정보처리기사

제4과목 소프트웨어공학

01 소프트웨어공학의 개요1

문혜영교수

목차

1 소프트웨어

2 소프트웨어 특징

3 소프트웨어 위기

4 소프트웨어 공학

- 소프트웨어의 특징
  - ✓상품성
  - ✓견고성
  - ✓복잡성
  - ✓순응성
  - ✓비가시성
  - ✓비마모성
  - ✓비제조성
  - ✓비과학성

- 시스템
  - ✓구성요소 : 입력, 처리, 출력, 제어, 피드백

- 소프트웨어 위기

✓원인 : 소프트웨어 개발 속도가 하드웨어 개발 속도를 따라가지 못해 소프트웨어에 대한 사용자들의 요구사항을 처리할 수 없는 문제가 발생했다.

✓결과 : 개발 인력의 부족과 인건비 상승,
소프트웨어에 대한 사용자들이 요구사항 증가

문제

- 소프트웨어 위기의 현상으로 보기 어려운 것은?

① 프로젝트 개발 일정과 예산 측정의 어려움

② 소프트웨어 유지 보수 비용의 증가

③ 소프트웨어 개발 적체 현상

④ 소프트웨어 개발 인력의 증가

정답 4

- 소프트웨어 공학의 개념
  - IEEE의 소프트웨어 공학 표준용어 사전: 소프트웨어의 개발, 운용, 유지보수, 폐기 처분에 대한 체계적인 접근 방안
  - Fairley : 지정된 비용과 기간 내에 소프트웨어를 체계적으로 생산하고 유지 보수하는 데 관련된 기술적이고 관리적인 원리
  - Boehm : 과학적인 지식을 소프트웨어 설계와 제작에 응용하는 것이며 이를 개발, 운용, 유지 보수하는 데 필요한 문서 작성 과정
  - 소프트웨어 공학은 가장 경제적인 방법으로 양질의 제품을 생산하는 것이다.

- 소프트웨어 공학의 기본 원칙
  - ✓현대적인 프로그래밍 기술을 계속적으로 적용해야 한다.
  - ✓개발된 소프트웨어 품질이 유지되도록 지속적으로 검증해야 한다.
  - ✓소프트웨어 개발 관련 사항 및 결과에 대한 명확한 기록을 유지해야 한다.

- 공학적으로 잘 작성된 소프트웨어
  - ✓사용법, 구조의 설명, 성능 기능이 이해하기 쉬워야 한다.
  - ✓사용자 수준에 따른 적당한 사용자 인터페이스를 제공해야 한다.
  - ✓유지보수가 용이해야 한다.
  - ✓실행 속도가 빠르고 기억용량을 적게 차지해야 한다.

문제

- 소프트웨어 공학에 대한 설명으로 거리가 먼것은?
  ① 소프트웨어 공학이란 소프트웨어의 개발, 운용, 유지보수 및 파기에 대한 체계적인 접근방법이다.
  ② 소프트웨어 공학은 소프트웨어 제품의 품질을 향상시키고 소프트웨어 생산성과 작업 만족도를 증대시키는 것이 목적이다.
  ③ 소프트웨어 공학의 궁극적 목표는 최대의 비용으로 계획된 일정보다 가능한 빠른시일 내에 소프트웨어를 개발하는 것이다.
  ④ 소프트웨어 공학은 신뢰성 있는 소프트웨어을 경제적인 사용으로 획득하기 위해 공학적 원리를 정립하고 이를 이용하는 학문이다.

정답 3

정보처리기사

제4과목 소프트웨어공학
02 소프트웨어공학의 개요2 (생명주기)

문혜영교수

목차

1 생명주기

2 폭포수형

3 프로토타입형

4 나선형

- 소프트웨어 생명주기의 역할
  - ✓프로젝트 비용 산정과 개발 계획을 수립할 수 있는 기본 골격이 된다.
  - ✓프로젝트 진행 방향을 명확하게 파악하게 한다.
  - ✓용어 및 기술의 표준화를 가능하게 한다.
  - ✓프로젝트 관리를 용이하게 한다.
  - ✓여러 소프트웨어 간에 상호 일관성을 유지하게 한다.

- 일반적인 소프트웨어 생명주기
  - ✓정의 단계
  - ✓개발 단계
    - 설계단계
    - 구현단계 : 코딩하고 번역하는 단계
    - 테스트단계
  - ✓유비보수 단계 : 시간과 비용이 가장 많이 든다.

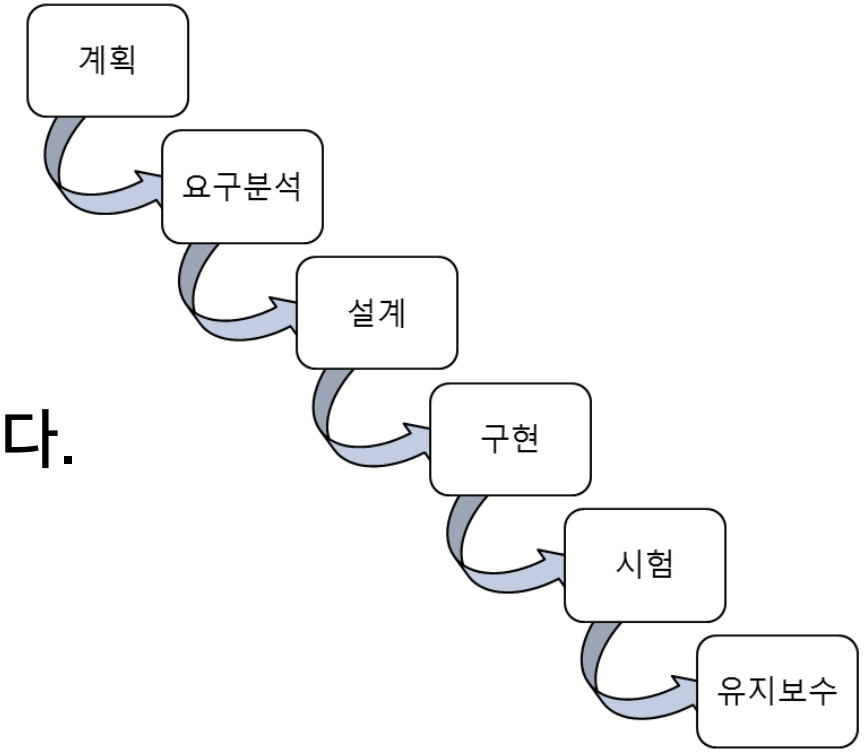
문제

- 현재 소프트웨어 개발 중 가장 많은 비용이 요구되는 단계는?
  ① 분석
  ② 설계
  ③ 구현
  ④ 유지보수

정답 4

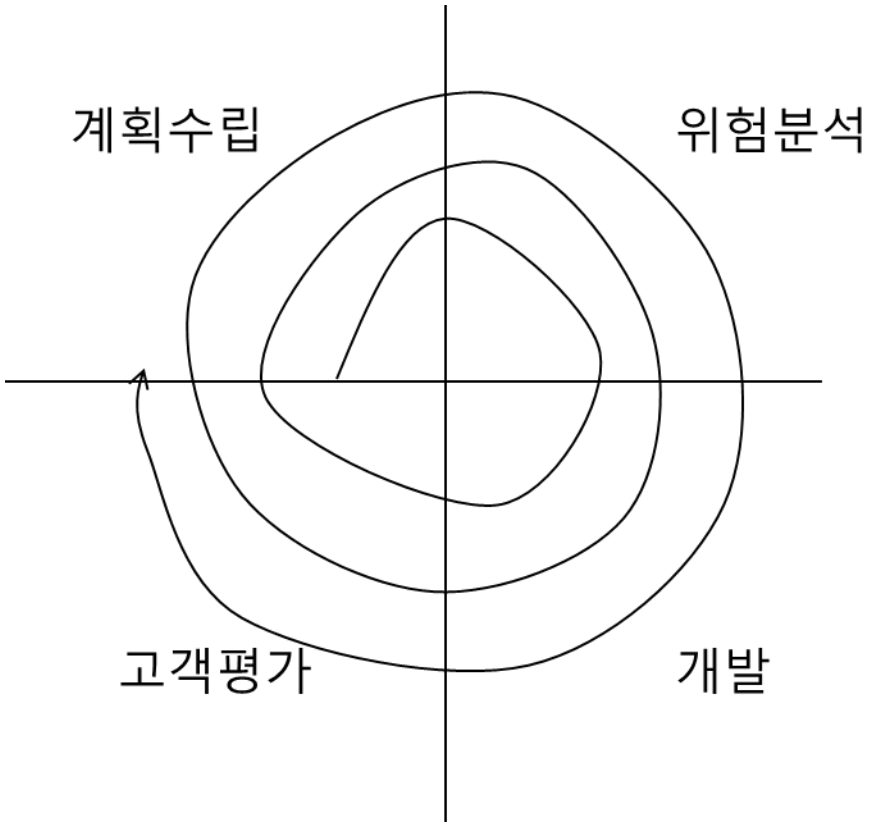
소프트웨어 생명주기

- 폭포수형 모형
  - ✓각 단계를 마무리 하고 그 결과를 검토하여 다음단계를 진행하며 이전단계로 넘어갈 수 없다.
  - ✓고전적 생명주기, 전통적인 방법이다
  - ✓선형 순처적 모형이다. 순차적으로 진행된다.
  - ✓장점 –성공사례가 많다. 단계별 정의가 분명하다.
  - ✓단점-요구사항 변경이 어렵다.



- 프로토타입 모형
  - ✓요구사항을 정확히 파악하기위해 견본(시제품)을 만들어 최종결과물을 예측한다.
  - ✓사용자와 시스템사이의 인터페이스에 중점을 두어 개발한다.
  - ✓폭포수형이 개발이 완료된 시점에서 오류가 발견되는 단점을 보완한것이다.
  - ✓개발단계안에서 유지보수가 이루어지기때문에 별도의 유지보수단계가 없다.
  - ✓요구분석-설계-프로토타입구축-고객평가-프로토타입조정-구현
  - ✓의뢰자나 개발자 모두에게 공동의 참조모델을 제공한다.

- 나선형 모형
  - ✓폭포수형과 프로토타입모형의 장점에 위험분석 기능을 추가한 것이다.
  - ✓대규모 프로젝트에 적합하다.
  - ✓점진적으로 개발과정이 반복되므로 추가된 요구사항을 첨가할 수 있다.
  - ✓위험성 평가에 크게 의존하기 때문에 이를 발견하지 못하면 문제가 발생한다.



- 익스트림 프로그래밍(eXtreme Programming, XP) 모델
  - ✓1996년 켄트 백(Kent Back)과 워드 커닝험(Ward Cunningham)은 함께 다임러 크라이슬러 프로젝트를 진행하면서 새로운 소프트웨어 개발 방법을 정리하게 되는데, 그것이 XP 방법론이다. XP는 고객이 원하는 소프트웨어를 고객이 원하는 시기에 제공하는 것을 목표로 하며, 비즈니스 상의 요구가 시시각각 변동이 심한 경우에 적합한 개발방법이다. 12개의 구체적인 실천방법을 정의하고 있다.

문제

- 프로토타입 모형에 대한 설명으로 옳지 않은 것은?
  ① 프로토타입 모형은 발주자나 개발자 모두에게 공동의 참조 모델을 제공한다.
  ② 사용자의 요구사항을 충실히 반영할 수 있다.
  ③ 프로토타입 모형은 소프트웨어 생명주기에서 유지보수가 없어지고 개발단계 안에서 유지보수가 이루어지는 것을 볼 수 있다.
  ④ 최종결과물이 만들어 지는 소프트웨어 개발 완료시점에서 최초로 오류발견이 가능하다.

정답 4

정보처리기사

제4과목 소프트웨어공학
03 프로젝트 관리

문혜영교수

목차

1 프로젝트 관리

2 프로젝트 계획 수립

3 프로젝트 추산

4 비용산정기법

- 프로젝트 관리대상
  - ✓계획관리 : 비용산정, 일정계획
  - ✓품질관리
  - ✓위험관리
- 효과적인 프로젝트 관리를 위한 3P
  - ✓People
  - ✓Problem
  - ✓Process

문제

- 소프트웨어 프로젝트 관리에 중요한 영향을 주는 3대 요소는?
  ① 사람, 문제, 프로세스
  ② 문제, 프로젝트, 작업
  ③ 사람, 문제, 도구
  ④ 작업, 문제, 도구

정답 1

- 프로젝트 계획
  - ✓ 비용, 일정, 성능을 예측
- 소프트웨어개발 영역
  - ✓ 소프트웨어 개발 영역을 결정하는 주요요소는 기능, 성능, 제약조건, 인터페이스, 신뢰도 등이다.
- 프로젝트 추산
  - ✓ 프로젝트관리의후반까지 프로젝트 예측을 가능한 연기한다.(-> 현실성없다)
  - ✓ 유사 프로젝트를 참조한다.
  - ✓ 프로젝트를 작게 분리하여 예측한다.
  - ✓ 경험적 예측 모델을 활용한다.
- 프로젝트 계획 수립시 가장 먼저 고려해야 할 사항은 프로젝트 규모이다.

문제

- 소프트웨어 프로젝트 계획 수립시 소프트웨어 영역 결정의 주요요소로 거리가 먼것은?
  ① 기능
  ② 인적 자원
  ③ 인터페이스
  ④ 성능

정답 2

- 프로젝트 비용결정 요소
  - ✓프로젝트 요소 : 제품의 복잡도, 시스템이 크기, 요구되는 신뢰도
  - ✓자원 요소 : 인적자원, 하드웨어 자원, 소프트웨어 자원
  - ✓생산성 요소 : 개발자의 능력, 개발기간

문제

- 소프트웨어 프로젝트의 비용 결정 요소와 관련이 적은 것은?
  ① 개발자의 능력
  ② 요구되는 신뢰도
  ③ 하드웨어의 성능
  ④ 개발제품의 복잡도

정답 3

- 비용산정기법
  - ✓하향식-전문가 감정기법, 델파이 기법
  - ✓상향식-LOC기법, 개발단계별 인월수 기법,
    수학적 산정기법(COCOMO모형, Putnam모형, 기능점수(FP)모형

- LOC기법
  - ✓소프트웨어 각 기능의 원시 코드 라인수의 비관치, 낙관치, 기대치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 기법이다.
  - ✓예측치 $= \frac{a+4m+b}{6}$ (a:낙관치, b:비관치, c:기대치)
    - 낙관치 : 가장 적게 측정된 코드라인수
    - 비관치 : 가장 많이 측정된 코드라인수
    - 기대치 : 측정된 모든 코드라인수의 평균
  - ✓노력(인월)=개발기간 * 투입인원
  - ✓개발비용=노력(인월)*단위비용(1인당월평균 인건비)
  - ✓생산성=LOC/노력(인월)

정보처리기사

제4과목 소프트웨어공학
04 프로젝트 관리 2

문혜영교수

목차

1 수학적 산정기법

2 일정계획

3 PERT/CPM

4 WBS, 간트차트

- 비용산정기법
  - ✓하향식-전문가 감정기법, 델파이 기법
  - ✓상향식-LOC기법, 개발단계별 인월수 기법,
    수학적 산정기법(COCOMO모형, Putnam모형, 기능점수(FP)모형

- COCOMO모형
  - ✓보헴이 제안
  - ✓비용견적의 강도 분석 및 비용 견적의 유연성이 높아 소프트웨어 개발비 견적에 널리 통용되고 있다.
  - ✓같은 규모의 프로그램이라도 그 성격에 따라 비용이 다르게 책정된다.
  - ✓비용산정 결과는 프로젝트를 완성하는데 필요한 노력으로 나타난다.

- COCOMO의 소프트개발 유형
  - ✓조직형 : 중소규모, 5만라인 이하
  - ✓반분리형 : 30만라인 이하
  - ✓내장형 : 최대형 규모, 30만라인 이상
- COCOMO모형의 종류
  - ✓기본형
  - ✓중간형
  - ✓발전형

- Putnam모형

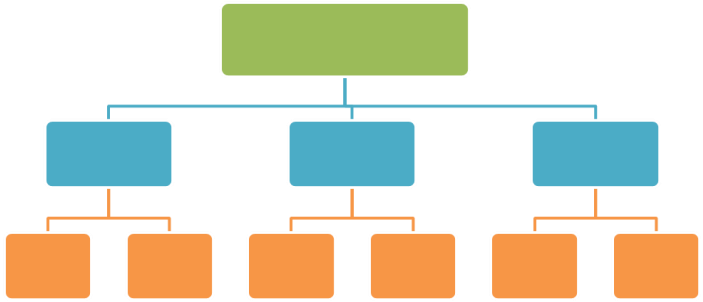
- 기능점수(FP)모형

- 비용산정의 자동화 도구
  - ✓SLIM : Rayleigh-Norden곡선과 Putnam예측모델을 기초로 개발됨
  - ✓ESTMACS : FP모형을 기초로 개발된

- 프로젝트 일정계획
  - ✓WBS
  - ✓PERT/CPM
  - ✓간트차트

- 브룩스 법칙
  - ✓프로젝트 진행중에 새로운 인력을 투입할 경우 일정을 더욱 지연시키고 프로젝트에 혼란을 가져오게 된다.

- WBS



- PERT/CPM
  - ✓프로젝트의 지연을 방지하고 계획대로 진행되게 하기위한 일정을 계획하는 것이다.
  - ✓대단위 계획의 조직적인 추진을 위해 자원의 제약하에 비용을 적게 사용하면서 초단시간 내 계획 완성을 위한 프로젝트 일정방법이다.

- PERT
  - ✓ 전체 작업의 상호관계를 표시하는 네트워크로 낙관적인 경우, 가능성이 있는 경우, 비관적인 경우로 나누어 각 단계별 종료시기를 결정하는 방법이다.
  - ✓결정경로, 작업에 대한 경계시간, 작업간의 상호 관련성을 알 수 있다.
- CPM
  - ✓프로젝트 완성에 필요한 작업을 나열하고 작업에 필요한 소요기간을 예측하는 데 사용되는 기법이다.
  - ✓전 작업이 완료된 후 다음 작업을 진행 할 수 있다.

- 간트차트

  ✓작업일정을 막대 도표를 이용하여 표시하는 것으로 시간선 차트라고도 한다.

  ✓이정표, 작업일정, 작업기간, 산출물로 구성된다.

  ✓수평 막대길이는 각 작업의 기간을 나타낸다.

정보처리기사

제4과목 소프트웨어공학
05 프로젝트 관리 3

문혜영교수

목차

1 프로젝트 조직구성

2 품질관리

3 품질표준

4 품질보증

- 프로젝트 조직 구성

✓분산형 팀 구성

- 팀원 모두가 의사결정에 참여 민주주의식 팀구성

✓중앙집중형 팀 구성

- 한 관리자가 의사결정, 책임 프로그래머 팀 구성

✓계층적 팀 구성

- 5~7명의초급 프로그래머를 작은 그룹으로 나누고 각 그룹을 고급 프로그래머가 관리한다.

문제

- 프로젝트를 추진하기 위하여 팀 구성원들의 특성을 분석해 보니 1명이 책임(고급) 프로그래머이고 몇 명의 중급 프로그래머가 포함되어 있었다. 이와 이와 같은 경우가 가장 적합한 팀 구성방식은?

  ① 책임 프로그래머 팀

  ② 민주주의식 팀

  ③ 계층형 팀

  ④ 구조적 팀

정답 1

- 소프트웨어 품질 표준(목표)
  - ✓ 소프트웨어 운영특성
    - 정확성
    - 신뢰성
    - 효율성
    - 무결성
    - 사용용이성
  - ✓ 소프트웨어 변경수용능력
    - 유지보수성
  - ✓ 소프트웨어 적응능력
    - 이식성
    - 재사용성
    - 상호운용성

문제

- 소프트웨어 품질 목표 중에서 옳고 일관된 결과를 얻기 위하여 요구되는 기능을 수행할 수 있는 정도를 나타낸 것은?

  ① 정확성

  ② 신뢰성

  ③ 효율성

  ④ 무결성

  정답 2

- 소프트웨어 품질보증
  - ✓ 개발단계에서는 정형기술검토를 통하여 품질목표의 충족여부를 점검한다.
  - ✓ 개발후에는 디버깅과 시험과정을 거친다.
- 정형기술검토
  - ✓ 검토회의, 검열등이 있다.
- 정형기술검토의 지침사항
  - ✓ 논쟁과 반박을 제한하라
  - ✓ 참가자의 수를 제한하라
  - ✓ 체크리스트를 개발하라
- 검토회의
  - ✓ 오류의 조기검출을 목적으로 하며 발견된 오류는 문서화한다.

정보처리기사

제4과목 소프트웨어공학
06 프로젝트 관리 4

문혜영교수

목차

1 신뢰성

2 위험관리

3 위험종류

4 형상관리

- 소프트웨어 신뢰성(가용성)

✓소프트에어 신뢰성은 측정과 예측이 가능하다.

✓한 프로그램이 주어진 시점에서 요구사항에 따라 운영되는 확률을 의미한다.

✓가용성=$\frac{MTBF}{MTBF+MTTR} * 100\% = \frac{MTTF}{MTTF+MTTR} * 100\%$

- 위험관리
  - ✓ 위험 관리는 프로젝트 추진과정에서 예상되는 각종 돌발상황을 미리 예상하고 이에 적절한 대책을 수립하는 일련의 활동이다.
  - ✓ 위험식별, 위험 분석 및 평가, 위험 관리 계획, 위험 감시 및 조치 순서이다.
- 위험의 종류
  - ✓ 인력부족, 예산관리, 일정 관리, 사용자 요구사항 변경
  - ✓ 가장 대표적인 위험 요소는 사용자 요구사항 변경이다.
- 위험 분석 및 평가
  - ✓ 내재한 위험 요소를 인식하고 그 영향을 분석하는 활동이다.
- 위험 감시(Risk Monitoring)
  - ✓ 위험 요소 징후들에 대하여 계속적으로 인지하는 것

문제

- 프로젝트 추진 과정에서 예상되는 각종 돌발상황을 미리 예상하고 이에 대한 적절한 대책을 수립하는 일련의 활동을 무엇이라고 하는가?

① 위험관리

② 일정관리

③ 코드 관리

④ 모형관리

정답 1

- 소프트웨어 형상관리
  - ✓소프트웨어 개발과정에서 소프트웨어의 변경사항을 관리하기 위해 개발된 일련의 활동이다.
  - ✓전 단계에 적용되는 활동이며 유지보수 단계에서도 수행된다.
- 형상관리 기능
  - ✓형상식별
  - ✓버전제어
  - ✓변경제어

문제

- 소프트웨어 형상관리란?
  ① 소프트웨어 구성항목을 관리하는 것
  ② 개발과정의 변화되는 사항을 관리하는 것
  ③ 테스트 과정에서 소프트웨어를 통합하는 것
  ④ 개발 인력을 관리하는 것

정답 2

정보처리기사

제4과목 소프트웨어공학
07 SW개발 방법

문혜영교수

목차

1 요구사항 분석

2 구조적 분석기법

3 자료흐름도

4 자료사전

- 요구사항 분석
  - ✓소프트웨어 개발의 첫 단계로 사용자의 요구사항을 이해하고 문서화 하는 활동이다.
- 요구사항 분석작업
  - ✓문제인식-사용자와의 면담, 설문조사, 각종 문서 검토

- 요구사항 분석의 어려움
  - ✓ 사용자와 개발자의 지식배경의 다양화, 용어 불일치로 의사소통 어려움
  - ✓ 시스템규모와 대상이 광범위하고 소프트웨어의 복잡화
  - ✓ 사용자 생각의 부정확성
- 요구사항 분석가의 자질
  - ✓ 세부적인 요소를 관찰할 수 있는 능력
- 구조적 분석기법
  - ✓ 자료흐름도, 자료사전, 소단위명세서, 개체관계도, 상태전이도, 제어 명세서등의 도구를 이용한다.

문제

- 사용자의 요구사항 분석 작업이 어려운 이유와 거리가 먼것은?
  ① 개발자와 사용자간의 지식이나 표현의 차이가 커서 상호 이해가 쉽지 않다.
  ② 사용자의 요구는 예외가 거의 없어 열거와 구조화하기 어렵지 않다.
  ③ 사용자의 요구사항이 모호하고 부정확하며, 불완전하다.
  ④ 개발하고자 하는 시스템 자체가 복잡하다.

정답 2

- 자료 흐름도

✓자료의흐름과 처리를 중심으로 하는 구조적 분석기법이다.

✓자료는 처리를 거쳐 변환될때마다 새로운 이름이 부여되고 처리는 입력 자료가 발생하면 기능을 수행한 후 출력 자료를 산출한다.

기 호	표기법	의미
◯	프로세스(process)	자료를 변환시키는 과정을 나타낸다.
═══	자료저장소(data store)	시스템상의 자료저장소로서 파일 또는 데이터베이스 테이블
→	자료흐름(data flow)	외부객체나 프로세서 사이에 전달되는 데이터흐름.자료의 흐름을 나타낸다.
▭	단말(terminator)	개발하고자 하는 소프트웨어 또는 시스템으로 외부에서 데이터를 주고받는 주체. 자료의 출처와 도착지를 나타낸다.

- 자료사전(DD, data dictionary)

✓자료흐름도에 나타난 자료를 더 자세히 기록한 것이다.

✓자료 흐름도에 시각적으로 표시된 자료에 대한 정보를 체계적이고 조직적으로 모아 개발자나 사용자가 편리하게 사용할 수 있다

기 호	의 미
=	자료의 정의
+	자료의 구성, 연결, 그리고(and)
{ }	자료의 반복, $\{\ \}_k$ 최소 k번 반복, $\{\ \}^k$ 최대k번 반복
[|]	자료의 선택, 또는(or)
( )	생략 가능한 자료
**	자료의 주석(설명)

문제

- 자료흐름도의 구성요소와 표시 기호의 연결이 옳지 않은 것은?

① 종착지 : 오각형

② 자료흐름 : 화살표

③ 처리공정 : 원

④ 자료저장소 : 직선(평행선)

정답 1

정보처리기사

제4과목 소프트웨어공학
08 SW개발 방법2

문혜영교수

목차

1 소단위명세서

2 개체 관계도

3 상태전이도

4 HIPO

- 소단위 명세서(mini specification)
  - ✓세분화된 자료 흐름도에서 최하위 단계의 처리 절차를 기술한 것으로 프로세스 명세서라고도 한다.
  - ✓소단위 명세서는 분석가의 문서이며, 자료 흐름도(DFD)를 지원하기 위하여 작성한다.
  - ✓작성 툴에는 서술문장, 구조적언어, 의사결정나무, 의사결정표, 그래프 등이 있다.

- 개체관계도(ERD)

✓ 개체(entity)와 속성(attribute), 이들의 관계를 이용하여 현실세계를 개념적으로 표현한다.

✓ 개체란 사람이나 사물과 같이 구별되는 모든 것, 소프트웨어에 의해 인식되는 여러종류의 자료

예) 서점에 필요한 개체 : 고객, 책

예) 학교에 필요한 개체 : 학과, 과목

✓ 속성(attribute)이란 개체나 관계가 가지고 있는 고유의특성이다.

기 호	모양	의 미
	사각형	개체
	마름모, 다이아몬드	관계
	타원	속성
	타원밑줄	기본키 속성
	선, 링크	개체와 속성을 연결

- 상태전이도(STD)
  - ✓사각형은 시스템의 상태를 나타내고 화살표는 상태 전이를 나타낸다.

- 요구사항 분석을 위한 CASE(자동화 도구)
  - ✓SADT, SREM, PLS/PSA, TAGS, EPOS

- HIPO
  - ✓ 시스템의 분석 및 설계나 문서화 할때 사용되는 기법으로 시스템 실행과정인 입력, 처리, 출력의 기능을 나타낸다.
    - 기본 시스템 모델은 입력,처리, 출력으로 구성된다.
    - 하향식 소프트웨어 개발을 위한 문서화 도구이다.
    - 보기 쉽고 이해하기 쉽다.
    - 여러 개의 고유모듈들로 분할하고 이들간의 인터페이스를 계층적 구조로 표현한것을 HIPO Chart라고 한다.
  - ✓ HIPO Chart의 종류
    - 가시적 도표
    - 총체적 도표
    - 세부적 도표

문제

- 다음 내용을 자료사전의 형태로 옳게 표기한 것은?
  " 고객명세는 고객성명, 고객번호, 고객주소로 구성되어 있으며, 고객성명과 고객번호는 둘 중 하나만 선택이 가능함"

  ① 고객명세:<고객성명|고객번호>+고객주소

  ② 고객명세={고객명세|고객번호}+고객주소

  ③ 고객명세=[고객성명|고객번호]+고객주소

  ④ 고객명세:(고객성명|고객번호)+고객주소

정답 3

문제

- 다음 설명에 해당하는 것은?
  "세분화된 자료 흐름도에서 최하위 단계 프로세스의 처리 절차를 기술한 것으로 작성 툴에는 서술문장, 구조적언어, 의사결정나무, 의사결정표, 그래프 등이 있다."

  ① ERD

  ② Mini-spec

  ③ DD

  ④ STD

  정답 2

정보처리기사

제4과목 소프트웨어공학
09 SW개발 방법3

문혜영교수

목차

1 설계

2 바람직한 설계

3 모듈화

4 결합도

설계

- 설계모형
  - ✓데이터 설계
  - ✓구조 설계(architectural design)
  - ✓인터페이스 설계
  - ✓절차 설계(프로시저)

- 설계 기본 원리
  - ✓ 모듈화
  - ✓ 추상화
    - 기능추상화
    - 제어추상화
    - 자료추상화
  - ✓ 단계적 정제
  - ✓ 정보은닉
  - ✓ 프로그램 구조
    - 공유도 fan-in
    - 제어도 fan-out

- 바람직한 설계의 특징
  - ✓독립적인 기능적 특성을 가진 모듈로 구성되어야 한다.
  - ✓논리적 요소들로 분리되는 구조를 가져야 한다.
  - ✓효과적인 제어를 위해 계층적 자료조직이 제시되어야 한다.
  - ✓자료와 프로시저에 대한 분명하고 분리된 표현을 포함해야 한다.
  - ✓모듈간와 외부개체간의 연결 복잡성을 줄이는 인터페이스를 가져야 한다.
  - ✓요구사항을 모두 구현해야 하고 유지보수가 용이해야 한다.
  - ✓모듈의 크기를 유지하고 결합도는 낮추고 응집도는 높인다.
  - ✓전체적이고 포괄적인 개념을 설계한 후 차례대로 세분화하여 구체화 시켜 나간다.

문제

- 바람직한 모듈의 설계 지침이 아닌 것은?
  ① 유지보수가 용이해야 한다.
  ② 가능한 모듈을 독립적으로 생성하고 결합도를 최대화 한다.
  ③ 복잡도와 중복성을 줄이고 일관성을 유지시킨다.
  ④ 모듈의 기능은 지나치게 제한적이어서는 안된다.

정답 2

- 모듈의 기능적 독립성
  - ✓모듈이 하나의 기능만을 수행하고 다른 모듈과의 과도한 상호작용을 배제함으로써 이루어진다.
  - ✓모듈이란 소프트웨어를 각 기능별로 분할하는 것으로, 소프트웨어 구조를 이루는 기본단위이다.
  - ✓모듈화는 소프트웨어를 독립적인 몇 개의 작은 부분으로 분할하여 복잡한 문제를 작고, 간단하게 나누어 해결하는 방법이다.
  - ✓모듈화는 독립성이 높아야 하며, 모듈의 독립성은 결합도와 응집도에 의해 측정된다

- 결합도

✓한 모듈과 다른 모듈간의 상호의존도를 측정하는 것으로 독립적인 모듈이 되기 위해서 는 결합도가 약해야 한다

결합도 강함	내용 결합도(Content coupling)
	공통 결합도(Common coupling)
	외부 결합도(External coupling)
↕	제어 결합도(Control coupling)
	스탬프 결합도(Stamp coupling)
결합도 약함	데이터 결합도(Data coupling)

정보처리기사

제4과목 소프트웨어공학
10 SW개발 방법4

문혜영교수

목차

4 구현

3 절차설계

2 구조설계

1 응집도

- 응집도

✓한 모듈 내에 있는 구성 요소들이 서로 관련되어 있는 정도를 의미하며, 관련성을 측정하는 것으로 응집도가 높도록 설계되도록 해야 한다

응집도 강함	기능적 응집도(functional cohesion)
	순차적 응집도(sequential cohesion)
	통신적(교환적) 응집도(communicational cohesion)
↕	절차적 응집도(procedural cohesion)
	시간적 응집도(temporal cohesion)
	논리적 응집도(logical cohesion)
응집도 약함	우연적 응집도(coincidental cohesion)

- 효과적인 모듈화 설계방안
  - ✓ 결합도는 줄이고 응집도는 높여서 모듈의 독립성을 높임
  - ✓ 모듈의 기능은 예측이 가능해야 하며 지나치게 제한적이어서는 안 됨
  - ✓ 유지보수가 용이해야 함
  - ✓ 모듈 크기는 시스템의 전반적인 기능과 구조를 이해하기 쉬운 크기로 분해
  - ✓ 하나의 입구와 하나의 출구를 갖도록 해야 함

문제

- 한 모듈이 다른 모듈의 내부 기능 및 그 내부 자료를 참조하는 경우 이를 무슨 결합이라고 하는가?
  ① 내용 결함
  ② 제어 결함
  ③ 공통 결함
  ④ 스탬프 결함

정답 1

설계 방법

- 구조적 설계 절차(자료흐름 중심설계)

  ① 정보 흐름의 유형 설정

  ② 흐름의 경계 표시

  ③ 자료 흐름도를 프로그램 구조로 사상

  ④ 제어 계층을 분해(Factoring)시켜서 정의

  ⑤ 경험적 방법으로 구체화

- 인터페이스 설계

✓인터페이스 설계는 소프트웨어와 상호작용하는 시스템과 사용자등과 어떻게 통신하는지를 기술하는 과정이다.

✓사용자 인터페이스 설계시 오류 메시지나 경고에 관한 지침

- 메시지 내용은 이해하기 쉬어야 함
- 오류 회복을 위한 구체적인 설명이 제공되어야 함
- 소리나 색 등을 이용하여 듣거나 보기 쉽도록 의미를 전달해야 함
- 오류로 인해 발생될 수 있는 부정적인 내용은 절대 사용 금지

- 절차(프로시저)설계

✓흐름도 : 박스는 처리, 다이아몬드는 조건, 화살표는 제어흐름을 나타낸다.

  - 순차
  - 선택
  - 반복

✓N-S차트 : 논리의 기술에 중점 둔 도형을 이용한 표현방법이다.

  - 순차
  - 반복
  - 선택

✓프로그램 설계언어(PDL)

문제

- N-S차트에 대한 설명으로 거리가 먼것은?

① 논리의 기술에 중점을 둔 도형식 표현 방식이다.

② 연속, 선택 및 다중선택, 반복등의 제어 논리구조로 표현한다.

③ 주로 화살표를 사용하여 논리적인 제어구조로 흐름을 표현한다.

④ 조건이 복합되어 있는 곳의 처리를 시각적ㅇ로 명확히 식별하는데 적합하다.

정답 3

구현

- 구현 단계
  - 컴퓨터가 알수 있는 모습으로 변환하는 과정으로 프로그래밍 또는 코딩 단계
  - 디버깅 및 변경, 검사가 용이하도록 해야 한다.
- 프로그래밍 언어의 선정기준
  - 과거의 개발 실적
  - 대상 업무의 성격
  - 개발 담당자의 경험과 지식

- 구조적 프로그래밍
  - Dijkstra에 의해 제안된 것
  - 기본 제어구조는 순차, 선택, 반복이다.
  - 프로그래밍의 규칙

✓ 프로그램의 제어 흐름 선형화

✓ 단일 입구와 단일 출구만 가지게 함

✓ GOTO문을 사용하지 않음

✓ 순차, 선택, 반복의 세 가지 기본 제어 구조 사용

정보처리기사

제4과목 소프트웨어공학
11 SW개발 방법5

문혜영교수

목차

1 검사

2 검사기법

3 화이트박스 테스트

4 블랙박스 테스트

검사

- 검사(Test)
  - ✓노출되지 않은 숨어있는 결함을 찾기 위해 소프트웨어를 작동시키는 일련의 행위와 절차로 오류 발견을 목적으로 프로그램을 실행하여 품질을 평가하는 과정이다.
- 소프트웨어 테스트의 목적
  - ✓ 프로그램에 잠재된 오류의 발견
  - ✓ 기술적인 기능 및 성능의 확인
  - ✓ 사용자 요구만족도 향상
  - ✓ 제품신뢰도 향상

- 검사기법
  - ✓화이트박스 테스트
    - 기초경로검사, 제어 흐름 테스트, 분기(branch) 테스트, 제어구조검사(조건검사, 루프검사, 데이터 흐름검사) 등이 있다
  - ✓블랙박스 테스트
    - 동치분할검사, 경계값 분석(Boundary Value Analysis), 원인-효과그래프(Cause Effect Graphing Testing), 오류 예측 검사, 비교검사(Comparison Testing) 등이 있다.

- 화이트박스 테스트(White-Box Test)
  - ✓ 논리위주의 검사방식으로 모듈안의 작동을 자세히 관찰할 수 있다.
  - ✓ 소프트웨어 내부 소스 코드를 테스트하는 기법이다.
  - ✓ 프로그램 내부의 논리적 구조 및 복잡도 등에 대한 TEST이다.
  - ✓ 모듈 안의 논리적 구조를 검사하는 것으로 원시코드의 모든 문장을 한번 이상 수행함으로써 수행된다.
  - ✓ 모듈 안의 작동을 직접 관찰
  - ✓ 원시 코드(모듈)의 모든 문장을 한 번 이상 수행함으로써 수행됨
  - ✓ 프로그램의 논리적 경로를 제어
  - ✓ 각 조건에서의 참과 거짓의 모든 논리적 결정이 적어도 한 번 이상 실행됨
  - ✓ 논리 흐름도, 루프 구조, 순환 복잡도에 관한 오류 찾기 가능

- 화이트박스 테스트

✓기초경로검사

- 절차
  - ① 설계나 원시 코드를 기초로 흐름도 작성
  - ② 흐름도의 논리적 복잡도 측정
  - ③ 독립 경로들의 기초 집합 결정
  - ④ 검사 사례 선정
- 제어 흐름도
- 순환 복잡도
  - V=E-N+2 (E화살표수, N노드수)

- 화이트박스 테스트
  - ✓제어구조검사
    - 조건검사
    - 루프검사
    - 데이터흐름검사

- 블랙박스 테스트(Black Box Test)
  - ✓ 블랙박스 검사는 어떤 소프트웨어를 내부 구조나 작동 원리를 모르는 상태에서 소프트웨어의 동작을 검사하는 방법이다.
  - ✓ '이 소프트웨어는 무슨 역할을 수행해야 되는가'와 같이 대상이 되는 소프트웨어의 특징이나 요구 사항 등에 초점을 맞춰 검사가 이루어진다.
  - ✓ 설계된 모든 기능들이 정상적으로 작동되는지 검사하는 방법으로 기능검사라고도 한다.
  - ✓ 설계된 모든 기능들이 정상적으로 작동되는지 검사하는 방법으로 기능검사라고도 한다.
  - ✓ 부정확하거나 누락된 기능, 인터페이스 오류, 자료구조나 성능오류등을 발견하기 위해 사용된다.

- 블랙박스 테스트
  - ✓동치분할검사,
  - ✓경계값 분석(Boundary Value Analysis),
  - ✓원인-효과그래프(Cause Effect Graphing Testing),
  - ✓오류 예측 검사,
  - ✓비교검사(Comparison Testing)

문제

- 블랙박스 테스팅을 통해 발견하기 힘든 오류는?
  ① 성능 오류
  ② 부정확한 오류
  ③ 인터페이스 오류
  ④ 논리구조상의 오류

정답 4

문제

- 화이트 박스 검사로 찾기 힘든 오류는?
  ① 논리 흐름도
  ② 루프 구조
  ③ 자료 구조
  ④ 순환 복잡도

정답 4

정보처리기사

제4과목 소프트웨어공학
12 SW개발 방법6

문혜영교수

목차

1 검사전략

2 통합검사

3 유지보수활동

4 유지보수 부작용

검사(Test)

- 검사 전략
  - ✓단위(코드)검사
  - ✓통합(설계)검사
  - ✓검증(요구사항)검사
  - ✓시스템 건사

- 하향식 통합검사

✓깊이우선통합법이나 넓이 우선 통합법을 사용할 수 있다.

✓절차

- 주요 제어 모듈을 드라이버로 사용, 종속 모듈들은 스터브(Stub)로 대체
- 통합 방식에 따라 종속 스터브들이 실제 모듈로 교체
- 모듈이 통합될 때마다 검사 실시
- 회귀 검사 실시

- 상향식 통합검사

✓하위모듈에서 상위모듈방향으로 통합하면서 검사

✓절차

(1) 하위 모듈들을 클러스터(Cluster)로 결합
(2) 검사 사례 입·출력을 조정하기 위해 드라이버(Driver) 작성
(3) 클러스터를 검사
(4) 드라이버를 제거하고, 클러스터는 프로그램 구조의 상위로 이동하여 결합

- 검증검사
  - ✓소프트웨어가 사용자의 요구사항을 충족시키는가에 중점을 두고 검사하는 방법
  - ✓알파 검사 : 개발자의 장소에서 사용자가 개발자 앞에서 행하는 검사 기법
  - ✓베타 검사 : 선정된 최종 사용자가 여러 명의 사용자 앞에서 행하는 검사 기법
- 시스템 검사
  - ✓개발된 소프트웨어가 해당 컴퓨터에서 완벽하게 수행하는가 검사
  - ✓복구검사, 보안검사, 강도검사, 성능검사

유지보수

- 유지보수
  - ✓개발단계중 가장 많은 노력과 비용이 투입되는 단계된다.
  - ✓사용자에게 인수되어 설치된 후 발생하는 모든 공학적 작업이다.
  - ✓유지보수를 용이하게 하려면 시험 용이성, 이해성, 수정용이성, 이식성등이 고려되어야 한다.

유지보수

- 유지보수 종류
  - ✓수정(Corrective) 보수
  - ✓적응(Adaptive) 보수
  - ✓완전화(Perfective) 보수
  - ✓예방(Preventive) 보수
- 유지보수의 부작용
  - ✓코딩 부작용
  - ✓자료 부작용
  - ✓문서화 부작용

문제

- 소프트웨어 유지보수 유형중 현재 수행중인 기능의 수정, 새로운 기능의 추가, 전반적인 기능 개선등의 요구를 사용자로부터 받았을때 수행되는 유형으로서 유지보수 유형중 제일 많은 비용이 소요되는 것은?

  ① 예방보수

  ② 적응적보수

  ③ 수정보수

  ④ 완전화보수

정답 4

정보처리기사

제4과목 소프트웨어공학
13 객체지향 SW공학

문혜영교수

목차

1 객체

2 객체지향 기법

3 객체지향분석

4 럼바우 분석

객체지향 소프트웨어

- 객체지향 기법
  - ✓현실세계의 개체를 하나의 객체로 만들어 소프트웨어를 개발할때에도 객체를 조립해서 작성할 수 있도록 하는 기법
- 객체
  - ✓데이터
  - ✓함수
- 클래스
  - ✓인스턴스
- 메시지

- 객체지향 기법
  - ✓캡슐화
  - ✓정보은닉
  - ✓추상화
  - ✓상속성
  - ✓다형성

문제

- 객체지향 개념에서 연관된 데이터와 함수를 함께 묶어 외부와 경계를 만들고 필요한 인터페이스만을 밖으로 드러내는 과정을 무엇이라 하는가?

  ① 정보은닉

  ② 클래스

  ③ 캡슐화

  ④ 통합

  정답 3

- 객체지향 분석

- 객체지향 분석 방법론
  - ✓Rumbaugh
  - ✓Booch(부치)
  - ✓Jacobson
  - ✓Coad와 Yourdon
  - ✓Wirfs-Brock

- 럼바우 분석기법
  - ✓객체 모델링(Object Modeling)
  - ✓동적 모델링(Dynamic Modeling)
  - ✓기능 모델링(Functional Modeling)

정보처리기사

제4과목 소프트웨어공학
14 객체지향 SW공학 2

문혜영교수

목차

1 객체지향 설계

2 객체지향 구현

3 객체지향 테스트

4 UML

- 객체지향 설계(OOD)
  - ✓최근의 소프트웨어 제품의 전형적인 타입인 사용자 중심, 대화식 프로그램의 개발에 적합하다.
  - ✓설계의 개념 : 추상화, 정보 은닉, 기능 독립성, 모듈화, 상속성을 바탕으로 하고 이중 가장 중요한 개념은 모듈화이다.
  - ✓일반적 순서 : 문제 정의 → 요구 명세화 → 객체 연산자 정의 →
    → 객체 인터페이스 결정 → 객체 구현

- 객체지향 구현

✓코딩하는 단계이다.

✓객체지향 프로그래밍(OOP)

- 객체라는 단위를 중심으로 프로그램을 개발하는 기법이다.
- 객체지향성 언어 : Smalltalk, C++, Objective C

- 객체지향 테스트
  - ✓클래스 테스트
  - ✓통합 테스트
    - 스레드 기반 테스트
    - 사용기반 테스트
  - ✓확인 테스트
  - ✓시스템 테스트

- UML
  - ✓객체지향 분석설계를 위한 모델링 언어로서 OMG에서 UML을 표준으로 지정하였다.
  - ✓개발자와 고객, 개발자 상호간의 의사소통을 원활하게 한다.
  - ✓도형과 기호를 사용하는 9개의 다이어그램 중심의 표기법이다.

- UML
  - ✓Use-case Diagram (사용자 사례)
  - ✓Class Diagram
  - ✓Sequence Diagram (순서)
  - ✓State Diagram(상태)
  - ✓Activity Diagram(활동)
  - ✓Object Diagram
  - ✓Communication Diagram
  - ✓Component Diagram
  - ✓Deployment Diagram

문제

- 럼바우의 객체 모델링에서 상태도와 자료흐름도는 각각 어느 모델링과 관련이 있는가?
  ① 상태도-기능 모델링, 자료 흐름도-동적 모델링
  ② 상태도-객체 모델링, 자료 흐름도-기능 모델링
  ③ 상태도-객체 모델링, 자료 흐름도-동적 모델링
  ④ 상태도-동적 모델링, 자료 흐름도-기능 모델링

정답 4

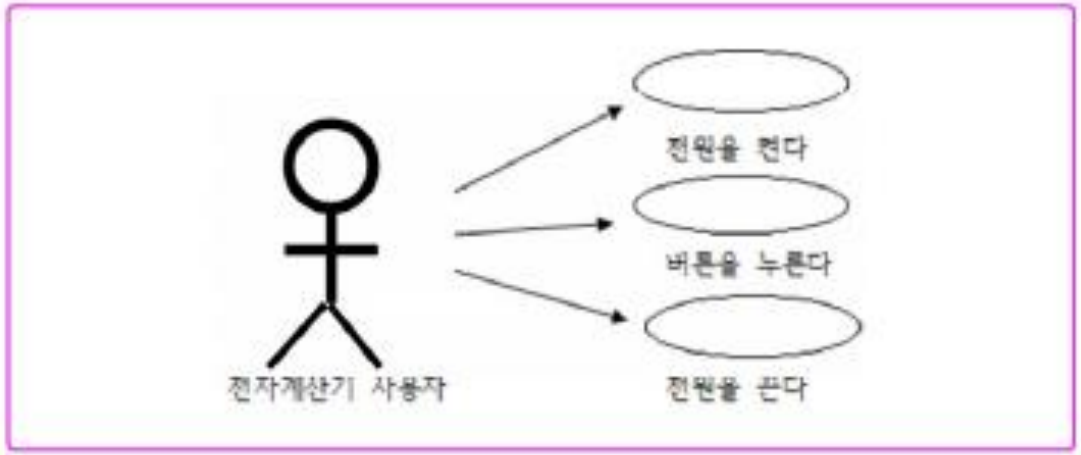
문제

- 객체지향 테스팅 전략 중에서 단위 테스팅에 사용되는 것은 어느것인가?
  ① 클래스 테스트
  ② 클러스터 테스트
  ③ 사용 기반 테스트
  ④ 스레드 기반 테스트

정답 1

문제

<그림>은 전자계산기(Calculator)를 객체 지향적으로 분석한 다이어그램이다. 어떤 다이어그램인가?



① Usecase Diagram
② Sequence Diagram
③ State Diagram
④ Class Diagram

정답 1

정보처리기사

제4과목 소프트웨어공학
15 SW공학의 발전적 추세

문혜영교수

목차

4 CASE

3 클라이언트/서버

2 재공학

1 재사용

- 소프트웨어 재사용(Re-use)
  - ✓이미 개발된 소프트웨어의 전체 또는 일부분을 새로운 소프트웨어 개발이나 유지에 사용하는 것이다.
  - ✓소프트웨어 재사용에 가장 많이 이용되는 것은 프로그램 소스코드이다
  - ✓이점
    - 개발 시간과 비용을 단축시킴
    - 소프트웨어 품질을 향상시킴
    - 소프트웨어 개발의 생산성을 향상시킴
    - 프로젝트 실패의 위험을 감소시킴

- 소프트웨어 재공학(Re-Engineering)
  - ✓기존시스템을 새로운 요구에 맞도록 소프트웨어 성능을 향상시켜서 보다 나은 시스템을 구축하는 것이다.
  - ✓재공학은 유지 보수성, 생산성, 품질의 향상을 목적으로 한다
  - ✓예방 유지보수 측면에서 소프트웨어 위기를 해결하는 방법이라고 할 수있다.
  - ✓소프트웨어 수명이 연장된다.

- 재공학의 방법에는 분석, 재구성, 역공학, 이식 등이 있다
  - ✓분석(Analysis) : 기존 소프트웨어의 명세서를 확인하고 소프트웨어의 동작을 이해하고 재공학 대상을 선정하는 활동이다.
  - ✓재구성(Restructuring) : 소프트웨어 기능을 변경하지 않고 소프트웨어 형태에 맞게 수정하는 활동으로, 상대적으로 추상적 수준에서 하나의 표현을 다른 형태로 변경한다.
  - ✓역공학(Reverse Engineering) : 기존 소프트웨어를 분석하여 소프트웨어 개발과 데이터처리 과정을 분석하고 설계 정보를 재발견하거나 다시 만드는 작업이다.
  - ✓이식(Migration) : 기존 소프트웨어를 다른 운영체제나 하드웨어 환경에서 사용할 수 있게 변환하는 작업이다

- 클라이언트/서버시스템의 소프트웨어 요소
  - ✓애플리케이션요소
  - ✓데이터베이스 요소
  - ✓프레젠테이션/상호작용 요소
- 미들웨어

- CASE(Computer Aided Software Engineering)
  - ✓ 컴퓨터와 전용 소프트웨어 도구를 이용하여 요구분석, 설계, 구현, 검사, 디버깅 등의 과정을 자동화는 작업이다.
  - ✓ CASE는 컴퓨터 회사들이 하드웨어 제작과정으로부터 아이디어를 빌려오기 시작한 1970년대에 생기기 시작되었으며, 그것을 소프트웨어 개발에 적용하였다.
  - ✓ CASE의 주요기능 : 소프트웨어 생명주기 전단계의 연결, 다양한 소프트웨어 개발모형지원, 그래픽 지원
  - ✓ CASE의 특징

    ① CASE 가격은 고가이지만 개발비용은 전체적으로 절감할 수 있다.

    ② CASE는 스스로 동작하는 것이 아니므로 분석가의 지원이 필요하다.

    ③ 수정이 용이하고 정확하다.

    ④ 유지보수가 간편하다.

    ⑤ 재사용성이 높다
  - ✓ CASE분류
    - 상위 CASE
    - 하위 CASE
    - 통합 CASE

- 정보저장소
  - ✓소프트웨어를 개발하는 과정동안 모여진 정보를 보관하여 관리하는 곳이다.
  - ✓오늘날에는 데이터베이스가 정보저장소 역할을 담당한다.
- 정보저장소 사용의 이점
  - ✓도구들과 생명주기활동, 사용자들, 응용소프트웨어들 사이의 통신과 소프트웨어 시스템의 정보 공유향상
  - ✓시스템 정보가 정보저장소에 의해 관리되므로 유지 보수성향상

CASE에 대한 설명으로 거리가 먼 것은?

① 개발도구와 개발 발명품이 결합된 것이다.

② 시스템 개발과정의 일부 또는 전체를 자동화 하는 것이다.

③ 유지보수성을 높이기 위해 기존 소프트웨어를 재구성하고 새로운 기술을 적용 시키는 것이다.

④ 정형화된 구조 및 매커니즘을 소프트웨어 개발에 적용하여 소프트웨어 생산성 향상을 구현하는 공학기법이다.

정답 3

정보처리기사

제4과목 소프트웨어공학
16 모의고사1

문혜영교수

61. 객체에게 어떤 행위를 하도록 지시하는 명령은?

① Class ② Instance

③ Object ④ Message

62. 소프트웨어 품질 목표 중 사용자의 요구 기능을 충족시키는 정도를 의미하는 것은?

- ① Correctness ② Integrity
- ③ Flexibility ④ Portability

63. 다음 중 가장 결합도가 강한 것은?

① data coupling ② stamp coupling

③ common coupling ④ control coupling

64. 럼바우 분석 기법에서 정보 모델링이라고도 하며, 시스템에서 요구되는 객체를 찾아내어 속성과 연산 식별 및 객체들 간의 관계를 규정하여 객체 다이어그램으로 표시하는 모델링은?

① 동적 모델링 ② 객체 모델링

③ 기능 모델링 ④ 정적 모델링

65. 소프트웨어 개발의 생산성에 영향을 미치는 요소로 가장 거리가 먼 것은?

① 프로그래머의 능력

② 팀 의사 전달

③ 제품의 복잡도

④ 소프트웨어 사용자의 능력

66. 다음 중 상위 CASE 도구가 지원하는 중요 기능으로 볼 수 없는 것은?

- ① 모델들 사이의 모순 검사 기능
- ② 모델의 오류 검증 기능
- ③ 원시 코드 생성 기능
- ④ 자료흐름도 작성 기능

67. 프로토타이핑 모형(Prototyping Model)에 대한 설명으로 옳지 않은 것은?

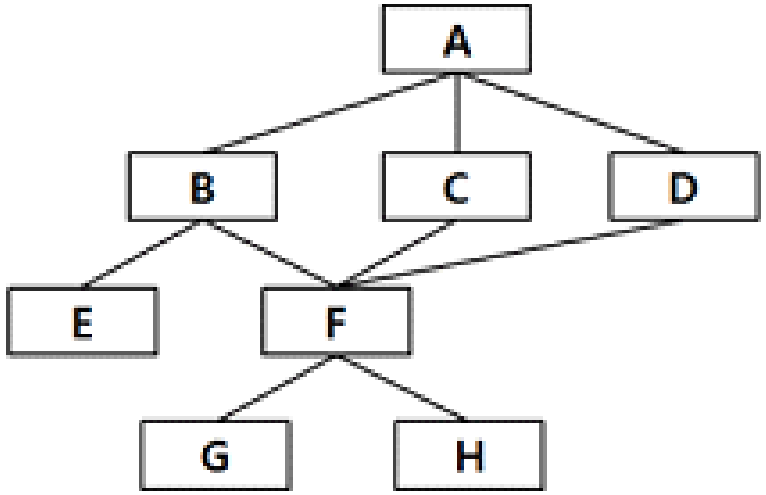
① 개발단계에서 오류 수정이 불가하므로 유지보수 비용이 많이 발생한다.

② 최종 결과물이 만들어지기 전에 의뢰자가 최종 결과물의 일부 또는 모형을 볼 수 있다.

③ 프로토타입은 발주자나 개발자 모두에게 공동의 참조 모델을 제공한다.

④ 프로토타입은 구현단계의 구현 골격이 될 수 있다.

68. 다음은 어떤 프로그램 구조를 나타낸다. 모듈 F에서의 fan-in과 fan-out의 수는 얼마인가?



① fan-in: 2 fan-out:3

② fan-in: 3 fan-out:2

③ fan-in: 1 fan-out:2

④ fan-in: 2 fan-out:1

69. 모듈(module)의 응집도(cohesion)가 약한 것부터 강한 순서로 옳게 나열된 것은?

① 기능적응집 → 시간적응집 → 논리적응집

② 시간적응집 → 기능적응집 → 논리적응집

③ 논리적응집 → 시간적응집 → 기능적응집

④ 논리적응집 → 기능적응집 → 시간적응집

70. 소프트웨어 프로젝트(Project)의 특징에 대한 설명으로 가장 거리가 먼 것은?

① 모든 소프트웨어 프로젝트는 항상 시작과 끝이 있다.

② 모든 소프트웨어 프로젝트는 서로 다르다.

③ 모든 소프트웨어 프로젝트는 대단위 사업을 의미한다.

④ 모든 소프트웨어 프로젝트는 개략적인 범위 정의에서부터 시작하여 점차 구체화하여 구현해 간다.

71. 소프트웨어 개발 모델 중 나선형 모델의 네 가지 주요활동이 순서대로 나열된 것은?

Ⓐ 계획 수립
Ⓑ 고객 평가
Ⓒ 개발 및 검증
Ⓓ 위험 분석

① Ⓐ-Ⓑ-Ⓓ-Ⓒ 순으로 반복
② Ⓐ-Ⓓ-Ⓒ-Ⓑ 순으로 반복
③ Ⓐ-Ⓑ-Ⓒ-Ⓓ 순으로 반복
④ Ⓐ-Ⓒ-Ⓑ-Ⓓ 순으로 반복

72. 블랙박스 검사 기법에 해당하는 것으로만 짝지어진 것은?

가. 데이터 흐름 검사 나. 루프 검사 다. 동치 분할 검사 라. 경계값 분석 마. 원인-호과 그래프 기법 바. 비교 검사

① 가, 나　　② 가, 라, 마, 바

③ 나, 라, 마, 바　　④ 다, 라, 마, 바

73. 소프트웨어 공학의 전통적인 개발 방법인 선형 순차 모형의 순서를 옳게 나열한 것은?

① 구현 → 분석 → 설계 → 테스트 → 유지보수

② 유지보수 → 테스트 → 분석 → 설계 → 구현

③ 분석 → 설계 → 구현 → 테스트 → 유지보수

④ 테스트 → 설계 → 유지보수 → 구현 → 분석

74. 객체에 대한 특성을 설명한 것으로 가장 옳지 않은 것은?

① 객체마다 각각의 상태를 갖고 있다.

② 식별성을 가진다.

③ 행위에 대하여 그 특징을 나타낼 수 있다.

④ 일정한 기억장소를 가지고 있지 않다.

75. 소프트웨어 품질보증을 위한 FTR의 지침사항으로 가장 옳지 않은 것은?

① 논쟁과 반박의 제한성
② 의제의 무제한성
③ 제품 검토의 집중성
④ 참가인원의 제한성

76. 소프트웨어 재공학은 어떤 유지보수 측면에서 소프트웨어 위기를 해결하려고 하는 방법인가?

① 수정(Corrective) 유지보수

② 적응(Adaptive) 유지보수

③ 완전화(Perfective) 유지보수

④ 예방(Preventive) 유지보수

77. 소프트웨어 재사용에 대한 설명으로 틀린 것은?

① 새로운 개발 방법론의 도입이 용이하다.

② 개발 시간과 비용이 감소한다.

③ 프로그램 생성 지식을 공유할 수 있다.

④ 기존 소프트웨어에 재사용 소프트웨어를 추가하기 어려운 문제점이 발생할 수 있다.

78. 비용예측방법에서 원시 프로그램의 규모에 의한 방법(COCOMO model)중 초대형 규모의 트랜잭션 처리시스템이나 운영체제 등의 소프트웨어를 개발하는 유형은?

① Organic ② Semi-detached

③ Embedded ④ Sequential

79. 소프트웨어 설계 시 제일 상위에 있는 main user function에서 시작하여 기능을 하위 기능들로 분할해 가면서 설계하는 방식은?

① 객체 지향 설계 ② 데이터 흐름 설계

③ 상향식 설계 ④ 하향식 설계

80. 어떤 모듈이 다른 모듈의 내부 논리 조직을 제어하기 위한 목적으로 제어신호를 이용하여 통신하는 경우이며, 하위 모듈에서 상위 모듈로 제어신호가 이동하여 상위 모듈에게 처리 명령을 부여하는 권리 전도현상이 발생하게 되는 결합도는?

① data coupling ② stamp coupling

③ control coupling ④ common coupling

정보처리기사

제4과목 소프트웨어공학
17 모의고사2

문혜영교수

61. 객체지향 테스트 중 구조적 기법에서의 단위 테스트(Unit Test)와 같은 개념은?

① 메소드 ② 클래스

③ 필드 ④ 서브시스템

62. 구현 단계에서의 작업 절차를 순서에 맞게 나열한 것은?

㉠ 코딩한다.
㉡ 코딩작업을 계획한다.
㉢ 코드를 테스트한다.
㉣ 컴파일 한다.

① ㉠-㉡-㉢-㉣

② ㉡-㉠-㉣-㉢

③ ㉢-㉠-㉡-㉣

④ ㉣-㉡-㉠-㉢

63. 화이트박스 테스트에 대한 설명으로 가장 옳지 않은 것은?

① 제품의 내부 요소들이 명세서에 따라 수행되고 충분히 실행되는가를 보장하기 위한 검사이다.

② 모듈 안의 작동을 직접 관찰한다.

③ 프로그램 원시 코드의 논리적인 구조를 커버하도록 테스트 케이스를 설계한다.

④ 화이트박스 테스트 기법에는 조건 검사, 루프 검사, 비교 검사 등이 있다.

64. 소프트웨어 위험의 대표적 특성으로 짝지어진 것 중 가장 적합한 것은?

① 연쇄작용, 확실성　　② 불확실성, 손실

③ 연쇄작용, 예측　　④ 확실성, 예측

65. Software Engineering의 Engineering이 가지는 의미와 가장 관계없는 것은?

① 예술성 ② 경제성

③ 보편타당성 ④ 적시성

66. 객체지향 기법에서 다음 설명에 해당하는 것으로 가장 옳은 것은?

- 다른 객체에게 자신의 정보를 숨기고 자신의 연산만을 통하여 접근한다.
- 유지보수와 소프트웨어 확장 시 오류를 최소화할 수 있다.

① Abstraction ② Inheritance

③ Information Hiding ④ Polymorphism

67. 모듈의 응집도(Cohesion)에 대한 설명으로 틀린 것은?

① 모듈의 응집도란 모듈안의 요소들이 서로 관련되어 있는 정도를 말한다.

② 기능적 응집도(Functional Cohesion)는 한 모듈 내부의 한 기능 요소에 의한 출력 자료가 다음 기능 원소의 입력 자료로서 제공되는 형태이다.

③ 교환적 응집도(Communication Cohesion)는 동일한 입력과 출력을 사용하는 소작업들이 모인 모듈에서 볼 수 있다.

④ 논리적 응집도(Logical Cohesion)는 유사한 성격을 갖거나 특정형태로 분류되는 처리요소들로 하나의 모듈이 형성되는 경우이다.

68. 소프트웨어 재공학의 주요활동 중 다음 설명에 해당하는 것은?

기존 소프트웨어를 분석하여 설계를 추정하는 것으로서, 소프트웨어 개발 과정과 데이터 처리 과정을 설명하는 분석 및 설계 정보를 재발견하거나 다시 만들어 내는 작업

① Analysis
② Restructuring
③ Reverse Engineering
④ Migration

69. Putnam 모형을 기초로 해서 만든 자동화 추정 도구는?

① BYL ② SLIM

③ ESTIMACS ④ PERT

70. 자료 흐름도(DFD)를 작성하는데 지침이 될 수 없는 항목은?

① 자료 흐름은 처리(Process)를 거쳐 변환 될 때마다 새로운 이름을 부여한다.

② 어떤 처리(Process)가 출력자료를 산출하기 위해서는 반드시 입력 자료가 발생해야 한다.

③ 자료저장소에 입력 화살표가 있으면 반드시 출력 화살표도 표시되어야 한다.

④ 상위 단계의 처리(Process)와 하위 자료흐름도의 자료흐름은 서로 일치되어야 한다.

71. 소프트웨어 품질보증에서 FTR의 지침 사항으로 가장 옳지 않은 것은?

① 논쟁과 반박을 제한하지 않는다.

② 자원과 시간 일정을 할당한다.

③ 문제 영역을 명확히 표현한다.

④ 모든 검토자들을 위해 의미 있는 훈련을 행한다.

72. 설계 기법 중 하향식 설계 방법과 상향식 설계 방법에 대한 비교 설명으로 가장 옳지 않은 것은?

① 하향식 설계에서는 통합 검사 시 인터페이스가 이미 정의 되어 있어 통합이 간단하다.

② 하향식 설계에서 레벨이 낮은 데이터 구조의 세부 사항은 설계 초기 단계에서 필요하다.

③ 상향식 설계는 최하위 수준에서 각각의 모듈들을 설계하고 이러한 모듈이 완성되면 이들을 결합하여 검사한다.

④ 상향식 설계에서는 인터페이스가 이미 성립되어 있지 않더라도 기능 추가가 쉽다.

73. 소프트웨어의 위기현상과 가장 거리가 먼 것은?

① 유지보수의 어려움

② 개발인력의 급증

③ 성능 및 신뢰성의 부족

④ 개발기간의 지연 및 개발비용의 증가

74. 객체지향 분석 방법론 중 E-R 다이어그램을 사용하여 객체의 행위를 모델링하며, 객체 식별, 구조식별, 주제 정의, 속성과 인스턴스 연결 정의, 연산과 메시지 연결 정의 등의 과정으로 구성되는 것은?

① Coad와 Yourdon 방법
② Booch 방법
③ Jacobson 방법
④ Wirfs-Brock 방법

75. LOC 기법에 의하여 예측된 총 라인수가 50000라인, 프로그래머의 월 평균 생산성이 200라인, 개발에 참여할 프로그래머가 10 인 일 때, 개발 소요 기간은?

① 25개월 ② 50개월

③ 200개월 ④ 2000개월

76. 다음 중 가장 약한 결합도(Coupling)는?

① Common Coupling
② Control Coupling
③ External Coupling
④ Stamp Coupling

77. 나선형 모형의 각 개발 단계에 대한 설명으로 가장 옳은 것은?

① Planning 단계에서는 위험 요소와 타당성을 분석하여 프로젝트의 추진 여부를 결정한다.

② Development 단계에서는 선택된 기능을 수행하는 프로토 타입을 개발한다.

③ Risk Analysis 단계에서는 개발 목적과 기능 선택, 제약조건 등을 결정하고 분석한다.

④ Evaluation 단계에서는 고객 평가와 검증 과정을 수행하여 개발된 결과를 평가한다.

78. CASE의 주요기능으로 가장 옳지 않은 것은?

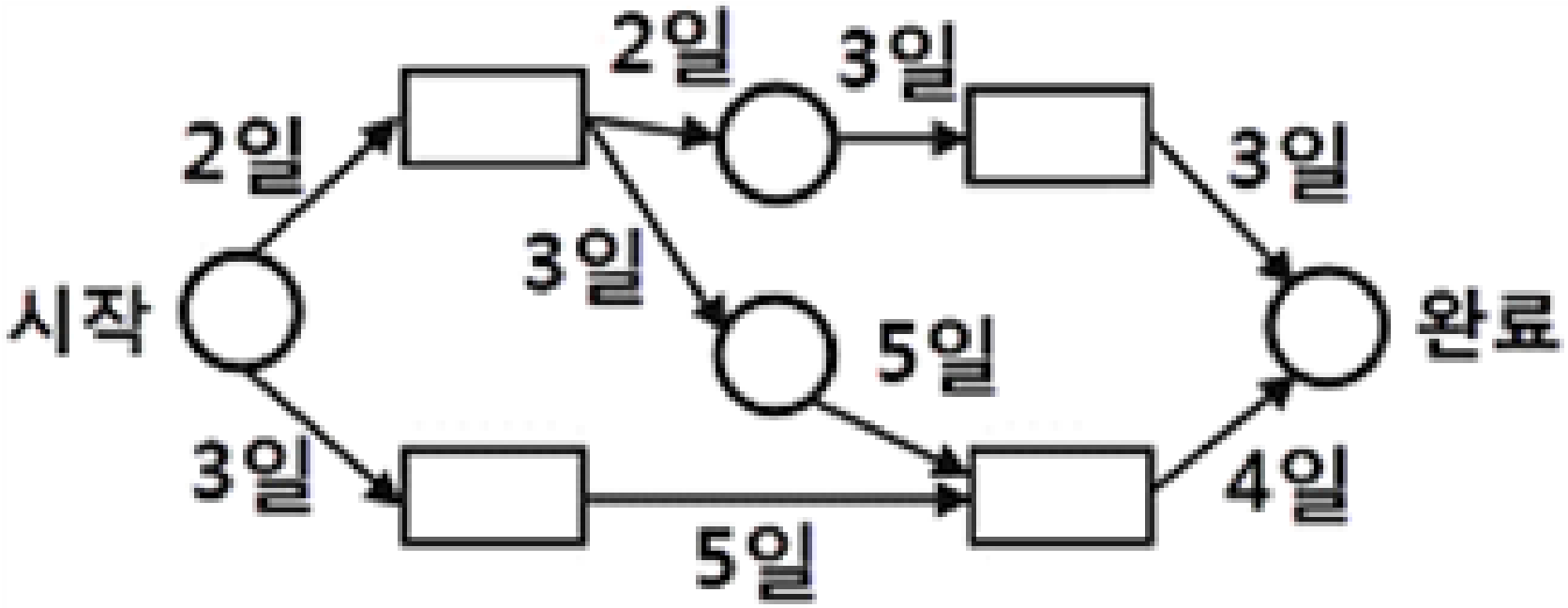
① S/W 라이프 사이클 전 단계의 연결

② 그래픽 지원

③ 다양한 소프트웨어 개발 모형 지원

④ 언어 번역

79. CPM 네트워크가 다음과 같을 때 임계경로의 소요기일은?



① 10일 ② 12일

③ 14일 ④ 16일

80. 공학적으로 잘 작성된 소프트웨어의 특성에 관한 설명으로 가장 옳지 않은 것은?

① 소프트웨어는 신뢰성이 높아야 하며 효율적이어야 한다.

② 소프트웨어는 사용자가 원하는 대로 동작해야 한다.

③ 소프트웨어는 편리성이나 유지보수성에 점차 비중을 적게 두는 경향이 있다.

④ 소프트웨어는 잠재적인 오류가 가능한 적어야 하며 유지보수가 용이해야 한다.