필기수록 예상문제
2020년 3회 대비 정보처리기사 실기

***** 수험자 유의사항 *****

1. 시험 문제지를 받는 즉시 응시하고자 하는 종목의 문제지가 맞는지를 확인하여야 합니다.
2. 시험 문제지 총면수·문제번호 순서·인쇄상태 등을 확인하고, 수험번호 및 성명을 답안지에 기재하여야 합니다.
3. 문제 및 답안(지), 채점기준은 일절 공개하지 않으며 자신이 작성한 답안, 문제 내용 등을 수험표 등에 이기(옮겨 적는 행위) 등은 관련 법 등에 의거 불이익 조치 될 수 있으니 유의하시기 바랍니다.
4. 수험자 인적사항 및 답안작성(계산식 포함)은 흑색 필기구만 사용하여야 하며 흑색을 제외한 유색 필기구 또는 연필류를 사용하였을 경우 그 문항은 0점 처리됩니다.
5. 답란(답안 기재란)에는 문제와 관련 없는 불필요한 낙서나 특이한 기록사항 등을 기재하여서는 안되며 부정의 목적으로 특이한 표식을 하였다고 판단될 경우에는 모든 문항이 0점 처리됩니다.
6. 답안을 정정할 때에는 반드시 정정부분을 두 줄(=)로 그어 표시하여야 하며, 두 줄로 긋지 않은 답안은 정정하지 않은 것으로 간주합니다. (수정테이프, 수정액 사용불가)
7. 답안의 한글 또는 영문의 오탈자는 오답으로 처리됩니다. 단, 답안에서 영문의 대·소문자 구분, 띄어쓰기는 여부에 관계 없이 채점합니다.
8. 계산 또는 디버깅 등 계산 연습이 필요한 경우는 <문 제> 아래의 연습란을 사용하시기 바라며, 연습란은 채점대상이 아닙니다.
9. 문제에서 요구한 가지 수(항수) 이상을 답란에 표기한 경우에는 답안기재 순으로 요구한 가지 수(항수)만 채점하고 한 항에 여러 가지를 기재하더라도 한 가지로 보며 그 중 정답과 오답이 함께 기재란에 있을 경우 오답으로 처리됩니다.
10. 한 문제에서 소문제로 파생되는 문제나, 가지수를 요구하는 문제는 대부분의 경우 부분채점을 적용합니다. 그러나 소문제로 파생되는 문제 내에서의 부분 배점은 적용하지 않습니다.
11. 답안은 문제의 마지막에 있는 답란에 작성하여야 합니다.
12. 부정 또는 불공정한 방법(시험문제 내용과 관련된 메모지사용 등)으로 시험을 치른 자는 부정행위자로 처리되어 당해 시험을 중지 또는 무효로 하고, 2년간 국가기술자격검정의 응시자격이 정지됩니다.
13. 시험위원이 시험 중 신분확인을 위하여 신분증과 수험표를 요구할 경우 반드시 제시하여야 합니다.
14. 시험 중에는 통신기기 및 전자기기(휴대용 전화기 등)를 지참하거나 사용할 수 없습니다.
15. 국가기술자격 시험문제는 일부 또는 전부가 저작권법상 보호되는 저작물이고, 저작권자는 한국산업인력공단입니다. 문제의 일부 또는 전부를 무단 복제, 배포, 출판, 전자출판 하는 등 저작권을 침해하는 일체의 행위를 금합니다.

※ 수험자 유의사항 미준수로 인한 채점상의 불이익은 수험자 본인에게 전적으로 책임이 있음

실기 교재에서 다루지 않아 필기 교재에만 수록되어 있는 내용을 모두 문제와 해설로 제공해드리는 자료입니다.
문제의 지문과 정답의 해설까지 모두 빠뜨리지 않고 학습하도록 하세요.

문제 1 **다음이 설명하고 있는 소프트웨어 설계와 관련 용어를 쓰시오.**

- 소프트웨어 개발 방법론의 바탕이 되는 것으로, 소프트웨어를 개발하기 위해 정의하고 운용, 유지보수 등의 과정을 각 단계별로 나눈 것이다.
- 소프트웨어 개발 단계와 각 단계별 주요 활동, 그리고 활동의 결과에 대한 산출물로 표현한다.
- 일반적으로 사용되는 모형에는 폭포수 모형, 프로토타입 모형, 나선형 모형, 애자일 모형 등이 있다.

답 : 소프트웨어 생명 주기(Software Life Cycle)

문제 2 **소프트웨어 개발 방법론과 관련하여 다음 설명에 해당하는 모형이 무엇인지 쓰시오.**

- 소프트웨어 개발 이전 단계로 돌아갈 수 없다는 전제하에 각 단계를 확실히 매듭짓고 그 결과를 철저하게 검토하여 승인 과정을 거친 후에 다음 단계를 진행하는 개발 방법론이다.
- 소프트웨어 공학에서 가장 오래되고 가장 폭넓게 사용된 전통적인 소프트웨어 생명 주기 모형으로, 고전적 생명 주기 모형이라고도 한다.
- 소프트웨어 개발 과정의 한 단계가 끝나야만 다음 단계로 넘어갈 수 있는 선형 순차적 모형이다.

답 : 폭포수 모형(Waterfall Model)

문제 3 **소프트웨어 개발 방법론 중 프로토타입 모형(Prototype Model)에 대해 간략히 서술하시오.**

답 : 사용자의 요구사항을 정확히 파악하기 위해 실제 개발될 소프트웨어에 대한 시제품을 만들어 최종 결과물을 예측하는 모형이다.

※ 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

[병행학습]

프로토타입 모형(Prototype Model, 원형 모형)

- 사용자의 요구사항을 정확히 파악하기 위해 실제 개발될 소프트웨어에 대한 시제(견본)품(Prototype)을 만들어 최종 결과물을 예측하는 모형이다.
- 시제품은 사용자와 시스템 사이의 인터페이스에 중점을 두어 개발한다.
- 소프트웨어의 개발이 완료된 시점에서 오류가 발견되는 폭포수 모형의 단점을 보완하기 위한 모형이다.

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 4 **소프트웨어 개발 방법론과 관련하여 다음 설명에 해당하는 모형이 무엇인지 쓰시오.**

• 폭포수 모형과 프로토타입 모형의 장점에 위험 분석 기능을 추가한 모형으로, 보헴(Boehm)이 제안하였다. • 여러 번의 소프트웨어 개발 과정을 거쳐 점진적으로 완벽한 최종 소프트웨어를 개발하는 것이다. • 소프트웨어를 개발하면서 발생할 수 있는 위험을 관리하고 최소화하는 것을 목적으로 한다. • 점진적으로 개발 과정이 반복되므로 누락되거나 추가된 요구사항을 첨가할 수 있고, 정밀하며, 유지보수 과정이 필요 없다.

답 : 나선형 모형(Spiral Model, 점진적 모형)

문제 5 **소프트웨어 생명 주기 모형에 대한 다음 설명에서 괄호에 들어갈 알맞은 용어를 한글 또는 영문(Fullname 또는 약어)으로 쓰시오.**

• ( ) 모형은 고객의 요구사항 변화에 유연하게 대응할 수 있도록 일정한 주기를 반복하면서 개발과정을 진행한다. • 어느 특정 개발 방법론이 아니라 좋은 것을 빠르고 낭비 없게 만들기 위해 고객과의 소통에 초점을 맞춘 방법론을 통칭한다. • 각 개발주기에서는 고객이 요구사항에 우선순위를 부여하여 개발 작업을 진행한다. • 소규모 프로젝트, 고도로 숙달된 개발자, 급변하는 요구사항에 적합하다.

답 : 애자일 모형(Agile Model)

문제 6 **다음이 설명하고 있는 용어를 쓰시오.**

• 애자일 모형을 기반으로 하는 소프트웨어 개발 방법론이다. • 럭비에서 반칙으로 경기가 중단된 경우 양 팀의 선수들이 럭비공을 가운데 두고 상대팀을 밀치기 위해 서로 대치해 있는 대형을 가리키는 말에서 유래한 것으로 팀이 중심이 되어 개발의 효율성을 높인다는 의미가 내포된 용어이다. • 팀원 스스로가 팀을 구성(self-organizing)해야 하며, 개발 작업에 관한 모든 것을 스스로 해결(cross-functional)하는 팀 위주의 개발 방법론이다.

답 : 스크럼(Scrum) 기법

[병행학습]

스크럼(Scrum) 개발 프로세스 관련 용어

- **제품 백로그(Product Backlog)** : 제품 개발에 필요한 모든 요구사항(User Story)을 우선순위에 따라 나열한 목록

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

- **스프린트 계획 회의(Sprint Planning Meeting)** : 제품 백로그 중 이번 스프린트에서 수행할 작업을 대상으로 단기 일정을 수립하는 회의
- **스프린트(Sprint)** : 실제 개발 작업을 진행하는 과정으로, 보통 2 ~ 4주 정도의 기간 내에서 진행
- **일일 스크럼 회의(Daily Scrum Meeting)** : 모든 팀원이 매일 약속된 시간에 약 15분 정도의 짧은 시간동안 진행 상황을 점검하는 회의
- **스프린트 검토 회의(Sprint Review)** : 부분 또는 전체 완성 제품이 요구사항에 잘 부합되는지 사용자가 포함된 참석자 앞에서 테스팅을 수행하는 회의
- **스프린트 회고(Sprint Retrospective)** : 스프린트 주기를 되돌아보며 정해놓은 규칙을 잘 준수했는지, 개선할 점은 없는지 등을 확인하고 기록

문제 7 **애자일 기반의 개발 방법론과 관련하여 다음 설명에 해당하는 모형을 쓰시오.**

- 수시로 발생하는 고객의 요구사항에 유연하게 대응하기 위해 고객의 참여와 개발 과정의 반복을 극대화하여 개발 생산성을 향상시키는 모형이다.
- 짧고 반복적인 개발 주기, 단순한 설계, 고객의 적극적인 참여를 통해 소프트웨어를 빠르게 개발하는 것을 목적으로 한다.
- 릴리즈 테스트마다 고객을 직접 참여시킴으로써 요구한 기능이 제대로 작동하는지 고객이 직접 확인할 수 있다.
- 의사소통(Communication), 단순성(Simplicity), 용기(Courage), 존중(Respect), 피드백(Feedback)을 핵심 가치로 삼는다.

답 : XP(eXtreme Programming) 기법

[병행학습]

XP(eXtreme Programming) 개발 프로세스 관련 용어

- **사용자 스토리(User Story)** : 고객의 요구사항을 간단한 시나리오로 표현한 것
- **스파이크(Spike)** : 요구사항의 신뢰성을 높이고 기술 문제에 대한 위험을 감소시키기 위해 별도로 만드는 간단한 프로그램
- **릴리즈(Release)** : 몇 개의 스토리가 적용되어 부분적으로 기능이 완료된 제품을 제공하는 것
- **이터레이션(Iteration)** : 하나의 릴리즈를 더 세분화 한 단위

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 8 다음은 XP(eXtreme Programming) 개발 방법론의 주요 실천 방법(Practice)에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 실천 방법을 쓰시오.

실천 방법	내용
(　①　)	다른 사람과 함께 프로그래밍을 수행함으로써 개발에 대한 책임을 공동으로 나눠 갖는 환경을 조성한다.
Test-Driven Development (테스트 주도 개발)	• 개발자가 실제 코드를 작성하기 전에 테스트 케이스를 먼저 작성하므로 자신이 무엇을 해야 할지를 정확히 파악한다. • 테스트가 지속적으로 진행될 수 있도록 자동화된 테스팅 도구(구조, 프레임워크)를 사용한다.
(　②　)	개발에 참여하는 모든 구성원(고객 포함)들은 각자 자신의 역할이 있고 그 역할에 대한 책임을 가져야 한다.
Continuous Integration (계속적인 통합)	모듈 단위로 나눠서 개발된 코드들은 하나의 작업이 마무리될 때마다 지속적으로 통합된다.
Design Improvement (디자인 개선) 또는 Refactoring(리팩토링)	프로그램 기능의 변경 없이, 단순화, 유연성 강화 등을 통해 시스템을 재구성한다.
Small Releases (소규모 릴리즈)	릴리즈 기간을 짧게 반복함으로써 고객의 요구 변화에 신속히 대응할 수 있다.

답

• ① : Pair Programming(짝 프로그래밍)

• ② : Whole Team(전체 팀)

문제 9 다음이 설명하고 있는 용어를 영문(Fullname 또는 약어)으로 쓰시오.

• 소프트웨어의 품질 특성과 평가를 위한 국제 표준 지침이다. • 주요 품질 특성은 기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성이다. • 소프트웨어의 품질에 대한 요구사항을 기술하거나 개발중인 또는 개발이 완료된 소프트웨어의 품질 평가 등에 사용된다.

답 : ISO/IEC 9126

[병행학습]

ISO/IEC 25010

• 2011년 ISO/IEC 9126을 개정하여 만든 소프트웨어 제품에 대한 국제 표준이다.

• **품질 특성** : 신뢰성, 사용성, 이식성, 유지보수성, 기능 적합성, 실행 효율성, 호환성, 보완성

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

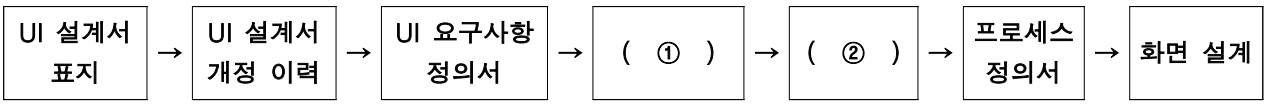
문제 10 **소프트웨어의 품질은 소프트웨어의 기능, 성능, 만족도 등 소프트웨어에 대한 요구사항이 얼마나 충족하는가를 나타내는 소프트웨어 특성의 총체이다. ISO/IEC 9126에서 정의하고 있는 6가지의 품질에 대한 특성 중 다음 괄호에 들어갈 알맞은 특성을 쓰시오.**

(①)	소프트웨어가 사용자의 요구사항을 정확하게 만족하는 기능을 제공하는지 여부를 나타낸다.
신뢰성 (Reliability)	소프트웨어가 요구된 기능을 정확하고 일관되게 오류 없이 수행할 수 있는 정도를 나타낸다.
(②)	사용자와 컴퓨터 사이에 발생하는 어떠한 행위에 대하여 사용자가 정확하게 이해하고 사용하며, 향후 다시 사용하고 싶은 정도를 나타낸다.
효율성 (Efficiency)	사용자가 요구하는 기능을 할당된 시간 동안 한정된 자원으로 얼마나 빨리 처리할 수 있는지 정도를 나타낸다.
(③)	환경의 변화 또는 새로운 요구사항이 발생했을 때 소프트웨어를 개선하거나 확장할 수 있는 정도를 나타낸다.
이식성 (Portability)	소프트웨어가 다른 환경에서도 얼마나 쉽게 적용할 수 있는지 정도를 나타낸다.

답

- ① : 기능성(Functionality)
- ② : 사용성(Usability)
- ③ : 유지 보수성(Maintainability)

문제 11 **다음은 UI 설계서의 작성 순서이다. 괄호에 들어갈 알맞은 용어를 쓰시오.**



답

- ① : 시스템 구조
- ② : 사이트 맵

[병행학습]

UI 설계서 작성 순서

1. **UI 설계서 표지** : 다른 문서와 혼동되지 않도록 프로젝트명 또는 시스템명을 포함시켜 작성
2. **UI 설계서 개정 이력** : UI 설계서가 수정될 때마다 어떤 부분이 어떻게 수정되었는지를 정리해 놓은 문서
3. **UI 요구사항 정의서** : 사용자의 요구사항을 확인하고 정리한 문서로, 사용자 요구사항의 UI 적용 여부를 요구사항별로 표시

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

4. **시스템 구조** : UI 요구사항과 UI 프로토타입에 기초하여 전체 시스템의 구조를 설계한 것으로 사용자의 요구사항이 어떻게 시스템에 적용되는지 알 수 있음
5. **사이트 맵(Site Map)** : 사이트 맵은 시스템 구조를 바탕으로 사이트에 표시할 콘텐츠를 한 눈에 알아 볼 수 있도록 메뉴별로 구분하여 설계한 것
6. **프로세스(Process) 정의서** : 사용자 관점에서 사용자가 요구하는 프로세스들을 작업 진행 순서에 맞춰 정리한 것으로 UI의 전체적인 흐름을 파악할 수 있음
7. **화면 설계** : UI 프로토타입과 UI 프로세스를 참고하여 필요한 화면을 페이지별로 설계한 것

문제 12 **UI 설계와 관련한 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오.**

- ( )은 사용자가 시스템을 통해 원하는 목표를 얼마나 효과적으로 달성할 수 있는가에 대한 척도이다.
- UI의 주된 목적은 ( )이 뛰어난 UI를 제작하는 것이다.
- ( ) 평가는 사용자 측면에서 복잡한 시스템을 얼마나 편리하게 사용할 수 있는지를 평가하는 것으로, 시스템의 문제점을 찾아내고 개선 방향을 제시하기 위한 조사 과정이다.
- UI의 구조, 기능, 가치 등에 대해 사용자가 생각하는 사용자 모형과 시스템 설계자가 만들려고 하는 개발자 모형 간의 차이를 최소화해야 ( )이 뛰어난 UI를 설계할 수 있다.

답 : 유용성(Usability)

문제 13 **다음 설명에 해당하는 용어를 영문(Fullname 또는 약어)으로 쓰시오.**

- 사람이 시스템을 보다 편리하고 안전하게 사용할 수 있도록 연구하고 개발하는 학문으로, 최종 목표는 시스템을 사용하는데 있어 최적의 사용자 경험(UX)을 만드는 것이다.
- 원래 사람과 컴퓨터의 상호작용을 연구해서 사람이 컴퓨터를 편리하게 사용하도록 만드는 학문이었으나, 대상이 컴퓨터뿐만 아니라 서비스, 디지털 콘텐츠 등으로, 사람도 개인뿐만 아니라 사회나 집단으로 확대되었다.

답 : HCI(Human Computer Interaction/Interface)

문제 14 **다음이 설명하고 있는 용어를 쓰시오.**

- 제품이나 작업환경을 사용자의 감성에 알맞도록 설계 및 제작하는 기술로, 인문사회과학, 공학, 의학 등 여러 분야의 학문이 공존하는 종합과학이다.
- 인간의 삶을 편리하고 안전하며 쾌적하게 만들기 위해 생체계측 기술, 감각계측 기술, 센서, 인공지능, 생체제어 기술을 활용하여 인간의 감성을 구체적으로 제품 설계에 활용한다.

답 : 감성공학

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 15 **UI와 관련한 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 영문(Fullname 또는 약어)으로 쓰시오.**

- ( )는 사용자가 시스템이나 서비스를 이용하면서 느끼고 생각하게 되는 총체적인 경험으로, 단순히 기능이나 절차상의 만족뿐만 아니라 사용자가 참여, 사용, 관찰하고, 상호 교감을 통해서 알 수 있는 가치 있는 경험을 말한다.
- ( )는 기술을 효용성 측면에서만 보는 것이 아니라 사용자의 삶의 질을 향상시키는 하나의 방향으로 보는 새로운 개념이다.
- UI가 사용성, 접근성, 편의성을 중시한다면 ( )는 이러한 UI를 통해 사용자가 느끼는 만족이나 감정을 중시한다.
- ( )의 특징은 다음과 같다.

주관성(Subjectivity)	사람들의 개인적, 신체적, 인지적 특성에 따라 다르므로 주관적이다.
정황성(Contextuality)	경험이 일어나는 상황 또는 주변 환경에 영향을 받는다.
총체성(Holistic)	개인이 느끼는 총체적인 심리적, 감성적인 결과이다.

답 : UX(User Experience)

문제 16 **다음은 소프트웨어 아키텍처(Software Architecture)에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오.**

- 소프트웨어 아키텍처는 소프트웨어의 골격이 되는 기본 구조이자, 소프트웨어를 구성하는 요소들 간의 관계를 표현하는 시스템의 구조 또는 구조체이다.
- 소프트웨어 아키텍처는 애플리케이션의 분할 방법과 분할된 모듈에 할당될 기능, 모듈 간의 인터페이스 등을 결정한다.
- 소프트웨어 아키텍처 설계의 기본 원리는 다음과 같다.

모듈화 (Modularity)	소프트웨어의 성능을 향상시키거나 시스템의 수정 및 재사용, 유지 관리 등이 용이하도록 시스템의 기능들을 모듈 단위로 나누는 것이다.
(①)	문제의 전체적이고 포괄적인 개념을 설계한 후 차례로 세분화하여 구체화시켜 나가는 것이다.
단계적 분해 (Stepwise Refinement)	Niklaus Wirth에 의해 제안된 하향식 설계 전략으로, 문제를 상위의 중요 개념으로부터 하위의 개념으로 구체화시키는 분할 기법이다.
(②)	한 모듈 내부에 포함된 절차와 자료들의 정보가 감추어져 다른 모듈이 접근하거나 변경하지 못하도록 하는 기법이다.

답

- ① : 추상화(Abstraction)
- ② : 정보 은닉(Information Hiding)

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 17 **소프트웨어 아키텍처의 품질 속성에 대한 다음 설명에서 괄호에 들어갈 알맞은 품질 속성을 쓰시오.**

- 소프트웨어 아키텍처의 품질 속성은 소프트웨어 아키텍처가 이해 관계자들이 요구하는 수준의 품질을 유지 및 보장할 수 있게 설계되었는지를 확인하기 위해 품질 평가 요소들을 구체화시켜 놓은 것이다.
- 아키텍처 측면에서의 품질 속성은 다음과 같다.

품질 속성	내용
( )	전체 시스템과 시스템을 이루는 구성 요소들 간 일관성을 유지하는 것
정확성, 완결성	요구사항과 요구사항을 구현하기 위해 발생하는 제약사항들을 모두 충족시키는 것
구축 가능성	모듈 단위로 구분된 시스템을 적절하게 분배하여 유연하게 일정을 변경할 수 있도록 하는 것
기타 속성	변경성, 시험성, 적응성, 일치성, 대체성 등

답 : 개념적 무결성

[병행학습]

소프트웨어 아키텍처의 시스템 측면 속성

품질 속성	내용
성능	사용자의 요청과 같은 이벤트가 발생했을 때, 이를 적절하고 빠르게 처리하는 것
보안	허용되지 않은 접근을 막고, 허용된 접근에는 적절한 서비스를 제공하는 것
가용성	장애 없이 정상적으로 서비스를 제공하는 것
기능성	사용자가 요구한 기능을 만족스럽게 구현하는 것
사용성	사용자가 소프트웨어를 사용하는데 헤매지 않도록 명확하고 편리하게 구현하는 것
변경 용이성	소프트웨어가 처음 설계 목표와 다른 하드웨어나 플랫폼에서 동작할 수 있도록 구현하는 것
확장성	시스템의 용량, 처리능력 등을 확장시켰을 때 이를 효과적으로 활용할 수 있도록 구현하는 것
기타 속성	테스트 용이성, 배치성, 안정성 등

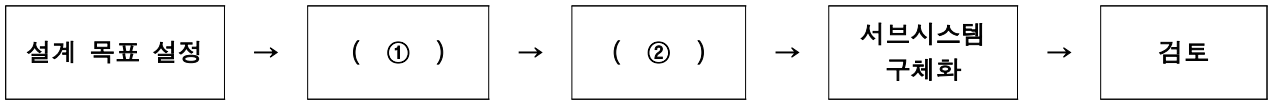
소프트웨어 아키텍처의 비즈니스 측면 속성

품질 속성	내용
시장 적시성	정해진 시간에 맞춰 프로그램을 출시하는 것
비용과 혜택	• 개발 비용을 더 투자하여 유연성이 높은 아키텍처를 만들 것인지를 결정하는 것 • 유연성이 떨어지는 경우 유지보수에 많은 비용이 소모될 수 있다는 것을 고려해야 함
예상 시스템 수명	• 시스템을 얼마나 오랫동안 사용할 것인지를 고려하는 것 • 수명이 길어야 한다면 시스템 품질의 '변경 용이성', '확장성'을 중요하게 고려해야 함
기타 속성	목표 시장, 공개 일정, 기존 시스템과의 통합 등

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 18 다음은 소프트웨어 아키텍처의 설계 과정이다. 괄호(①, ②)에 들어갈 알맞은 설계 과정을 쓰시오.



답

- ① : 시스템 타입 결정
- ② : 아키텍처 패턴 적용

[병행학습]

소프트웨어 아키텍처의 설계 과정

1. **설계 목표 설정** : 시스템의 개발 방향을 명확히 하기 위해 설계에 영향을 주는 비즈니스 목표, 우선순위 등의 요구사항을 분석하여 전체 시스템의 설계 목표를 설정
2. **시스템 타입 결정** : 시스템과 서브시스템의 타입을 결정하고, 설계 목표와 함께 고려하여 아키텍처 패턴을 선택
3. **아키텍처 패턴 적용** : 아키텍처 패턴을 참조하여 시스템의 표준 아키텍처를 설계
4. **서브시스템 구체화** : 서브시스템의 기능 및 서브시스템 간의 상호작용을 위한 동작과 인터페이스를 정의
5. **검토** : 아키텍처가 설계 목표에 부합하는지, 요구사항이 잘 반영되었는지, 설계의 기본 원리를 만족하는지 등을 검토

문제 19 소프트웨어 설계와 관련된 다음 설명에 해당하는 용어를 한글 또는 영문(Fullname 또는 약어)으로 쓰시오.

- 아키텍처를 설계할 때 참조할 수 있는 전형적인 해결 방식 또는 예제를 의미한다.
- 소프트웨어 시스템의 구조를 구성하기 위한 기본적인 윤곽을 제시한다.
- 서브시스템들과 그 역할이 정의되어 있으며, 서브시스템 사이의 관계와 여러 규칙·지침 등이 포함되어 있다.

답 : 아키텍처 패턴(Architecture Pattern)

문제 20 아키텍처 패턴에 대한 다음 설명에 해당하는 패턴을 한글 또는 영문(Fullname 또는 약어)으로 쓰시오.

- 시스템을 계층(Layer)으로 구분하여 구성하는 고전적인 방법 중의 하나이다.
- 각각의 서브시스템들이 계층 구조를 이루며, 상위 계층은 하위 계층에 대한 서비스 제공자가 되고, 하위 계층은 상위 계층의 클라이언트가 되는 패턴이다.
- 대표적인 모델로 OSI 참조 모델이 있다.

답 : 레이어 패턴(Layers Pattern)

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 21 **아키텍처 패턴에 대한 다음 설명에 해당하는 패턴을 한글 또는 영문(Fullname 또는 약어)으로 쓰시오.**

- 하나의 서버 컴포넌트와 다수의 클라이언트 컴포넌트로 구성되는 패턴이다.
- 사용자가 클라이언트를 통해 서버에 요청하고 클라이언트가 응답을 받아 사용자에게 제공하는 방식으로 서비스를 제공한다.

답 : 클라이언트-서버 패턴(Client-Server Pattern)

문제 22 **아키텍처 패턴에 대한 다음 설명에 해당하는 패턴을 한글 또는 영문(Fullname 또는 약어)으로 쓰시오.**

- 데이터 스트림 절차의 각 단계를 필터 컴포넌트로 캡슐화하여 파이프를 통해 데이터를 전송하는 패턴이다.
- 컴포넌트는 재사용성이 좋고, 추가가 쉬워 확장이 용이하며, 재배치를 통해 다양한 처리 루틴을 구축하는 것이 가능하다.
- 주로 데이터 변환, 버퍼링, 동기화 등에 사용된다.

답 : 파이프-필터 패턴(Pipe-Filter Pattern)

문제 23 **아키텍처 패턴에 대한 다음 설명에 해당하는 패턴을 한글 또는 영문(Fullname 또는 약어)으로 쓰시오.**

- 서브시스템을 3개의 부분으로 구조화하는 패턴으로, 각 부분은 별도의 컴포넌트로 분리되어 있으므로 서로 영향을 받지 않고 개발 작업을 수행할 수 있다.
- 한 개의 모델에 대해 여러 개의 뷰를 필요로 하는 대화형 애플리케이션에 적합하다.
- 각 서브시스템의 역할은 다음과 같다.
  - 모델(Model) : 서브시스템의 핵심 기능과 데이터를 보관한다.
  - 뷰(View) : 사용자에게 정보를 표시한다.
  - 컨트롤러(Controller) : 사용자로부터 받은 입력을 처리한다.

답 : 모델-뷰-컨트롤러 패턴(Model-View-Controller Pattern) 또는 MVC 패턴(MVC Pattern)

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 24 **다음은 소프트웨어 아키텍처 패턴의 종류별 특징에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 패턴을 한글 또는 영문(Fullname 또는 약어)으로 쓰시오.**

마스터-슬레이브 패턴 (Master-Slave Pattern)	• 마스터 컴포넌트에서 슬레이브 컴포넌트로 작업을 분할한 후 슬레이브 컴포넌트에서 처리된 결과물을 다시 돌려받는 방식으로 작업을 수행하는 패턴이다. • 장애 허용 시스템과 병렬 컴퓨팅 시스템에서 주로 활용된다.
브로커 패턴 (Broker Pattern)	• 사용자가 원하는 서비스와 특성을 브로커 컴포넌트에 요청하면 브로커 컴포넌트가 요청에 맞는 컴포넌트와 사용자를 연결해주는 패턴이다. • 분산 환경 시스템에서 주로 활용된다.
피어-투-피어 패턴 (Peer-To-Peer Pattern)	피어(Peer)를 하나의 컴포넌트로 간주하며, 각 피어는 서비스를 호출하는 클라이언트가 될 수도, 서비스를 제공하는 서버가 될 수도 있는 패턴이다.
(①)	소스가 특정 채널에 이벤트 메시지를 발행(Publish)하면, 해당 채널을 구독(Subscribe)한 리스너들이 메시지를 받아 이벤트를 처리하는 방식이다.
블랙보드 패턴 (Blackboard Pattern)	• 모든 컴포넌트들이 공유 데이터 저장소와 블랙보드 컴포넌트에 접근하여 검색을 통해 원하는 데이터를 찾을 수 있는 패턴이다. • 음성 인식, 차량 식별, 신호 해석 등에 주로 활용된다.
(②)	• 코드의 각 라인을 수행하는 방법을 지정하고, 기호마다 클래스를 갖도록 구성되는 패턴이다. • 특정 언어로 작성된 프로그램 코드를 해석하는 컴포넌트를 설계할 때 사용된다.

답

- ① : 이벤트-버스 패턴(Event-Bus Pattern)
- ② : 인터프리터 패턴(Interpreter Pattern)

문제 25 **코드(Code)에 대한 다음 설명에서 괄호에 들어갈 알맞은 코드 기능을 쓰시오.**

- 코드는 컴퓨터를 이용하여 자료를 처리하는 과정에서 분류·조합 및 집계를 용이하게 하고, 특정 자료의 추출을 쉽게 하기 위해서 사용하는 기호이다.
- 코드는 정보를 신속·정확·명료하게 전달할 수 있게 하며, 일정한 규칙에 따라 작성되어 정보 처리의 효율과 처리된 정보의 가치에 많은 영향을 미친다.
- 코드의 주요 기능은 다음과 같다.

( )	데이터의 간의 성격에 따라 구분이 가능하다.
분류 기능	특정 기준이나 동일한 유형에 해당하는 데이터를 그룹화 할 수 있다.
배열 기능	의미를 부여하여 나열할 수 있다.

답 : 식별 기능

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 26 다음은 코드(Code)의 종류별 특징에 대한 설명이다. 괄호(①~③)에 들어갈 알맞은 코드 종류를 쓰시오.

순차 코드 (Sequence Code)	자료의 발생 순서, 크기 순서 등 일정 기준에 따라서 최초의 자료부터 차례로 일련번호를 부여하는 방법으로, 순서 코드 또는 일련 번호 코드라고도 한다.
블록 코드 (Block Code)	코드화 대상 항목 중에서 공통성이 있는 것끼리 블록으로 구분하고, 각 블록 내에서 일련번호를 부여하는 방법으로, 구분 코드라고도 한다.
(①)	코드화 대상 항목을 0~9까지 10진 분할하고, 다시 그 각각에 대하여 10진 분할하는 방법을 필요한 만큼 반복하는 방법으로, 도서 분류식 코드라고도 한다.
(②)	코드화 대상 항목을 일정 기준에 따라 대분류, 중분류, 소분류 등으로 구분하고, 각 그룹 안에서 일련번호를 부여하는 방법이다.
연상 코드 (Mnemonic Code)	코드화 대상 항목의 명칭이나 약호와 관계있는 숫자나 문자, 기호를 이용하여 코드를 부여하는 방법이다.
(③)	코드화 대상 항목의 성질, 즉 길이, 넓이, 부피, 지름, 높이 등의 물리적 수치를 그대로 코드에 적용시키는 방법으로, 유효 숫자 코드라고도 한다.
합성 코드 (Combined Code)	필요한 기능을 하나의 코드로 수행하기 어려운 경우 2개 이상의 코드를 조합하여 만드는 방법이다.

답

- ① : 10진 코드(Decimal Code)
- ② : 그룹 분류 코드(Group Classification Code)
- ③ : 표의 숫자 코드(Significant Digit Code)

문제 27 소프트웨어 설계와 관련된 다음 설명에 해당하는 용어를 쓰시오.

- 각 모듈의 세분화된 역할이나 모듈들 간의 인터페이스와 같은 코드를 작성하는 수준의 세부적인 구현 방안을 설계할 때 참조할 수 있는 전형적인 해결 방식 또는 예제를 의미한다.
- 재사용할 수 있는 기본형 코드들이 포함되어 있다.
- 1995년 GoF가 생성 패턴 5개, 구조 패턴 7개, 행위 패턴 11개, 총 23개의 패턴으로 정리한 것이 지금까지도 소프트웨어 공학이나 현업에서 많이 사용되고 있다.

답 : 디자인 패턴(Design Pattern)

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 28 **다음은 GoF의 디자인 패턴 중 생성 패턴(Creational Pattern)에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 디자인 패턴을 쓰시오.**

- 생성 패턴은 객체의 생성과 관련된 패턴으로 총 5개의 패턴이 있다.
- 생성 패턴은 객체의 생성과 참조 과정을 캡슐화 하여 객체가 생성되거나 변경되어도 프로그램의 구조에 영향을 크게 받지 않도록 하여 프로그램에 유연성을 더해준다.
- 생성 패턴의 종류는 다음과 같다.

(①)	• 구체적인 클래스에 의존하지 않고, 인터페이스를 통해 서로 연관·의존하는 객체들의 그룹으로 생성하여 추상적으로 표현한다. • 연관된 서브 클래스를 묶어 한 번에 교체하는 것이 가능하다.
빌더(Builder)	• 작게 분리된 인스턴스를 건축 하듯이 조합하여 객체를 생성한다. • 객체의 생성 과정과 표현 방법을 분리하고 있어, 동일한 객체 생성에서도 서로 다른 결과를 만들어 낼 수 있다.
팩토리 메소드 (Factory Method)	• 객체 생성을 서브 클래스에서 처리하도록 분리하여 캡슐화한 패턴이다. • 상위 클래스에서 인터페이스만 정의하고 실제 생성은 서브 클래스가 담당한다.
프로토타입 (Prototype)	• 원본 객체를 복제하는 방법으로 객체를 생성하는 패턴이다. • 일반적인 방법으로 객체를 생성하면 비용이 큰 경우 주로 이용한다.
(②)	• 하나의 객체를 생성하면 생성된 객체를 어디서든 참조할 수 있지만, 여러 프로세스가 동시에 참조할 수는 없다. • 클래스 내에서 인스턴스가 하나뿐임을 보장하며, 불필요한 메모리 낭비를 최소화 할 수 있다.

답

- ① : 추상 팩토리(Abstract Factory)
- ② : 싱글톤(Singleton)

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 29 **다음은 GoF의 디자인 패턴 중 구조 패턴(Structural Pattern)에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 디자인 패턴을 쓰시오.**

- 구조 패턴은 클래스나 객체들을 조합하여 더 큰 구조로 만들 수 있게 해주는 패턴으로 총 7개의 패턴이 있다.
- 구조 패턴은 구조가 복잡한 시스템을 개발하기 쉽게 도와준다.
- 구조 패턴의 종류는 다음과 같다.

(①)	• 호환성이 없는 클래스들의 인터페이스를 다른 클래스가 이용할 수 있도록 변환해주는 패턴이다. • 기존의 클래스를 이용하고 싶지만 인터페이스가 일치하지 않을 때 이용한다.
브리지 (Bridge)	• 구현부에서 추상층을 분리하여, 서로가 독립적으로 확장할 수 있도록 구성한 패턴이다. • 기능과 구현을 두 개의 별도 클래스로 구현한다.
컴포지트 (Composite)	• 여러 객체를 가진 복합 객체와 단일 객체를 구분 없이 다루고자 할 때 사용하는 패턴이다. • 객체들을 트리 구조로 구성하여 디렉터리 안에 디렉터리가 있듯이 복합 객체 안에 복합 객체가 포함되는 구조를 구현할 수 있다.
데코레이터 (Decorator)	• 객체 간의 결합으로 능동적으로 기능들을 확장할 수 있는 패턴이다. • 임의의 객체에 부가적인 기능을 추가하기 위해 다른 객체들을 덧붙이는 방식으로 구현한다.
(②)	• 복잡한 서브클래스들을 피해 더 상위에 인터페이스를 구성함으로써 서브클래스들의 기능을 간편하게 사용할 수 있도록 하는 패턴이다. • 서브클래스들 사이의 통합 인터페이스를 제공하는 wrapper 객체가 필요하다.
플라이웨이트 (Flyweight)	• 인스턴스가 필요할 때마다 매번 생성하는 것이 아니고 가능한 공유해서 사용함으로써 메모리를 절약하는 패턴이다. • 다수의 유사 객체를 생성하거나 조작할 때 유용하게 사용할 수 있다.
프록시 (Proxy)	• 접근이 어려운 객체와 여기에 연결하려는 객체 사이에서 인터페이스 역할을 수행하는 패턴이다. • 네트워크 연결, 메모리의 대용량 객체로의 접근 등에 주로 이용한다.

답

- ① : 어댑터(Adapter)
- ② : 퍼싸드(Facade)

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 30 **다음은 GoF의 디자인 패턴 중 행위 패턴(Behavioral Pattern)에 대한 설명이다. 괄호(①~③)에 들어갈 알맞은 디자인 패턴을 한글 또는 영문(Fullname 또는 약어)으로 쓰시오.**

- 행위 패턴은 클래스나 객체들이 서로 상호작용하는 방법이나 책임 분배 방법을 정의하는 패턴으로 총 11개의 패턴이 있다.
- 행위 패턴은 하나의 객체로 수행할 수 없는 작업을 여러 객체로 분배하면서 결합도를 최소화 할 수 있도록 도와준다.
- 행위 패턴의 종류는 다음과 같다.

(①)	• 요청을 처리할 수 있는 객체가 둘 이상 존재하여 한 객체가 처리하지 못하면 다음 객체로 넘어가는 형태의 패턴이다. • 요청을 처리할 수 있는 각 객체들이 고리(chain)로 묶여 있어 요청이 해결될 때까지 고리를 따라 책임이 넘어간다.
커맨드 (Command)	• 요청을 객체의 형태로 캡슐화하여 재이용하거나 취소할 수 있도록 요청에 필요한 정보를 저장하거나 로그에 남기는 패턴이다. • 요청에 사용되는 각종 명령어들을 추상 클래스와 구체 클래스로 분리하여 단순화한다.
인터프리터 (Interpreter)	• 언어에 문법 표현을 정의하는 패턴이다. • SQL이나 통신 프로토콜과 같은 것을 개발할 때 사용한다.
반복자 (Iterator)	• 자료 구조와 같이 접근이 잦은 객체에 대해 동일한 인터페이스를 사용하도록 하는 패턴이다. • 내부 표현 방법의 노출 없이 순차적인 접근이 가능하다.
중재자 (Mediator)	• 수많은 객체들 간의 복잡한 상호작용(interface)을 캡슐화하여 객체로 정의하는 패턴이다. • 객체 사이의 의존성을 줄여 결합도를 감소시킬 수 있다.
메멘토 (Memento)	• 특정 시점에서의 객체 내부 상태를 객체화함으로써 이후 요청에 따라 객체를 해당 시점의 상태로 돌릴 수 있는 기능을 제공하는 패턴이다. • [Ctrl]+[Z]와 같은 되돌리기 기능을 개발할 때 주로 이용한다.
(②)	• 한 객체의 상태가 변화하면 객체에 상속되어 있는 다른 객체들에게 변화된 상태를 전달하는 패턴이다. • 주로 분산된 시스템 간에 이벤트를 생성·발행하고(publish), 이를 수신해야 할 때(subscribe) 이용한다.
상태(State)	• 객체의 상태에 따라 동일한 동작을 다르게 처리해야 할 때 사용하는 패턴이다. • 객체 상태를 캡슐화하고 이를 참조하는 방식으로 처리한다.
전략 (Strategy)	• 동일한 계열의 알고리즘들을 개별적으로 캡슐화하여 상호교환 할 수 있게 정의하는 패턴이다. • 클라이언트는 독립적으로 원하는 알고리즘을 선택하여 사용할 수 있으며, 클라이언트에 영향 없이 알고리즘의 변경이 가능하다.
(③)	• 상위 클래스에서 골격을 정의하고, 하위 클래스에서 세부 처리를 구체화하는 구조의 패턴이다. • 유사한 서브 클래스를 묶어 공통된 내용을 상위 클래스에서 정의함으로써 코드의 양을 줄이고 유지보수를 용이하게 해준다.
방문자 (Visitor)	• 각 클래스들의 데이터 구조에서 처리 기능을 분리하여 별도의 클래스로 구성하는 패턴이다. • 분리된 처리 기능은 각 클래스를 방문(visit)하여 수행한다.

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

답

- ① : 책임 연쇄(Chain of Responsibility)
- ② : 옵서버(Observer)
- ③ : 템플릿 메소드(Template Method)

문제 31 **시스템 인터페이스는 독립적으로 떨어져 있는 시스템들끼리 서로 연동하여 상호 작용하기 위한 접속 방법이나 규칙을 의미한다. 개발을 목표로 하는 시스템과 외부 시스템을 연동하는데 필요한 시스템 인터페이스에 대한 요구사항을 기술한 문서가 무엇인지 쓰시오.**

답 : 시스템 인터페이스 요구사항 명세서

문제 32 **인터페이스에 대한 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오.**

- ( )은 인터페이스의 설계 및 구현 전에 사용자들의 요구사항이 요구사항 명세서에 정확하고 완전하게 기술되었는지 검토하고 개발 범위의 기준인 베이스라인을 설정하는 것이다.
- 인터페이스의 설계 및 구현 중에 요구사항 명세서의 오류가 발견되어 이를 수정할 경우 많은 비용이 소요되므로 프로젝트에서 ( )은 매우 중요하다.

답 : 요구사항 검증(Requirements Verification)

문제 33 **다음은 인터페이스 요구사항 검증 방법에 대한 설명이다. 괄호에 들어갈 알맞은 검증 방법을 쓰시오.**

( )	요구사항 명세서의 오류 확인 및 표준 준수 여부 등의 결함 여부를 검토 담당자들이 수작업으로 분석하는 방법으로, 동료검토, 워크스루, 인스펙션(Inspection) 등이 있다.
프로토타이핑 (Prototyping)	사용자의 요구사항을 정확히 파악하기 위해 실제 개발될 소프트웨어에 대한 견본품(Prototype)을 만들어 최종 결과물을 예측한다.
테스트 설계	요구사항은 테스트할 수 있도록 작성되어야 하며, 이를 위해 테스트 케이스(Test Case)를 생성하여 이후에 요구사항이 현실적으로 테스트 가능한지를 검토한다.
CASE 도구 활용	일관성 분석(Consistency Analysis)을 통해 요구사항 변경사항의 추적 및 분석, 관리하고, 표준 준수 여부를 확인한다.

답 : 요구사항 검토(Requirements Review)

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 34 **다음은 인터페이스 요구사항 검증 방법 중 요구사항 검토(Requirements Review)에 대한 설명이다. 괄호에 들어갈 알맞은 검증 방법을 쓰시오.**

동료검토 (Peer Review)	요구사항 명세서 작성자가 명세서 내용을 직접 설명하고 동료들이 이를 들으면서 결함을 발견하는 형태의 검토 방법이다.
( )	검토 회의 전에 요구사항 명세서를 미리 배포하여 사전 검토한 후에 짧은 검토 회의를 통해 결함을 발견하는 형태의 검토 방법이다.
인스펙션 (Inspection)	요구사항 명세서 작성자를 제외한 다른 검토 전문가들이 요구사항 명세서를 확인하면서 결함을 발견하는 형태의 검토 방법이다.

답 : 워크스루(Walk Through)

문제 35 **다음은 인터페이스 요구사항 검증 시 검토해야 하는 주요 항목에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 항목을 쓰시오.**

(①)	사용자의 모든 요구사항이 누락되지 않고 완전하게 반영되어 있는가?
일관성(Consistency)	요구사항이 모순되거나 충돌되는 점 없이 일관성을 유지하고 있는가?
명확성(Unambiguity)	모든 참여자가 요구사항을 명확히 이해할 수 있는가?
(②)	요구사항이 '어떻게(How to)' 보다 '무엇을(What)'에 중점을 두고 있는가?
검증 가능성(Verifiability)	요구사항이 사용자의 요구를 모두 만족하고, 개발된 소프트웨어가 사용자의 요구 내용과 일치하는지를 검증할 수 있는가?
추적 가능성(Traceability)	요구사항 명세서와 설계서를 추적할 수 있는가?
변경 용이성 (Easily Changeable)	요구사항 명세서의 변경이 쉽도록 작성되었는가?

답

- ① : 완전성(Completeness)
- ② : 기능성(Functionality)

문제 36 **다음은 인터페이스의 처리 유형에 대한 설명이다. 괄호에 들어갈 알맞은 용어를 쓰시오.**

- 인터페이스 처리 유형은 송·수신 데이터를 어떤 형태로 처리할 것인지에 대한 방식을 의미한다.
- 인터페이스 처리 유형은 업무의 성격과 송·수신 데이터 전송량을 고려하여 다음과 같이 구분한다.

실시간 방식	사용자가 요청한 내용을 바로 처리해야 할 때 사용하는 방식이다.
( ) 방식	데이터를 매건 단위로 처리할 경우 비용이 많이 발생할 때 사용하는 방식이다.
배치 방식	대량의 데이터를 처리할 때 사용하는 방식이다.

답 : 지연 처리

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 37 **다음은 송·수신 데이터 식별에 대한 설명이다. 괄호에 공통으로 들어갈 용어를 쓰시오.**

- 식별 대상 데이터는 송·수신 시스템 사이에서 교환되는 데이터이다.
- 식별 대상 데이터는 규격화된 표준 형식에 따라 전송되며, 교환되는 데이터의 종류에는 인터페이스 표준 항목, 송·수신 데이터 항목, 공통 코드가 있다.
- 인터페이스 표준 항목은 송·수신 시스템을 연계하는데 표준적으로 필요한 데이터로, ( )와 거래 공통부로 나뉜다.

( )	• 시스템 간 연동 시 필요한 공통 정보이다. • 구성 정보에는 인터페이스 ID, 전송 시스템 정보, 서비스 코드 정보, 응답 결과 정보, 장애 정보 등이 있다.
거래 공통부	• 시스템들이 연동된 후 송·수신 되는 데이터를 처리할 때 필요한 정보이다. • 구성 정보에는 직원 정보, 승인자 정보, 기기 정보, 매체 정보 등이 있다.

답 : 시스템 공통부

문제 38 **다음은 시스템 연계 기술에 대한 설명이다. 괄호에 들어갈 알맞은 용어를 영문(Fullname 또는 약어)으로 쓰시오.**

- 시스템 연계 기술은 개발할 시스템과 내·외부 시스템을 연계할 때 사용되는 기술을 의미한다.
- 주요 시스템 연계 기술은 다음과 같다.

DB Link	DB에서 제공하는 DB Link 객체를 이용하는 방식이다.
( )	송신 시스템의 데이터베이스(DB)에서 데이터를 읽어 와 제공하는 애플리케이션 프로그래밍 인터페이스 프로그램이다.
연계 솔루션	EAI 서버와 송·수신 시스템에 설치되는 클라이언트(Client)를 이용하는 방식이다.
Socket	서버는 통신을 위한 소켓(Socket)을 생성하여 포트를 할당하고 클라이언트의 통신 요청 시 클라이언트와 연결하여 통신하는 네트워크 기술이다.
Web Service	웹 서비스(Web Service)에서 WSDL과 UDDI, SOAP 프로토콜을 이용하여 연계하는 서비스이다.

답 : API/Open API

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 39 **다음은 시스템 인터페이스 설계서에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오.**

• 시스템 인터페이스 설계서는 시스템의 인터페이스 현황을 확인하기 위해 시스템이 갖는 인터페이스 목록과 각 인터페이스의 상세 데이터 명세를 정의한 문서로, (①)과 (②)로 구성된다. • 시스템 인터페이스 설계서는 인터페이스 송·수신 방법과 인터페이스 송·수신 데이터 명세화 과정에서 작성한 산출물을 기반으로 작성한다. • 시스템 인터페이스 설계서를 작성한 후에는 (①)에 있는 각각의 인터페이스를 (②)의 내용과 비교하여 누락되거나 보완이 필요한 경우 내용을 수정한다. • 시스템 인터페이스 설계서는 내·외부 모듈 간 공통적으로 제공되는 기능과 각 데이터의 인터페이스를 확인하는데 사용된다.

답

• ① : 시스템 인터페이스 목록
• ② : 시스템 인터페이스 정의서

문제 40 **다음은 미들웨어(Middleware)에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 미들웨어 종류를 영문(Fullname 또는 약어)으로 쓰시오.**

• 미들웨어는 미들(Middle)과 소프트웨어(Software)의 합성어로, 운영체제와 해당 운영체제에서 실행되는 응용 프로그램 사이에서 운영체제가 제공하는 서비스 이외에 추가적인 서비스를 제공하는 소프트웨어이다.
• 미들웨어는 표준화된 인터페이스를 제공함으로써 시스템 간의 데이터 교환에 일관성을 보장한다.
• 미들웨어의 종류는 다음과 같다.

DB(DataBase)	데이터베이스 벤더에서 제공하는 클라이언트에서 원격의 데이터베이스와 연결하기 위한 미들웨어이다.
(①)	응용 프로그램의 프로시저를 사용하여 원격 프로시저를 마치 로컬 프로시저처럼 호출하는 방식의 미들웨어이다.
MOM(Message Oriented Middleware)	• 메시지 기반의 비동기형 메시지를 전달하는 방식의 미들웨어이다. • 온라인 업무보다는 이기종 분산 데이터 시스템의 데이터 동기를 위해 많이 사용된다.
(②)	• 항공기나 철도 예약 업무 등과 같은 온라인 트랜잭션 업무에서 트랜잭션을 처리 및 감시하는 미들웨어이다. • 사용자 수가 증가해도 빠른 응답 속도를 유지해야 하는 업무에 주로 사용된다.
ORB(Object Request Broker)	• 객체 지향 미들웨어로 코바(CORBA) 표준 스펙을 구현한 미들웨어이다. • TP-Monitor의 장점인 트랜잭션 처리와 모니터링 등을 추가로 구현한 제품도 있다.
WAS(Web Application Server)	정적인 콘텐츠를 처리하는 웹 서버와 달리 사용자의 요구에 따라 변하는 동적인 콘텐츠를 처리하기 위해 사용되는 미들웨어이다.

답

• ① : RPC(Remote Procedure Call)
• ② : TP-Monitor(Transaction Processing Monitor)

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.