한 요구사항에 대해 정의된 검증 기준과 절차에 따라서 요구사항을 확인할 수 있다. 2.3 업무 분석가가 수집하고 분석한 요구사항이 개발하고자 하는 응용소프트웨어에 미칠 영향에 대해서 검토하고 확인할 수 있다.	
분석모델 확인하기	3.1 소프트웨어 공학기술의 요구사항 도출 기법을 활용하여 업무 분석가가 제시한 분석모델에 대해서 확인할 수 있다. 3.2 업무 분석가가 제시한 분석모델이 개발할 응용소프트웨어에 미칠 영향을 검토하여 기술적인 타당성 조사를 할 수 있다. 3.3 업무 분석가가 제시한 분석모델에 대해서 응용소프트웨어를 개발하기 위해 필요한 추가적인 의견을 제시할 수 있다.	

- 지식, 기술, 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> <li>• 가상화 개념</li> <li>• 데이터베이스 이해</li> <li>• 모델링 기법</li> <li>• 산업 분야별 업무 특성</li> <li>• 소프트웨어 개발 방법론</li> <li>• 애플리케이션 요구사항 이해</li> <li>• 요구공학(Requirement Engineering)</li> <li>• 요구사항 분석 기법</li> <li>• 요구사항 영향도 분석 방법</li> <li>• 용어사전 작성 방법</li> <li>• 운영체제 특징</li> <li>• 유즈케이스 이해</li> <li>• 타당성 검토 기법</li> <li>• 타당성 분석기법</li> <li>• 프로젝트 환경 및 특수성</li> <li>• 플랫폼 기능 및 성능 특성</li> <li>• 해당 플랫폼 지식</li> </ul>
기 술	<ul style="list-style-type: none"> <li>• DBMS(Database Management System) 구성 및 관리 기술</li> <li>• UML(Unified Modeling Language) 모델링 기술</li> <li>• 가상화 관련 기술</li> <li>• 내부 및 외부 환경 분석 기술</li> <li>• 네트워크 구성 및 관리 능력</li> <li>• 리뷰 진행 기술</li> <li>• 분석 자동화 도구 활용 능력</li> <li>• 요구사항 검증 능력</li> <li>• 요구사항 관리 도구 활용 능력</li> <li>• 운영체제 구성 및 관리 능력</li> <li>• 유즈케이스 작성 능력</li> <li>• 저장장치 구성 및 관리 능력</li> </ul>
태 도	<ul style="list-style-type: none"> <li>• 검증할 항목 분석을 위한 치밀한 태도</li> <li>• 고객의 요청을 수용하고자 하는 태도</li> <li>• 기술 관련 각종 정보 수집에 대한 적극성</li> <li>• 분류 및 정리 태도</li> <li>• 비즈니스의 내용을 정확히 반영하는 프로세스 모델을 만들려는 치밀한 자세</li> <li>• 시스템과 개발 소프트웨어와의 관계를 파악하려는 태도</li> <li>• 시스템에 대한 정확성과 이해의 완전성을 갖고자 하는 태도</li> <li>• 요구사항에 대한 일관성을 부여하는 자세</li> <li>• 요구사항을 유기적 관계를 발견하는 자세</li> <li>• 요구사항을 정확하게 기술하려는 자세</li> <li>• 요구사항의 명확성을 찾으려는 자세</li> <li>• 요구사항의 정확성과 완전성을 확보하려는 자세</li> <li>• 용어정의의 정확성과 완전성을 기하고자 하는 의지</li> <li>• 정확한 유스케이스를 이해하고 분석하려는 자세</li> <li>• 책임감 및 검증에 대한 완벽함을 추구하는 태도</li> </ul>

# 서버프로그램 구현

## ● 훈련목표

- 서버프로그램 구현이란 애플리케이션 설계를 기반으로 개발에 필요한 환경을 구성하고, 프로그래밍 언어와 도구를 활용하여 공통모듈, 업무프로그램과 배치 프로그램을 구현하는 능력을 함양.

## - 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
개발환경 구축하기	1.1 응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어의 필요 사항을 검토하고 이에 따라, 개발환경에 필요한 준비를 수행할 수 있다. 1.2 응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어를 설치하고 설정하여 개발환경을 구축할 수 있다. 1.3 사전에 수립된 형상관리 방침에 따라, 운영정책에 부합하는 형상관리 환경을 구축할 수 있다.	<ul style="list-style-type: none"> <li>분석/설계 산출물 및 다이어그램 분석</li> <li>개발 애플리케이션의 주요 메커니즘 및 적용 개발언어 분석 능력</li> <li>코드 이름규칙, 주석처리 방법</li> <li>형상관리 절차 및 기준의 이해</li> </ul>
공통 모듈 구현하기	2.1 공통 모듈의 상세 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 업무 프로세스 및 서비스의 구현에 필요한 공통 모듈을 작성할 수 있다. 2.2 소프트웨어 측정지표 중 모듈 간의 결합도는 줄이고 개별 모듈들의 내부 응집도를 높인 공통모듈을 구현할 수 있다. 2.3 개발된 공통 모듈의 내부 기능과 제공하는 인터페이스에 대해 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.	<ul style="list-style-type: none"> <li>서버프로그램 공통 리소스(예:데이터베이스) 활용의 모듈화 이해</li> <li>프로그래밍 언어별 데이터베이스 연동 방법 이해</li> <li>프로그래밍 언어별 웹 서버프로그램 개발 방법의 이해</li> </ul>
서버 프로그램 구현하기	3.1 업무 프로세스 맵과 세부 업무 프로세스를 확인할 수 있다. 3.2 세부 업무프로세스를 기반으로 프로그래밍 언어와 도구를 활용하여 서비스의 구현에 필요한 업무 프로그램을 구현할 수 있다. 3.3 개발하고자 하는 목표 시스템의 잠재적 보안 취약성이 제거될 수 있도록 서버 프로그램을 구현할 수 있다. 3.4 개발된 업무 프로그램의 내부 기능과 제공하는 인터페이스에 대해 테스트를 수행할 수 있다.	<ul style="list-style-type: none"> <li>서버프로그램에서 적용하는 주요 모듈구성방법(레이어별 패턴)의 이해</li> <li>컴포넌트 코드작성의 이해</li> <li>프로토콜에 따른 클라이언트와 서버 프로그램의 요청 및 응답 메커니즘 이해</li> <li>테스트를 위한 유효값 입력, 비유효값 입력, 예외 조건의 이해</li> </ul>
배치 프로그램 구현하기	4.1 애플리케이션 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 배치 프로그램 구현 기술에 부합하는 배치 프로그램을 구현할 수 있다. 4.2 목표 시스템을 구성하는 하위 시스템 간의 연동 시, 안정적이고 안전하게 동작할 수 있는 배치 프로그램을 구현할 수 있다. 4.3 개발된 배치 프로그램을 테스트를 수행할 수 있다.	<ul style="list-style-type: none"> <li>배치프로그램을 위한 데이터베이스와 연관된 메모리 활용방법의 이해</li> <li>네트워크 및 파일처리 기반의 배치프로그램 작성방법 이해</li> <li>배치프로그램 트랜잭션 처리 방법의 이해</li> </ul>

- 지식, 기술, 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> <li>• 개발 환경 구축 방법</li> <li>• 결함 관리</li> <li>• 네트워크 이해</li> <li>• 데이터베이스 이해</li> <li>• 분산 프로그래밍 기법</li> <li>• 설계 산출물 이해</li> <li>• 설계 산출물 해석 방법</li> <li>• 소프트웨어 개발 프레임워크</li> <li>• 소프트웨어 아키텍처 이해</li> <li>• 소프트웨어 테스트 이해</li> <li>• 테스트 지식 체계(ISTQB Syllabus, CSTE CBOK)</li> <li>• 프로그래밍 언어 활용 방법</li> <li>• 프로그램 언어 활용 방법</li> </ul>
기 술	<ul style="list-style-type: none"> <li>• DE(Integrated Development Environment) 도구 활용 능력</li> <li>• IDE(Integrated Development Environment) 도구 활용 능력</li> <li>• 네트워크 구성 및 관리 능력</li> <li>• 단위테스트 도구 활용 능력</li> <li>• 모듈화 능력</li> <li>• 보안도구 활용 능력</li> <li>• 애플리케이션 프레임워크 활용 기술</li> <li>• 인터페이스 시스템 활용 능력</li> <li>• 재사용 기술</li> <li>• 컴파일러 사용 능력</li> <li>• 프로그래밍 언어 코드 검토 기술</li> <li>• 프로그래밍 언어 활용 능력</li> <li>• 프로그래밍언어 활용 능력</li> <li>• 프로그램 디버깅 능력</li> <li>• 협업도구 활용 능력</li> <li>• 형상관리 도구 활용 능력</li> </ul>
태 도	<ul style="list-style-type: none"> <li>• 개발 수행에 요구되는 표준을 준수하려는 태도</li> <li>• 개발 팀원 간의 원활한 협업을 추구하는 태도</li> <li>• 개발환경 개선 의지</li> <li>• 기술 및 도구에 대해 적극적으로 배우고자 하는 태도</li> <li>• 산출물 완성도를 위한 적극적인 태도</li> <li>• 성공적인 개발을 위한 의지</li> <li>• 적극적이고 세밀히 분석하는 자세</li> <li>• 적극적인 업무 분석 태도</li> <li>• 제품의 품질 및 고객 우선주의 태도</li> <li>• 주어진 과제를 완수하는 책임감</li> <li>• 최적의 개발환경을 구축하고자 하는 의지</li> <li>• 효율적인 모듈 설계를 위한 의지</li> </ul>

# 애플리케이션 설계

## ● 훈련목표

- 애플리케이션 설계란 요구사항 확인을 통한 상세 분석 결과, 소프트웨어 아키텍처 가이드라인 및 소프트웨어 아키텍처 산출물에 의거하여 이에 따른 애플리케이션 구현을 수행하기 위해 공통모듈 설계, 타 시스템 연동에 대하여 상세 설계하는 능력을 함양.

## - 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
공통 모듈 설계하기	1.1 재사용성 확보와 중복 개발을 회피하기 위하여, 전체 시스템 차원과 단위시스템 차원의 공통부분을 식별하여 이에 대한 상세 명세를 작성할 수 있다. 1.2 개발할 응용소프트웨어의 전반적인 기능과 구조를 이해하기 쉬운 크기로 공통 모듈을 설계할 수 있다. 1.3 소프트웨어 측정지표 중 모듈 간의 결합도는 줄이고 개별 모듈들의 내부 응집도는 높이기 위한 공통모듈을 설계할 수 있다. 1.4 전반적인 처리 논리 구조에 예기치 못한 영향을 끼치지 않도록 공통 모듈 인터페이스의 인덱스 번호나 기능 코드를 설계할 수 있다.	<ul style="list-style-type: none"> <li>요구사항 추적성 확보</li> <li>소프트웨어 아키텍처와 일관성 확인</li> <li>프로토타입 작성 및 검증 능력</li> </ul>
타 시스템 연동설계하기	2.1 소프트웨어 아키텍처에서 정의한 타 시스템 연동 리스트 및 연동 방안을 참조하여, 타 시스템 연동 상세 설계의 가이드라인을 작성할 수 있다. 2.2 소프트웨어 아키텍처의 정의를 반영한 연동 상세 설계 가이드라인에 따라, 타 시스템 연동 상세 설계할 수 있다. 2.3 소프트웨어 아키텍처에 따라 선정된 개발 및 운영 환경에 사용될 기술영역별 미들웨어솔루션에 대하여 명세를 작성할 수 있다. 2.4 소프트웨어 아키텍처에 따른 시스템 간의 연동 시, 발생할 수 있는 오류를 예측하고 이의 대응 방안에 대해 제시할 수 있다.	

- 지식, 기술, 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> <li>• 네트워크 이해</li> <li>• 모듈 간 결합도</li> <li>• 모듈 내 응집도</li> <li>• 분산 프로그래밍 기법</li> <li>• 산업 분야별 업무 특성</li> <li>• 설계 모델링 기법</li> <li>• 소프트웨어 개발방법론</li> <li>• 소프트웨어 아키텍처 이해</li> <li>• 채널시스템 이해</li> </ul>
기 술	<ul style="list-style-type: none"> <li>• E-R(Entity-Relationship) 모델링 능력</li> <li>• IIIDE(Integrated Development Environment) 도구 활용 능력</li> <li>• UML(Unified Modeling Language) 모델링 기술</li> <li>• 기술영역별 미들웨어 및 솔루션 활용 능력</li> <li>• 모델링 기술</li> <li>• 연동프로그램 제어 기술</li> <li>• 프레임워크 활용 능력</li> </ul>
태 도	<ul style="list-style-type: none"> <li>• 고객의 요청을 적극적으로 수용하고자 하는 의지</li> <li>• 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도</li> <li>• 산출물 완성도를 위한 적극적인 태도</li> <li>• 아키텍트와 설계자 간의 협업을 추구하는 태도</li> <li>• 주어진 과제를 완수하는 책임감</li> <li>• 책임감 및 설계문서에 완벽함을 추구하는 태도</li> </ul>

# 인터페이스 구현

## ● 훈련목표

- 인터페이스 구현이란 인터페이스 설계서를 확인하고, 인터페이스 설계서에 따라 기능을 구현하고 검증하는 능력을 함양.

## - 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
인터페이스 설계서 확인하기	1.1 인터페이스 설계서를 기반으로 외부 및 내부 모듈 간의 공통적으로 제공되는 기능과 각 데이터의 인터페이스를 확인할 수 있다. 1.2 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 인터페이스의 기능을 식별할 수 있다. 1.3 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 인터페이스를 위한 데이터 표준을 확인할 수 있다.	<ul style="list-style-type: none"> <li>• 연계 메커니즘에 대한 이해도</li> <li>• 연계 표준/프로토콜에 대한 이해도</li> <li>• 미들웨어, 인터페이스 소프트웨어에 대한 이해도</li> </ul>
인터페이스 기능 구현하기	2.1 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 일관되고 정형화된 인터페이스 기능 구현을 정의할 수 있다. 2.2 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 공통적인 인터페이스를 구현할 수 있다. 2.3 개발하고자 하는 응용소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 인터페이스 기능 구현 실패 시 예외처리 방안을 정의할 수 있다. 2.4 응용소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 인터페이스 보안 기능을 적용할 수 있다.	<ul style="list-style-type: none"> <li>• 모니터링 도구 사용법 및 모니터링 산출물에 대한 이해도</li> </ul>
인터페이스 구현 검증하기	3.1 구현된 인터페이스 명세서를 참조하여 구현 검증에 필요한 감시 및 도구를 준비할 수 있다. 3.2 인터페이스 구현 검증을 위하여 외부 시스템과의 연계 모듈 상태를 확인할 수 있다. 3.3 인터페이스 오류처리 사항을 확인하고 보고서를 작성할 수 있다.	



- 지식, 기술, 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> <li>• 네트워크 이해</li> <li>• 데이터베이스 이해</li> <li>• 디자인 패턴</li> <li>• 분산 프로그래밍 기법</li> <li>• 산업 분야별 업무 특성</li> <li>• 설계 모델링 기법</li> <li>• 설계 산출물 이해</li> <li>• 소프트웨어 개발 방법론</li> <li>• 소프트웨어 아키텍처 이해</li> <li>• 소프트웨어 테스트 이해</li> </ul>
기 술	<ul style="list-style-type: none"> <li>• IDE(Integrated Development Environment) 도구 활용 능력</li> <li>• UML(Unified Modeling Language) 모델링 도구 활용 능력</li> <li>• 기술영역별 미들웨어 및 솔루션 활용 능력</li> <li>• 단위테스트 도구 활용 능력</li> <li>• 데이터 보안 기술</li> <li>• 데이터 연계 및 이관 도구 활용 능력</li> <li>• 모델링 기술</li> <li>• 애플리케이션 프레임워크 활용 능력</li> <li>• 프로그래밍 언어 코드 검토 기술</li> <li>• 프로그래밍 언어 활용 능력</li> <li>• 프로그램 디버깅 능력</li> <li>• 프로그램 언어 활용 기술</li> <li>• 형상관리 도구 활용 능력</li> </ul>
태 도	<ul style="list-style-type: none"> <li>• 개발 수행에 요구되는 표준을 준수하려는 태도</li> <li>• 개발 팀원 간의 원활한 협업을 추구하는 태도</li> <li>• 고객의 요청을 적극적으로 수용하고자 하는 의지</li> <li>• 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도</li> <li>• 산출물 완성도를 위한 적극적인 태도</li> <li>• 성공적인 개발을 위한 의지</li> <li>• 정확성과 완전성을 기하고자 하는 의지</li> <li>• 주어진 과제를 완수하는 책임감</li> </ul>

## 통합 구현

### ● 훈련목표

- 통합 구현이란 모듈 간의 분산이 이루어진 경우를 포함하여 단위 모듈 간의 데이터관계를 분석하여 이를 기반으로 한 메커니즘을 통해 모듈 간의 효율적인 연계를 구현하고 검증하는 능력을 함양.

### - 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
연계 데이터 구성하기	1.1 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 데이터 연계 요구사항을 분석할 수 있다. 1.2 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 데이터를 식별할 수 있다. 1.3 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계를 위한 데이터 표준을 설계할 수 있다.	<ul style="list-style-type: none"> <li>• 연계 메커니즘에 대한 이해도</li> <li>• 연계 표준/프로토콜에 대한 이해도</li> <li>• 미들웨어, 인터페이스 소프트웨어에 대한 이해도</li> <li>• 모니터링 도구 사용법 및 모니터링 산출물에 대한 이해도</li> </ul>
연계 메커니즘 구성하기	2.1 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 특성을 고려하여 효율적 데이터 송수신 방법을 정의할 수 있다. 2.2 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 데이터 연계 요구 사항을 고려하여 연계주기를 정의할 수 있다. 2.3 개발하고자 하는 응용소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 데이터 연계 실패 시 처리방안을 정의할 수 있다. 2.4 응용소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 송수신 시 보안을 적용할 수 있다.	
내외부 연계 모듈 구현하기	3.1 구성된 연계 메커니즘에 대한 명세서를 참조하여 연계모듈 구현을 위한 논리적, 물리적 환경을 준비할 수 있다. 3.2 구성된 연계 메커니즘에 대한 명세서를 참조하여 외부 시스템과의 연계 모듈을 구현할 수 있다. 3.3 연계모듈의 안정적인 작동여부와 모듈 간 인터페이스를 통해 연동된 데이터의 무결성을 검증할 수 있다. 3.4 구현된 연계모듈을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화할 수 있다.	

- 지식, 기술, 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> <li>• 네트워크 이해</li> <li>• 데이터베이스 이해</li> <li>• 분산 프로그래밍 기법</li> <li>• 산업 분야별 업무 특성</li> <li>• 설계 모델링 기법</li> <li>• 소프트웨어 개발방법론</li> <li>• 소프트웨어 아키텍처 이해</li> <li>• 소프트웨어 테스트 이해</li> <li>• 프로그래밍 언어 활용 방법</li> </ul>
기 술	<ul style="list-style-type: none"> <li>• IDE(Integrated Development Environment) 도구 활용 능력</li> <li>• UML(Unified Modeling Language) 모델링 기술</li> <li>• 기술영역별 미들웨어 및 솔루션 활용 능력</li> <li>• 단위테스트 도구 활용 능력</li> <li>• 데이터 무결성 검증 능력</li> <li>• 데이터 보안 기술</li> <li>• 데이터 연계 및 이관 도구 활용 능력</li> <li>• 모델링 기술</li> <li>• 애플리케이션 프레임워크 활용 능력</li> <li>• 테스트케이스 작성 능력</li> <li>• 프로그램 디버깅 능력</li> </ul>
태 도	<ul style="list-style-type: none"> <li>• 개발 수행에 요구되는 표준을 준수하려는 태도</li> <li>• 개발 팀원 간의 원활한 협업을 추구하는 태도</li> <li>• 고객의 요청을 적극적으로 수용하고자 하는 의지</li> <li>• 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도</li> <li>• 산출물 완성도를 위한 적극적인 태도</li> <li>• 성공적인 개발을 위한 의지</li> <li>• 정확성과 완전성을 기하고자 하는 의지</li> <li>• 주어진 과제를 완수하는 책임감</li> </ul>

# 애플리케이션 배포

## ● 훈련목표

- 애플리케이션 배포란 애플리케이션 배포 환경을 구성하고, 구현이 완료된 애플리케이션의 소스 검증 및 빌드를 수행하여 운영 환경에 배포하는 능력을 함양.

## - 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
애플리케이션 배포 환경 구성하기	1.1 애플리케이션 빌드와 배포를 위한 환경 구성 방안을 계획할 수 있다. 1.2 애플리케이션 배포를 위한 도구와 시스템을 결정할 수 있다. 1.3 결정한 애플리케이션 배포 환경을 위한 도구와 시스템을 설치할 수 있다. 1.4 설치한 시스템과 도구 운영을 위해 상세 구성 및 설정을 할 수 있다.	<ul style="list-style-type: none"> <li>• 애플리케이션 개발 및 운영 환경에 대한 이해 여부 점검</li> <li>• 애플리케이션 배포 환경 구성을 위한 관련 도구 설치 및 설정 가능 여부 평가</li> </ul>
애플리케이션 소스 검증하기	2.1 정상적으로 작동하는 소프트웨어 빌드를 위해 형상관리 서버로부터 소스코드를 체크아웃 할 수 있다. 2.2 소스코드 검증 도구를 활용하여 애플리케이션에서 사용한 라이브러리, 소스, 로직 등의 오류가 있는지를 검증할 수 있다. 2.3 소스 코드의 환경 설정, 운영 환경 정보, 대상 시스템 정보 등에 오류가 있는지 확인할 수 있다.	<ul style="list-style-type: none"> <li>• 빌드 환경 구성에 따른 각종 도구 사용 능력 평가</li> <li>• 소스코드 검증 기법에 대한 이해를 바탕으로 도구 사용 능력 점검</li> </ul>
애플리케이션 빌드하기	3.1 애플리케이션 소스코드 검증 결과 문제가 없는 경우 해당 소스코드를 빌드시스템으로 이관할 수 있다. 3.2 애플리케이션 빌드 절차에 따른 빌드 스크립트를 작성할 수 있다. 3.3 작성한 빌드 스크립트 또는 도구를 활용하여 애플리케이션 빌드를 실행할 수 있다. 3.4 애플리케이션 빌드 실행 결과를 확인하여 정상적으로 완료되었는지를 확인 할 수 있다. 3.5 애플리케이션 빌드 실패 시 문제 내용과 원인을 파악하여 개발자에게 설명 할 수 있다.	<ul style="list-style-type: none"> <li>• 애플리케이션 빌드 및 배포 절차에 대한 이해 및 도구 사용 능력 확인</li> <li>• 애플리케이션 빌드 및 배포 스크립트 작성 실습 및 점검</li> <li>• 애플리케이션 빌드 및 배포 절차에 따라 빌드 실행 능력 점검</li> </ul>
애플리케이션 배포하기	4.1 애플리케이션 실행 환경에 대한 정보를 확인할 수 있다. 4.2 애플리케이션 배포 절차에 따라 운영환경에 적용할 수 있다. 4.3 애플리케이션 배포 후 정상적으로 작동하는지를 확인할 수 있다. 4.4 애플리케이션 배포 결과 문제가 발생했을 경우 적용 내용을 이전 상태로 복원할 수 있다.	<ul style="list-style-type: none"> <li>• 애플리케이션 빌드 결과에 대한 분석 및 이해 능력 점검</li> <li>• 애플리케이션 빌드 실패 시 원복 수행 절차 이해 및 실시 여부 확인</li> </ul>

## - 지식, 기술, 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> <li>• 배포 도구 설치 및 사용법</li> <li>• 빌드 도구 설치 및 사용법</li> <li>• 소스코드 검증 기법</li> <li>• 소스코드 검증 기준</li> <li>• 소스코드 검증 도구</li> </ul>

구 분	주 요 내 용
	<ul style="list-style-type: none"> <li>• 소스코드 동적 검증 분석 기법</li> <li>• 소스코드 정적 검증 분석 기법</li> <li>• 애플리케이션 배포 환경 구성 지식</li> <li>• 애플리케이션 빌드 스크립트 문법</li> <li>• 애플리케이션 빌드 시스템 가이드</li> <li>• 애플리케이션 빌드 절차</li> <li>• 애플리케이션 운영 플랫폼 정보</li> <li>• 애플리케이션 운영 환경 배포 절차</li> <li>• 애플리케이션 운영 환경 정보</li> <li>• 애플리케이션 원복 절차</li> <li>• 형상관리 도구 사용법</li> <li>• 형상관리 도구 설치 방법</li> <li>• 형상관리 지침</li> </ul>
기 술	<ul style="list-style-type: none"> <li>• IDE(Integrated Development Environment) 도구 활용 능력</li> <li>• 기본 명령어 활용 능력</li> <li>• 배포 스크립트 작성 기술</li> <li>• 빌드 검증 기술</li> <li>• 빌드 스크립트 작성 기술</li> <li>• 빌드 실행 기술</li> <li>• 빌드 자동화 도구 활용 능력</li> <li>• 소스코드 검증 결과 분석 기술</li> <li>• 소스코드 검증 기술</li> <li>• 소스코드 검증 도구 사용 기술</li> <li>• 애플리케이션 원복 실행 도구 사용 능력</li> <li>• 애플리케이션 원복 절차 기술</li> <li>• 운영 배포 결과 확인 능력</li> <li>• 형상관리 도구 활용 능력</li> <li>• 환경 설정 능력</li> </ul>
태 도	<ul style="list-style-type: none"> <li>• 검증이 완료된 소스코드는 빌드 시스템으로 정확하게 이관하는 태도</li> <li>• 다양한 도구와 연동을 통한 최적의 배포 환경을 구성하고자 하는 태도</li> <li>• 다양한 분석도구를 다루면서 효율적으로 활용하고자 하는 자세</li> <li>• 문제 발생 애플리케이션에 대한 원상 복구 여부를 판단하는 자세</li> <li>• 배포 시 문제 발생 시 즉시 원인 파악을 하고자 하는 자세</li> <li>• 빌드 결과를 이해하고 문제 발생 시 즉시 조치하는 태도</li> <li>• 빌드 절차를 준수하여 성공적으로 빌드를 수행하고자 하는 자세</li> <li>• 빌드 진행 상태를 모니터링 하면서 결과를 확인하는 태도</li> <li>• 소스코드 검증 결과 문제 발생시 해결하고자 하는 적극적인 자세</li> <li>• 애플리케이션 배포 이전에 빌드 후 패키징 결과물의 이상 여부를 검증하는 자세</li> <li>• 애플리케이션 배포 이전에 정확한 환경 정보로 구성했는지를 확인하는 태도</li> <li>• 애플리케이션 배포 후 정상 작동 여부를 확인하는 자세</li> <li>• 애플리케이션 원복 결정 시 즉시 절차에 따라 실행하는 태도</li> <li>• 애플리케이션의 특성을 고려한 배포 환경을 구성하고자 하는 자세</li> <li>• 애플리케이션의 품질 향상을 위해 다양한 관점에서 소스코드를 검증하는 자세</li> <li>• 자동화 도구 외 수작업으로 검토 및 리뷰를 통해 검증을 실시하는 자세</li> <li>• 자동화 도구를 활용하여 소스코드를 검증하는 노력</li> <li>• 자동화 도구를 효율적으로 다룰 수 있는 자세</li> </ul>

## 애플리케이션 테스트 관리

### ● 훈련목표

- 애플리케이션 테스트 관리란 요구사항대로 응용소프트웨어가 구현되었는지를 검증하기 위해서 테스트케이스를 작성하고 개발자 통합 테스트를 수행하여 애플리케이션의 성능을 개선하는 능력을 함양.

### - 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
애플리케이션 테스트케이스 설계하기	1.1 개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상 과 범위를 결정하여 테스트케이스를 작성할 수 있다. 1.2 개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상 과 범위가 적용된 시나리오를 정의할 수 있다. 1.3 애플리케이션 테스트 수행에 필요한 테스트 데이터, 테스트 시작 및 종료조건 등을 준비할 수 있다.	<ul style="list-style-type: none"> <li>• 테스트케이스 작성 방 법</li> <li>• 테스트 수행에 필요한 준비사항</li> <li>• 애플리케이션 통합 테 스트 수행</li> </ul>
애플리케이션 통합 테스트하기	2.1 개발자 통합테스트 계획에 따라 통합 모듈 및 인터페이스가 요구사항 을 충족하는지에 대한 테스트를 수행할 수 있다. 2.2 개발자 통합테스트 수행 결과 발견된 결함에 대한 추이 분석을 통하 여 잔존결함을 추정할 수 있다. 2.3 개발자 통합테스트 결과에 대한 분석을 통해 테스트의 충분성 여부를 검증하고, 발견된 결함에 대한 개선 조치사항을 작성할 수 있다.	<ul style="list-style-type: none"> <li>• 결함에 대한 개선 조치 사항</li> <li>• 성능 저하 요인</li> <li>• 애플리케이션 성능 개 선</li> <li>• 소스코드 품질 분석 도 구 활용</li> </ul>
애플리케이션 성능 개선하기	3.1 애플리케이션 테스트를 통하여 애플리케이션의 성능을 분석하고, 성능 저하요인을 발견할 수 있다. 3.2 코드 최적화 기법, 아키텍처 조정 및 호출 순서 조정 등을 적용하여 애플리케이션 성능을 개선할 수 있다. 3.3 프로그래밍 언어의 특성에 대한 이해를 기반으로 소스코드 품질 분석 도구를 활용하여 애플리케이션 성능을 개선할 수 있다.	

- 지식, 기술, 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> <li>• 국제 표준 품질 특성</li> <li>• 산업 분야별 업무특성</li> <li>• 성능 평가 기준</li> <li>• 소스코드 인스펙션</li> <li>• 소프트웨어 아키텍처 국제 표준(IEEE 1471)</li> <li>• 소프트웨어 아키텍처 이해</li> <li>• 알고리즘</li> <li>• 자료구조</li> <li>• 테스트 레벨별 테스트 접근 방법</li> <li>• 테스트 지식 체계(ISTQB Syllabus, CSTE CBOK)</li> <li>• 테스트케이스 설계 기법</li> <li>• 프로그래밍 개발 표준</li> <li>• 프로그래밍 언어 활용 방법</li> </ul>
기 술	<ul style="list-style-type: none"> <li>• APM(Application Performance Management) 활용 능력</li> <li>• IDE(Integrated Development Environment) 도구 활용 능력</li> <li>• 성능 측정 도구 활용 능력</li> <li>• 소스코드 검증 도구 사용 기술</li> <li>• 컴파일러 사용 능력</li> <li>• 코드 검토 능력</li> <li>• 테스트 도구 활용 능력</li> <li>• 통합테스트 결과보고서 작성 기술</li> <li>• 통합테스트 계획서 작성 기술</li> <li>• 표준 단어, 용어, 도메인, 코드 정의 능력</li> <li>• 프로그래밍 언어 코드 검토 기술</li> <li>• 프로그램 디버깅 능력</li> <li>• 형상관리 도구 활용 능력</li> </ul>
태 도	<ul style="list-style-type: none"> <li>• 개발 수행에 요구되는 표준을 준수하려는 태도</li> <li>• 고객 및 제품 요구사항에 대하여 적극적으로 분석하는 자세</li> <li>• 기술 및 제품 동향에 관한 각종 정보 수집에 대한 적극성</li> <li>• 적극적이고 세밀히 분석하는 자세</li> <li>• 제품에 대한 책임감 및 불량에 대한 재발 방지를 위해 노력하는 태도</li> <li>• 제품의 품질 및 고객 우선주의 태도</li> </ul>