

3 과목 | 시스템 분석 및 설계

20.8, 20.6, 19.8, 19.4, 18.8, 18.3, 17.8, 17.5, 17.3, 16.8, 16.5, 16.3, 15.8, 15.5, 15.3, 14.8, 14.5, 4.3, 13.8, ...

핵심

111 시스템의 개요

- **정의** : 공통의 목적을 달성하기 위하여 여러 가지 상호 관련된 요소들을 유기적으로 결합한 것
- **특성**

목적성	서로 다른 기능을 가지고 있는 시스템의 각 구성 요소들이 어떤 하나의 공통된 최종 목표에 도달하고자 하는 특성
자동성	어떤 조건이나 상황의 변화에 대응하여 스스로 대처할 수 있는 특성
제어성	시스템이 오류 없이 그 기능을 발휘하기 위하여 정해진 규정이나 한계, 또는 궤도로부터 이탈되는 현상을 사전에 감지하여 그것을 바르게 수정해 가는 특성
종합성	항상 관련된 다른 시스템과 상호 의존 관계로 통합되는 특성

- **기본 요소**

입력(Input)	처리할 데이터, 처리 방법, 처리 조건을 시스템에 투입하는 것
처리(Process)	입력된 데이터를 처리 방법과 조건에 따라 변환하거나 가공하는 것
출력(Output)	처리된 결과를 시스템에서 산출하는 것
제어(Control)	자료가 입력되어 출력될 때까지의 처리 과정이 올바르게 행해지는지 감독하는 것
피드백(Feedback)	출력된 결과가 예정된 목적을 만족시키지 못한 경우 목적 달성을 위해 반복 처리하는 것

19.8, 18.8, 18.4, 16.5, 11.8, 07.3, 05.5, 04.5, 02.9, 02.5, 01.3, 99.8, 99.4

핵심

112 시스템 분석가(SA)

- 시스템의 전반적인 흐름과 사용자들의 요구 사항을 파악하고 해결책을 마련하는 사람이다.
- 기업의 목적과 현행 시스템의 문제점을 정확히 이해하고 해결책을 제시할 수 있어야 한다.
- 업무 내용이나 시스템에 대한 분석 능력이 있어야 한다.
- 컴퓨터 기술과 관리 기법을 알아야 한다.
- 시간 배정과 계획 등을 빠른 시간 내에 파악할 수 있어야 한다.
- 컴퓨터 하드웨어와 소프트웨어에 대한 전반적인 지식을 가져야 한다.
- 업계의 동향 및 관계 법규 등을 파악할 수 있어야 한다.
- 창조력, 응용력, 현장 분석 경험이 있어야 한다.
- 사용자와 프로그래머, 경영진 간의 의사소통을 원활히 하는 해결사 역할을 수행할 수 있어야 한다.

19.8, 19.4, 19.3, 18.4, 16.8, 16.3, 12.8, 09.8, 09.3, 08.9, 07.9, 07.5, 06.9, 06.5, 06.3, 05.4, 05.3, ...

핵심**113 시스템 개발 생명 주기(SDLC)**

- 시스템을 개발하는 과정에서 공통적으로 반복되는 단계를 말하는 것이다.
- **시스템 개발 생명 주기의 순서** : 시스템 조사 → 시스템 분석 → 시스템 설계 → 시스템 구현 → 테스트 → 시스템 운용 → 유지보수

시스템 조사	현행 시스템의 상태와 문제점을 파악하고 해결 방안을 제안하는 단계로, 예비 조사와 기초 조사로 나뉨
시스템 분석	<ul style="list-style-type: none"> • 조사 단계에서 조사된 사용자의 요구 사항과 현행 시스템의 문제점을 명확히 파악하여 요구 분석 명세서를 작성하는 과정으로, 기능 분석, 예비 설계, 비용 효과 분석 순으로 진행됨 • 기업 환경 조사, 현장 조사, 기업이 필요로 하는 기능과 활동을 조사하고, 자료 흐름도, 자료 사전, 소단위 명세서 등 기능 분석을 위한 도구를 사용하여 모델을 설계함
시스템 설계	시스템 분석에 의해 정의된 시스템 요구 분석 명세서를 토대로 하여 새로운 시스템을 구체화하는 단계 <ul style="list-style-type: none"> • 기본 설계 : 분석 결과에 따라 사용자 입장에서 시스템 전체를 개괄적으로 설계하는 단계 • 상세 설계 : 각 기능의 논리적인 절차를 확정하고, 구체적인 입·출력 내용, 코드 설계, 파일의 구체적 사양을 결정하기 위한 단계
시스템 구현	설계 단계에서 산출된 설계 사양에 따라 프로그래밍 언어를 이용하여 원시 코드를 작성하는 단계로, 프로그래밍(Programming) 또는 코딩(Coding)이라고도 함
테스트	사용자의 요구에 따라 시스템이 구현되었는지 검증하는 단계로, 테스트의 종류에는 통합 테스트, 시스템 테스트, 인수 테스트가 있음
시스템 운용(이행)	개발된 시스템을 실제 업무 처리에 적용하여 활용하는 단계
유지보수	<ul style="list-style-type: none"> • 시스템 개발 단계 중 가장 많은 비용이 투입되는 단계 • 종류 : 수정 유지보수, 적응 유지보수, 완전 유지보수, 예방 유지보수

- **T.M.Ho의 시스템 개발 주기 순서** : 목적 설정 → 상황 조사 → 현행 시스템의 연구 → 사용자 요구 사항 분석 → 대안 평가 → 새로운 하드웨어와 소프트웨어 선택 → 새 시스템의 설계 → 새 시스템의 구축 → 새 시스템의 인도

20.8, 20.6, 19.4, 19.3, 18.8, 18.4, 17.8, 17.5, 17.3, 16.8, 16.3, 15.8, 15.5, 15.3, 14.8, 14.5, 14.3, 13.8, 13.6, ...

핵심**114 코드 설계**

- 코드는 컴퓨터를 이용하여 자료를 처리하는 과정에서 분류·조합 및 집계를 용이하게 하고, 특정 자료의 추출을 쉽게 하기 위해서 사용하는 기호이다. 또한 어떤 단위별 수치를 알거나 파일을 체계화하기 위해서 사용된다.

- **코드의 기능**

3대 기능	분류, 식별, 배열
그 밖의 기능	간소화, 표준화, 암호화, 단순화, 연상(표의성), 오류 검출, 구별, 추출

- **코드 설계 순서** : 코드화 대상 선정 → 코드화 목적의 명확화 → 코드 부여 대상 수 확인 → 사용 범위 결정 → 사용 기간 결정 → 코드화 대상의 특성 분석 → 코드 부여 방식의 결정 → 코드의 문서화

코드화 대상 선정	<ul style="list-style-type: none"> 어떤 항목을 대상으로 코드화할 것인지를 결정하는 단계 정보의 체계화 유무, 정보처리 효율성 유무, 정보 호환성 유무, 정보 표준화 유무 등을 고려하여 대상 항목을 결정함
코드화 목적의 명확화	무엇을 위해 코드가 필요한지 목적을 명확하게 밝히고, 코드에 부여할 기능을 결정함
코드 부여 대상 수 확인	현장 조사 자료를 근거로 코드화 대상을 확인하되, 현재의 자료량과 이후 동향에 대해서도 고려함
사용 범위 결정	코드 대상 항목에 대하여 설계된 코드의 사용이 컴퓨터 처리에 한정되는가, 해당 업무에만 한정되는가, 관련 부문의 업무에 공통으로 사용되는가, 기업 전체에 사용되는가, 관련 있는 타기업 또는 공공 기관이 공통으로 사용할 것인지 등을 결정함
사용 기간 결정	만든 코드를 영구적으로 사용할 것인지, 한시적으로 사용할 것인지를 결정함
코드화 대상의 특성 분석	코드화 대상 항목의 사용 경로, 코드 변경 유무, 코드의 추가 및 삭제의 빈도를 등을 면밀히 분석함
코드 부여 방식의 결정	자료의 특성에 따라 코드 체계, 체크 디지털의 사용 여부, 코드 자릿수, 코드 부여 요령 등을 결정함
코드의 문서화	코드를 실제 사용할 프로그래머가 이해할 수 있도록 코드와 관련한 전반적인 내용을 문서화하여 작성함

- **코드 설계시 유의 사항** : 기계 처리의 용이성, 취급의 용이성, 분류의 편리성(공통성, 체계성), 확장성, 단순성, 고유성, 표의성, 함축성

20.8, 20.6, 19.8, 19.4, 18.8, 18.4, 18.3, 17.8, 17.5, 17.3, 16.5, 16.3, 15.8, 15.5, 15.3, 14.8, 14.5, 14.3, 13.8, ...

핵심

115 코드의 종류

순서 코드(Sequence Code)	<ul style="list-style-type: none"> 자료의 발생 순서, 크기 순서 등 일정 기준에 따라서 최초의 자료부터 차례로 일련 번호를 부여하는 방법(순차 코드, 일련 번호식 코드) 항목수가 적고, 변경이 적은 자료에 적합함 일정 순서대로 코드를 할당하기 때문에 기억 공간의 낭비가 없고 자릿수가 가장 짧음 단순·명료하고, 이해하기 쉬움 발생 순서에 따른 코드인 경우 추가가 매우 편리함(확장성 용이) 고유성이 있으므로 기억이 용이함 코드 중간에 누락된 자료를 삽입하기 어려움(융통성이 적음)
구분 코드(Block Code)	<ul style="list-style-type: none"> 코드화 대상 항목 중에서 공통성이 있는 것끼리 블록으로 구분하고, 각 블록 내에서 일련 번호를 부여하는 방법

	<ul style="list-style-type: none"> • 자릿수가 비교적 짧고, 블록별로 식별과 분류가 쉬움 • 블록마다 여유 코드를 두어 코드의 추가를 쉽게 할 수 있지만, 여유 코드는 코드 낭비의 요인이 되기도 함 • 기계 처리가 어려움
그룹 분류식 코드(Group Classification Code)	<ul style="list-style-type: none"> • 코드화 대상 항목을 일정 기준에 따라 대분류, 중분류, 소분류 등으로 구분하고, 각 그룹 안에서 일련 번호를 부여하는 방법 • 분류 기준이 명확한 경우에 이용도가 높으며, 기계 처리에 가장 적합함 • 각 자리가 특정한 의미를 가지고 있어 구분별 분류와 집계가 편리하며 그 의미가 명확함 • 여유 부분이 있어 자료 추가를 쉽게 할 수 있음 • 자릿수가 길어질 수 있음
10진 코드(Decimal Code)	<ul style="list-style-type: none"> • 코드화 대상 항목을 0~9까지 10진 분할하고, 그 각각에 대하여 다시 10진 분할하는 방법을 필요한 만큼 반복하는 코드(도서 분류식 코드) • 도서관에서 도서 정리 목적으로 널리 사용함 • 코드 체계가 명확하고, 무한대로 확장이 가능함 • 삽입 및 추가가 용이하고, 배열이나 집계가 가능함 • 분류 항목이 10개 이상일 때는 비효율적임 • 자릿수가 길어질 수 있고, 기계 처리가 어려움
표의 숫자 코드(Significant Digit Code)	<ul style="list-style-type: none"> • 코드화 대상 항목의 성질, 즉 길이, 넓이, 부피, 지름, 높이 등의 물리적 수치를 그대로 코드에 적용시키는 방법(유효 숫자 코드) • 코드에 대상체의 성질을 그대로 표시하므로 기억하기 쉬움 • 코드의 추가 및 삭제가 용이함 • 자릿수가 길어질 수 있고, 기계 처리에 불편함
연상 코드(Mnemonic Code)	<ul style="list-style-type: none"> • 코드화 대상 항목의 명칭이나 약호와 관계있는 숫자나 문자, 기호를 이용하여 코드를 부여하는 방법(기호 코드) • 코드만 보고도 대상 품목을 쉽게 연상할 수 있음 • 지명, 물건명, 상호명에 많이 적용함 • 자릿수가 길어질 수 있음

20.8, 20.6, 19.8, 18.8, 18.4, 18.3, 17.5, 16.5, 15.8, 15.3, 13.6, 12.8, 11.8, 10.3, 09.3, 08.9, 08.5, 08.3, ...

핵심**116 코드의 오류 발생 형태**

- **필사 오류(Transcription Error)** : 입력 시 임의의 한 자리를 잘못 기록한 경우 발생(오자 오류)
- **전위 오류(Transposition Error)** : 입력 시 좌우 자리를 바꾸어 기록한 경우 발생
- **이중 오류(Double Transposition Error)** : 전위 오류가 2개 이상 발생한 경우
- **생략 오류(Omission Error)** : 입력 시 한 자리를 빼놓고 기록한 경우 발생
- **추가 오류(Addition Error)** : 입력 시 한 자리를 더 추가하여 기록한 경우 발생
- **임의 오류(Random Error)** : 위의 오류가 2가지 이상 결합하여 발생한 경우

20.6, 19.8, 19.4, 19.3, 18.8, 18.4, 18.3, 17.5, 16.8, 16.3, 15.8, 15.5, 15.3, 14.8, 14.5, 14.3, 13.8, 13.6, 13.3, ...

핵심**117 입력 설계**

- **입·출력 설계의 표준화** : 방식의 표준화, 매체의 표준화, 형식의 표준화, 등록의 표준화, 코드의 표준화
- **입력 설계 순서** : 입력 정보의 발생 → 입력 정보의 수집 → 입력 정보의 매체화 → 입력 정보의 투입 → 입력 정보의 내용

입력 정보의 발생에 대한 설계	<ul style="list-style-type: none"> • 입력 정보의 명칭과 작성 목적, 입력 정보의 발생자와 발생 장소, 입력 정보의 발생 방법 및 발생 형태, 입력 정보의 발생 주기 및 시기와 발생 건수, 오류 검사 방법 결정 • 발생한 정보를 언제, 어디서, 누가, 무슨 용도로 사용하냐에 따라 다르게 설계됨
입력 정보의 수집에 대한 설계	수집 담당자와 수집 장소, 수집 방법과 형태, 수집 경로와 수집 주기 및 시기, 수집 시의 오류 검사 방법 결정
입력 정보의 매체화에 대한 설계	매체화 담당자와 장소, 매체화 장치, 레코드의 길이와 형식, 매체화 주기 및 시기, 매체화 시의 오류 검사 방법, 입력 정보의 형태 결정
입력 정보의 투입에 대한 설계	입력 매체의 모양과 서식, 입력 정보의 투입 주기 및 시기, 투입 장치, 투입 시의 오류 검사 방법 결정
입력 정보의 내용에 대한 설계	입력 항목의 배열 순서와 항목명, 입력 항목의 자릿수와 문자 구분, 입력 정보의 오류 검사 방법 결정

잠깐만요! 입력 매체 장치 선택 시 검토 사항

시스템의 이행 방법 및 운용 비용, 입력 정보 발생 분야에서의 업무 특성, 입력 매체와 매체화 장치의 특성, 출력 정보를 이용할 시점에 맞게 투입

• 데이터 입력 방식

집중 매체화 시스템	발생한 데이터를 전표 상에 기록하고, 일정 시간 단위로 일괄 수집하여 입력 매체에 수록하는 방식
분산 매체화 시스템	그 데이터가 발생한 장소에서 입력 정보로 매체화하고 그 입력 매체를 주기적으로 수집하여 컴퓨터에 입력시키는 방식
턴 어라운드 시스템	<ul style="list-style-type: none"> • 출력 시스템과 입력 시스템이 일치된 방식 • 입력된 자료가 처리되어 일단 출력된 후 이용자를 경유하여 다시 재입력되는 방식으로, 공과금, 보험료 징수 등의 지로 용지를 처리하는 데 사용됨
일괄 처리 시스템	일정 시간 동안 수집된 변동 자료를 컴퓨터의 입력 자료로 만들었다가 필요한 시점에서 이 자료들을 입력하여 실행한 후 그 결과를 출력시키는 방식의 시스템

잠깐만요! 대화형 입·출력 방식

- **프롬프트 방식** : 프롬프트가 위치한 곳에 사용자가 직접 명령어나 자료를 입력하는 방식으로 명령어 처리 방법의 구현이 쉽고, 복잡한 명령도 사용 가능함
- **메뉴 방식** : 시스템에서 사용되는 명령어나 선택 사항을 메뉴로 구성하여 화면에 진열하는 방식으로 배우기 쉽고 편리함
- **항목 채우기 방식** : 정보를 직접 입력할 수 없는 메뉴 방식의 단점을 개선한 방식으로, 화면에 항목의 이름과 함께 입력 영역을 같이 진열하여 사용자가 입력 영역에 값을 직접 입력하는 방식

- **아이콘 방식** : 화면에 여러 개의 항목을 진열하고 그 중의 하나를 선택 도구로 지정하여 직접 실행하는 방식

19.8, 19.4, 19.3, 18.3, 17.8, 17.3, 16.5, 16.3, 15.8, 15.5, 15.3, 14.8, 14.5, 13.6, 13.3, 12.8, 12.5, 12.3, 11.6, ...

핵심

118 출력 설계/COM 시스템

- **출력 설계 순서** : 출력 정보의 내용 → 출력 정보의 매체화 → 출력 정보의 분배 → 출력 정보의 이용

출력 정보 내용에 대한 설계	출력할 항목과 명칭, 출력 항목의 배열 순서, 크기, 자릿수, 출력 항목의 문자 표현 방법, 출력 항목에 대한 집계 방법, 출력 정보의 오류 검사 방법을 결정함
출력 정보 매체화에 대한 설계	출력 형식, 출력 매체 및 장치, 출력 정보의 양과 출력 복사의 매수, 출력 장소와 출력 시기 및 주기, 출력 배열 순서, 장치의 특성, 작동의 용이성을 결정함
출력 정보 분배에 대한 설계	분배 책임자, 분배 방법 및 형태, 분배 경로, 분배 주기 및 시기에 대해 결정함
출력 정보 이용에 대한 설계	출력 정보명과 출력 정보의 이용 목적, 출력 정보의 이용자와 이용 경로, 이용 주기 및 시기, 기밀성의 유무와 보존에 대해 결정함

- **COM 시스템** : 출력 정보를 마이크로 필름에 수록하는 것으로, 축소 보관과 반영구적 보존이 가능하고, 지도, 설계 도면, 학적부, 병원의 기록 등을 보존, 검색, 관리하기에 적합한 방식

20.8, 20.6, 17.8, 16.3, 15.3, 13.6, 13.3, 12.8, 12.3, 11.6, 11.3, 10.9, 10.5, 09.8, 08.9, 08.5, 08.3, 07.3, ...

핵심

119 데이터 파일의 종류

- **마스터 파일(Master File)**
 - 전표 처리에서의 원장 또는 대장에 해당하는 파일로, 자료 관리의 중추적 역할을 담당하며 기본이 되는 파일임
 - 트랜잭션 파일에 의해 갱신됨
- **트랜잭션 파일(Transaction File)** : 거래 내역이나 변동 내용 등 일시적인 성격을 지닌 정보를 기록하는 파일로, 마스터 파일을 갱신하거나 조회할 때 사용함
- **요약 파일(Summary File)** : 다른 파일의 중요 내용이나 합계를 요약해 놓은 파일로, 집계용으로 많이 사용됨
- **히스토리 파일(History File)** : 후일 통계 처리에 사용할 자료나 사고 발생 시 마스터 파일 등을 원상 복구시키기 위한 자료를 보존한 파일로, 현재까지 변화된 정보를 포함함
- **트레일러 파일(Trailer File)** : 마스터 파일을 목적에 따라 여러 개의 파일로 나누었을 때 가장 끝부분에 해당하는 파일

- **원시 파일(Source File)** : 입력 데이터를 알맞은 매체에 맞게 변환하여 만든 파일
- **백업 파일(Backup File)** : 만일의 사고에 대비하여 마스터 파일을 백업해 놓은 파일

19.8, 18.8, 18.3, 17.5, 16.5, 15.8, 14.8, 11.8, 10.5, 09.3, 07.9, 06.3, 05.9, 03.5

핵심

120 순차 편성(파일)

정의 및 특징

- 입력되는 데이터들을 논리적인 순서에 따라 물리적 연속 공간에 순차적으로 기록하는 방식이다.
- 주로 순차 접근이 가능한 자기 테이프에서 사용된다.
- 급여 관리 등과 같이 변동 사항이 크지 않고 기간별로 일괄 처리를 주로 하는 경우에 적합하다.

순차 파일의 장점

- 기록 밀도가 높아 기억공간을 효율적으로 사용할 수 있다.
- 매체 변환이 쉬워 어떠한 매체에도 적용할 수 있다.
- 레코드가 키 순서대로 편성되어 취급이 용이하다.
- 레코드를 기록할 때 사용한 키 순서대로 레코드를 처리하는 경우, 다른 편성법보다 처리 속도가 빠르다.

순차 파일의 단점

- 파일에 새로운 레코드를 삽입하거나 삭제하는 경우 파일 재구성을 위해 전체를 복사해야 하므로 시간이 많이 소요된다.
- 데이터 검색 시 처음부터 순차적으로 검색하기 때문에 검색 효율이 낮고, 응답시간이 느리다.