

정보처리기사 필기

소프트웨어 설계

2

요구사항 확인 ②

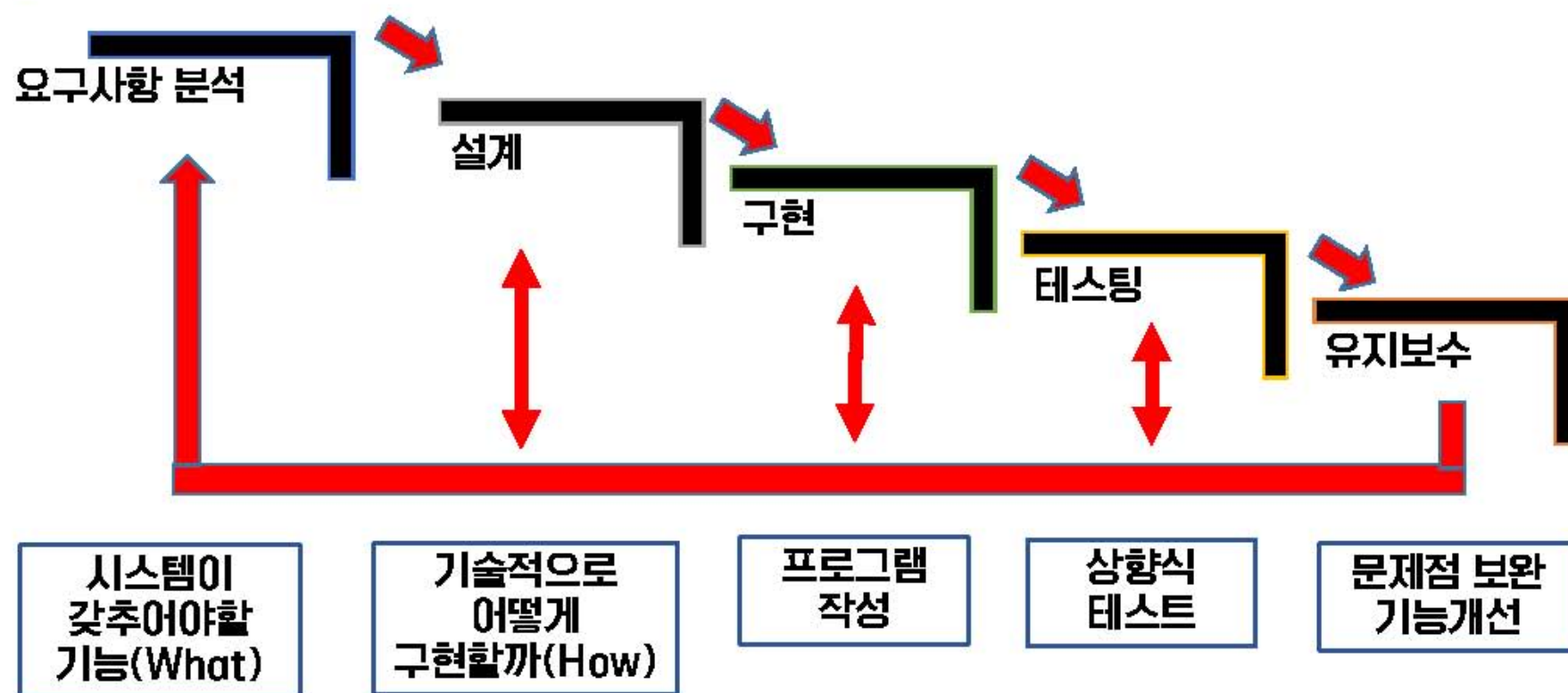
양문자 선생님

출처 : ncs 학습모듈(NCS능력단위 요구사항확인)

참고 : 시스템분석설계/이창희/정익사

소프트웨어 개발 방법론

소프트웨어 개발 과정



소프트웨어 개발 방법론

소프트웨어 개발 생명주기 모델

- 소프트웨어를 어떻게 개발할 것인가에 대한 전체적인 큰 흐름

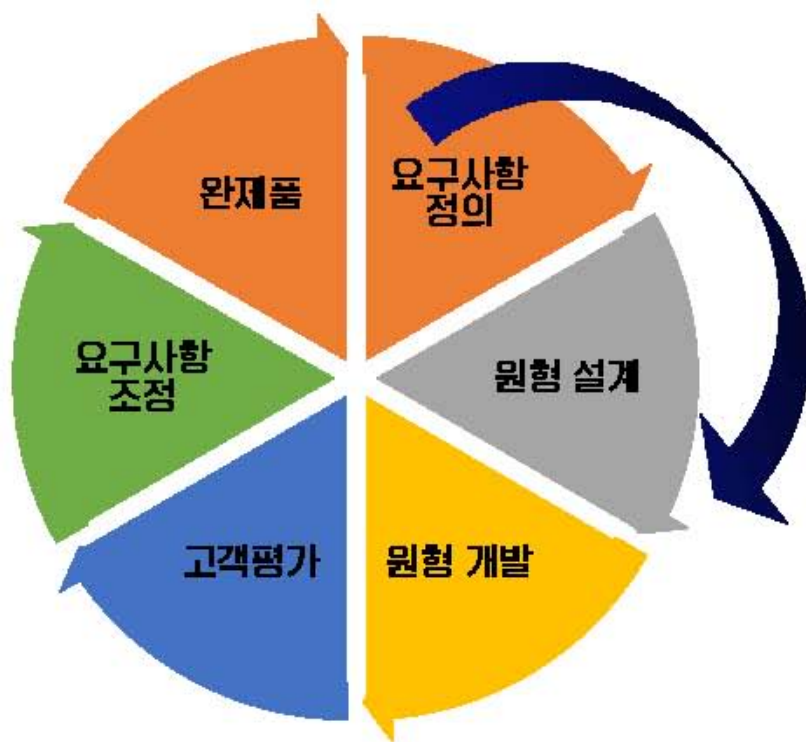


<폭포수 모델>

- 앞 단계가 완료될 때까지 대기 상태
- 완성된 모습을 후반부가 되기 전엔 볼 수 없다.
- 고객이 원하는 모습이 아니라면 수정이 어렵다.

소프트웨어 개발 방법론

소프트웨어 개발 생명주기 모델



<원형 모델>

- 점진적으로 시스템을 개발해 나가는 방법
- 원형(Prototype)을 가능한 빨리 개발한 후 고객과 검증하는 것이 목적

소프트웨어 개발 방법론

■ 소프트웨어 개발 생명주기 모델

1. 계획 및 정의

2. 위험 분석



4. 고객 평가

3. 개발

<나선형 모델>

- 고비용의 시스템 개발이나 큰 시스템 구축 시 효과적
- 프로젝트 수행 시 발생하는 위험을 최소화하려는 목적
- 개발자가 위험을 정확하게 분석하지 못했다면 심각한 문제

소프트웨어 개발 방법론

- 소프트웨어를 어떻게 개발할 것인가에 대한 전체적인 큰 흐름과 더불어 각 단계에서의 방법과 활동들을 포함
- 보다 상세하고 구체적인 프로세스들의 집합

소프트웨어 개발 방법론

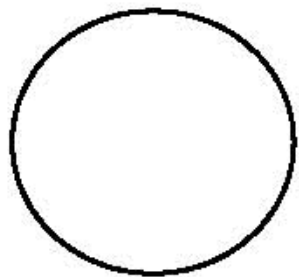
<구조적 분석>

- 효율적인 시스템 분석 명세서를 작성하기 위한
- 도형 중심의 분석용 도구를 이용하여
- 사용자의 요구사항을 파악하고 문서화하는
- 체계적인 분석 기법
- 사용자의 요구 사항을 정확하게 작성할 수 있음
- Yourdon등에 의해 개발되어 보급(1960년말 ~ 1980년대)

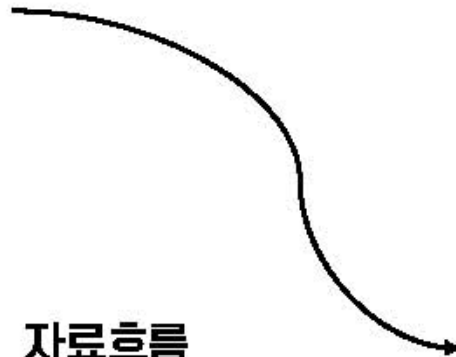


소프트웨어 개발 방법론

- 자료흐름도(Data Flow Diagram : DFD)



처리



자료흐름



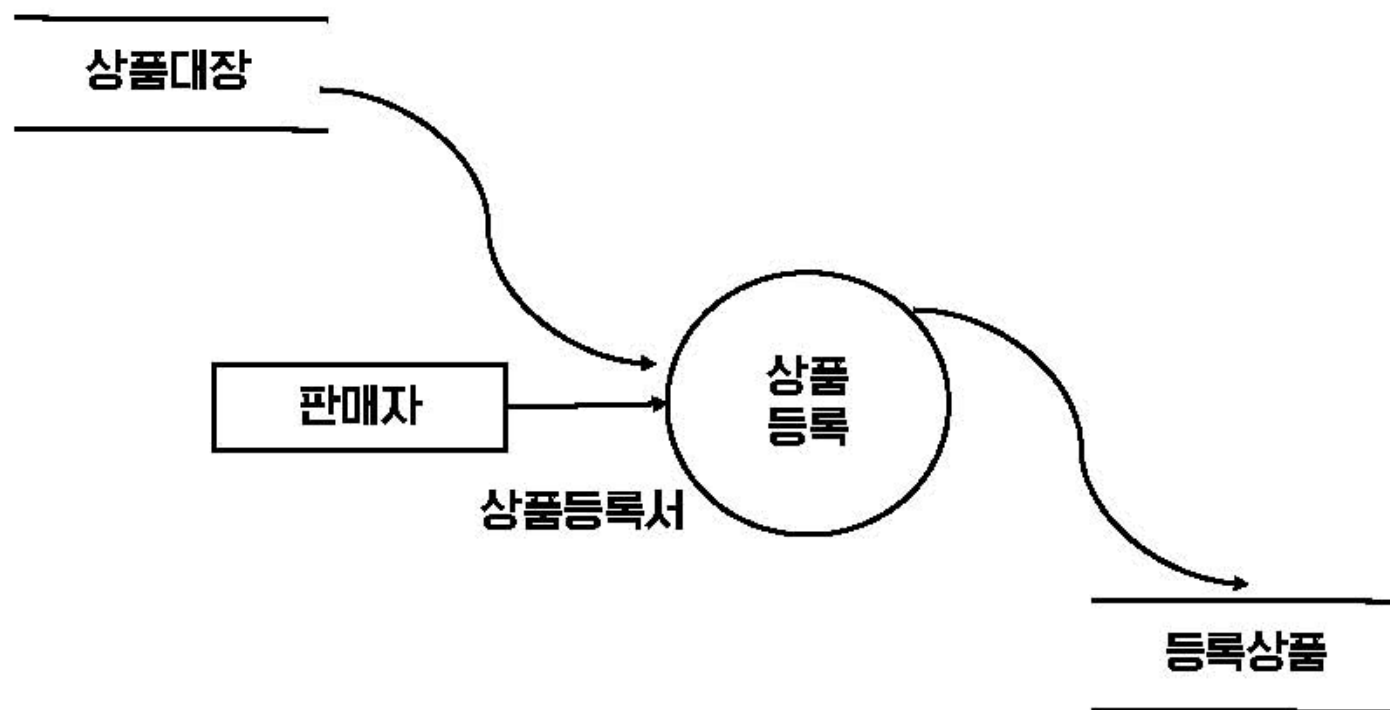
자료저장소



단말

소프트웨어 개발 방법론

- 자료흐름도(Data Flow Diagram : DFD)



소프트웨어 개발 방법론

- 자료사전(Data Dictionary: DD)

기호	뜻
=	자료 항목 정의
+	복합적인 자료 요소의 구성
{ }	반복
[]	선택
()	생략가능
**	주석
또는 ;	대체 항목 나열

소프트웨어 개발 방법론

- 자료사전(Data Dictionary: DD)

계산서 = { 계산서 번호 + 고객 이름 + 고객연락처 + 거래일 + { 지불 기한 일자 } +
 { 판매일자 + 판매번호 + { 항목내역 + 단가 + 수량 } + 소계 + 세금 + 총액 } }

계산방법 = [현금 | 카드 | 계좌이체 | pay]

고객파일 = * 고객 파일의 구성은 고객이름에 따라 순차적임 *
 = { 고객번호 + 고객이름 + 고객등급 + 입금사항 }

소프트웨어 개발 방법론

- 소단위 명세서(Mini Specification) : 자료흐름도에 나타난 모든 최소 단위의 처리에 대해 자료흐름이 변환되는 절차 또는 논리적인 활동을 기술하는 도구



소프트웨어 개발 방법론

- 소단위 명세서(Mini Specification) : 자료흐름도에 나타난 모든 최소 단위의 처리에 대해 자료흐름이 변환되는 절차 또는 논리적인 활동을 기술하는 도구

for 각 시간제 근로자에 대해

1. 시간제 근로자의 근로 형태를 파악

2. if 근로형태 = "장기"

1. then 급여 = 근무시간 * 12,000

2. else 급여 = 근무시간 * 10,000

3. 이름과 급여액을 급여 내역서에 기록

구조적언어


	사은품	쿠폰지급	배달료	합인
1. 50이상	Y	Y	N	N
2. 50이하	Y	N	Y	N
1. 처리1	-	X	-	X
2. 처리2	X	X	X	X
3. 처리3	-	-	X	X


의사결정표

소프트웨어 개발 방법론

애자일(Agile)

- 2001년 'Agile연합' 모임 결성
- 'Agile 소프트웨어 개발 선언문' 발표
- 원칙 - '가볍지만 충분한'

- 
- ★ 프로세스와 도구
 - ★ 문서화
 - ★ 고객과의 계약 협상

- 
- ★ 개인과의 상호작용
 - ★ 제대로 동작하는 소프트웨어의 개발에 집중
 - ★ 고객과의 협력

소프트웨어 개발 방법론

애자일(Agile) 방법론 유형

XP(eXtreme Programming)

- 의사소통 개선과 즉각적 피드백
- 5가지 가치 - 용기, 단순성, 의사소통, 피드백, 존중

스크럼(SCRUM)

- 프로젝트 관리를 위한 상호,점진적 개발방법론
- XP와 달리 진행 체계 수립, 역할, 정의에 중점
- 기존 폭포수 모델이나 프로토타이핑같은 모델과 달리 모든 LifeCycle를 담지 않는다.
- 5가지 가치 -확약, 전념, 정직, 존중, 용기

린(LEAN)

- 도요타사의 대표적인 생산방식
- 인력, 생산 설비 등을 필요한 만큼만 유지하면서 생산효율을 극대화하는 방식
- 전적으로 고객 관점

소프트웨어 개발 방법론

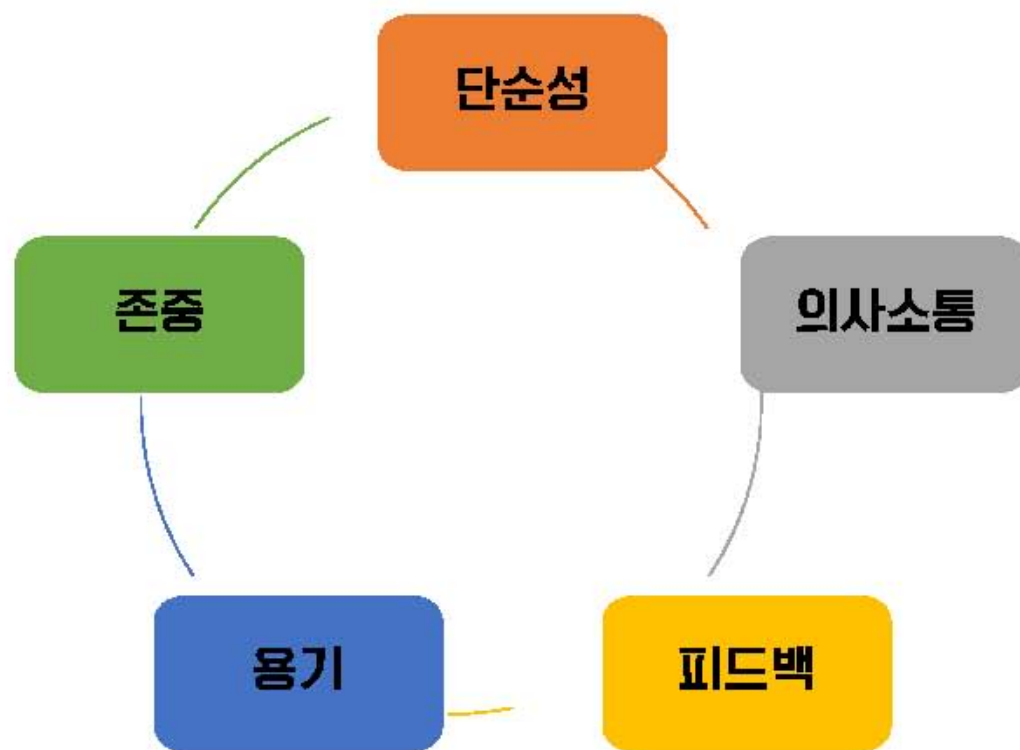
XP(eXtreme Programming)

- 애자일(Agile)의 가치를 적용한 개발방법론
- '고객의 요구사항은 변경된다' -> 고객과 개발팀의 커뮤니케이션 주장

용어	내용
스토리	요구사항(고객)
스토리 추정	스토리를 보고 기간과 강도 결정(개발자)
릴리즈	고객에게 구현된 제품 배포
반복	릴리즈 안에서 반복되는 작업
드라이버	코드 작성자
파트너	드라이버를 도와 조언

<XP 용어>

XP(eXtreme Programming)의 5가지 가치



XP(eXtreme Programming)의 12가지 실천사항

계획게임 (Planning game) or Planning process	짝 프로그래밍(Pair programming)
짧은 릴리즈 주기(Small/short releases)	코드 공동 소유(Collective ownership)
메타포(공통의 name system)	지속적인 통합, CI(Continuous integration)
단순 설계(Simple design)	주당 40시간 작업(40-hour week)
테스트 우선 개발(Testing)	고객의 참여(On-site customer)
리팩토링(Refactoring)	코딩 표준(Coding standards)

소프트웨어 개발 방법론

〈객체지향 방법론〉

- 소프트웨어 개발 과정은 기존 시스템을 분석하고 이를 표준화된 새로운 시스템 환경으로 전환하는데 초점을 맞추게 되었다.
- 소프트웨어를 개발할 때 작은 단위의 모듈들을 구성하여 만들고 추후 이를 재사용하여 소프트웨어의 효율성을 높이자는 생각에서 객체지향 방법론이 발전하게 되었다.
- 여러 객체지향 방법론들이 난립하게 되고 표기법만이라도 통일하자는 제안에 의해 UML을 사용하게 되었다.
- 객체, 클래스, 캡슐화, 데이터 은닉, 상속, 조합, 다형성의 개념

소프트웨어 개발 방법론

- 객체, 클래스, 캡슐화, 데이터 은닉, 상속, 다형성의 개념

객체	업무수행을 위한 대상이 되는 사람, 장소, 사물, 사건 및 개념
클래스	공통 속성과 행위를 가진 객체를 묶어 추상화한 개념
캡슐화	구현부가 외부에 노출되지 않게 싸여진 상태
데이터 은닉	각각의 객체가 자신의 속성(데이터)과 메서드(행위)를 다른 객체에게 숨기고 있는 것
상속	클래스가 가진 속성과 행위를 객체가 물려 받는것
조합	다른 객체를 사용하여 객체를 구성하는 것. 보다 복잡한 클래스를 만드는 일종의 '조립'
다형성	같은 메서드에 다르게 반응하는것.

소프트웨어 개발 방법론

<SOLID (객체 지향 설계)>

컴퓨터 프로그래밍에서 SOLID란 로버트 마틴이 2000년대 초반에 명명한 객체 지향 프로그래밍 및 설계의 다섯 가지 기본 원칙을 마이클 페더스가 두문자어 기억술로 소개한 것이다.

S	SRP	단일 책임 원칙 (Single responsibility principle)한 클래스는 하나의 책임만 가져야 한다.
O	OCP	개방-폐쇄 원칙 (Open/closed principle)“소프트웨어 요소는 확장에는 열려 있으나 변경에는 닫혀 있어야 한다.”
L	LSP	리스코프 치환 원칙 (Liskov substitution principle)“프로그램의 객체는 프로그램의 정확성을 깨뜨리지 않으면서 하위 타입의 인스턴스로 바꿀 수 있어야 한다.” 계약에 의한 설계를 참고하라.
I	ISP	인터페이스 분리 원칙 (Interface segregation principle)“특정 클라이언트를 위한 인터페이스 여러 개가 범용 인터페이스 하나보다 낫다.”
D	DIP	의존관계 역전 원칙 (Dependency inversion principle)프로그래머는 “추상화에 의존해야지, 구체화에 의존하면 안된다.” 의존성 주입은 이 원칙을 따르는 방법 중 하나다.

- 위키백과 -

소프트웨어 개발 방법론



기법	원리	특징
Rumbaugh(럼바우)의 OMT	<ul style="list-style-type: none"> 클래스의 외부 명세를 정의함 객체 모델링(객체 다이어그램), 동적 모델링(상태 다이어그램), 기능 모델링(자료흐름도)으로 분류함 	소프트웨어 생명 주기 지원 데이터베이스 구조화에 용이
Booch 방법론	<ul style="list-style-type: none"> 시스템 형성 구조를 모형화하는 DFD사용 클래스 다이어그램, 객체 다이어그램, 모듈 다이어그램, 프로세스 다이어그램 	전체 시스템 가시화에 유용함 실시간 처리에 유용함 설계를 위한 문서화 기법을 강조 분석 단계가 취약함
Coad/Yaurdon 방법론	<ul style="list-style-type: none"> 객체지향 특징을 가장 충족시키는 방법 	객체지향 CASE Tool 지원
shaler/Mellor 방법론	<ul style="list-style-type: none"> 하나의 시스템을 몇 개의 영역으로 분할하여 서브 시스템을 구성함 정보 모델링, 상태 모델링, 처리 모델링 	시스템의 분해성이 우수함 대형 시스템 개발에 유리함

소프트웨어 개발 방법론

문제풀이

1. XP(eXtreme Programming)의 기본원리로 볼 수 없는 것은?

- ① Linear Sequential Method ② Pair Programming
- ③ Collective Ownership ④ Continuous Integration

2. 애자일 방법론에 해당하지 않는 것은?

- ① 기능중심 개발
- ② 스크럼
- ③ 익스트림 프로그래밍
- ④ 모듈중심 개발

3. 소프트웨어 개발 단계에서 요구 분석 과정에 대한 설명으로 거리가 먼 것은?

- ① 분석 결과의 문서화를 통해 향후 유지보수에 유용하게 활용 할 수 있다.
- ② 개발 비용이 가장 많이 소요되는 단계이다.
- ③ 자료흐름도, 자료 사전 등이 효과적으로 이용될 수 있다.
- ④ 보다 구체적인 명세를 위해 소단위 명세서 (Mini-Spec)가 활용될 수 있다.

(2020년 4회 정보처리기사 필기 기출문제 소프트웨어 설계)

문제풀이

4. 럼바우(Rumbaugh) 객체지향 분석 기법에서 동적 모델링에 활용되는 다이어그램은?

- ① 객체 다이어그램(Object Diagram)
- ② 패키지 다이어그램(Package Diagram)
- ③ 상태 다이어그램(State Diagram)
- ④ 자료 흐름도(Data Flow Diagram)

5. 그래픽 표기법을 이용하여 소프트웨어 구성 요소를 모델링하는 럼바우 분석 기법에 포함되지 않는 것은?

- ① 객체 모델링
- ② 기능 모델링
- ③ 동적 모델링
- ④ 블랙박스 분석 모델링

(2020년 4회 정보처리기사 필기 기출문제 소프트웨어 설계)

문제풀이

(2020년 4회 정보처리기사 필기 기출문제 소프트웨어 설계)

6. DFD(data flow diagram)에 대한 설명으로 틀린 것은?

- ① 자료 흐름 그래프 또는 버블(bubble) 차트라고도 한다.
- ② 구조적 분석 기법에 이용된다.
- ③ 시간 흐름을 명확하게 표현할 수 있다.
- ④ DFD의 요소는 화살표, 원, 사각형, 직선(단선/이중선)으로 표시한다.

7. 다음 중 자료사전(Data Dictionary)에서 선택의 의미를 나타내는 것은?

- ① [] ② { }
- ③ + ④ =

8. 객체지향 기법의 캡슐화(Encapsulation)에 대한 설명으로 틀린 것은?

- ① 인터페이스가 단순화 된다.
- ② 소프트웨어 재사용성이 높아진다.
- ③ 변경 발생 시 오류의 파급효과가 적다.
- ④ 상위 클래스의 모든 속성과 연산을 하위 클래스가 물려받는 것을 의미한다.

문제풀이

9. 다음 내용이 설명하는 객체지향 설계 원칙은?

- 클라이언트는 자신이 사용하지 않는 메서드와 의존관계를 맺으면 안 된다.
- 클라이언트가 사용하지 않는 인터페이스 때문에 영향을 받아서는 안 된다.

① 인터페이스 분리 원칙

② 단일 책임 원칙

③ 개방 폐쇄의 원칙

④ 리스코프 교체의 원칙

(2020년 4회 정보처리기사 필기 기출문제 소프트웨어 설계)

소프트웨어 개발 방법론

문제풀이

10. 애자일 기법에 대한 설명으로 맞지 않은 것은?

- ① 절차와 도구보다 개인과 소통을 중요하게 생각한다.
- ② 계획에 중점을 두어 변경 대응이 난해하다.
- ③ 소프트웨어가 잘 실행되는데 가치를 둔다.
- ④ 고객과의 피드백을 중요하게 생각한다.

11. 자료 사전에서 자료의 반복을 의미하는 것은?

- ① = ② () ③ { } ④ []

12. 자료흐름도(Data Flow Diagram)의 구성요소로 옳은 것은?

- ① process, data flow, data store, comment
- ② process, data flow, data store, terminator
- ③ data flow, data store, terminator, data dictionary
- ④ process, data store, terminator, mini-spec

(2020년 3회 정보처리기사 필기 기출문제 소프트웨어 설계)

소프트웨어 개발 방법론

문제풀이

13. 객체 지향 소프트웨어 공학에서 하나 이상의 유사한 객체들을 묶어서 하나의 공통된 특성을 표현한 것은?

- ① 트랜잭션 ② 클래스 ③ 시퀀스 ④ 서브루틴

14. 객체지향 설계 원칙 중, 서브타입(상속받은 하위 클래스)은 어디에서나 자신의 기반타입(상위 클래스)으로 교체할 수 있어야 함을 의미하는 원칙은?

- ① ISP(Interface Segregation Principle) ② DIP(Dependency Inversion Principle)
③ LSP(Liskov Substitution Principle) ④ SRP(Single Responsibility Principle)

(2020년 3회 정보처리기사 필기 기출문제 소프트웨어 설계)

소프트웨어 개발 방법론

문제풀이

15. 자료 사전에서 자료의 생략을 의미하는 기호는?

- ① { } ② ** ③ = ④ ()

16. XP(eXtreme Programing)의 5가지 가치로 거리가 먼 것은?

- ① 용기 ② 의사소통 ③ 정형 분석 ④ 피드백

(2020년 1,2회 정보처리기사 필기 기출문제 소프트웨어 설계)

소프트웨어 개발 방법론

문제풀이

17. 럼바우(Rumbaugh)의 객체지향 분석 절차를 가장 바르게 나열한 것은?

- ① 객체 모형 → 동적 모형 → 기능 모형
- ② 객체 모형 → 기능 모형 → 동적 모형
- ③ 기능 모형 → 동적 모형 → 객체 모형
- ④ 기능 모형 → 객체 모형 → 동적 모형

18. 데이터 흐름도(DFD)의 구성요소에 포함되지 않는 것은?

- ① process ② data flow
- ③ data store ④ data dictionary

(2020년 1,2회 정보처리기사 필기 기출문제 소프트웨어 설계)