



2020년 2회 정보처리기사 실기 시험 100% 합격전략집

- | | | | | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 1
일차 | 2
일차 | 3
일차 | 4
일차 | 5
일차 | 6
일차 | 7
일차 | 8
일차 | 9
일차 | 10
일차 |
| 11
일차 | 12
일차 | 13
일차 | 14
일차 | 15
일차 | 16
일차 | 17
일차 | 18
일차 | 19
일차 | 20
일차 |

5장 서버 프로그램 구현

핵심 070 개발 환경 구축 – 하드웨어
환경

핵심 071 개발 환경 구축 – 소프트웨어
환경

핵심 072 모듈(Module)

핵심 073 결합도(Coupling)

핵심 074 응집도(Cohesion)

핵심 075 팬인(Fan-In) / 팬아웃(Fan-
Out)

핵심 076 공통 모듈

핵심 077 재사용(Reuse)

핵심 078 DBMS 접속(Connection)

핵심 079 서버 개발

핵심 080 서버 개발 과정

핵심 081 배치 프로그램(Batch Program)



2020년 2회 정보처리기사 실기 대비용 핵심요약

5장 | 서버 프로그램 구현

[핵심 070] 개발 환경 구축 - 하드웨어 환경

- 개발 환경 구축은 응용 소프트웨어 개발을 위해 개발 프로젝트를 이해하고 소프트웨어 및 하드웨어 장비를 구축하는 것을 의미한다.
- 하드웨어 환경은 사용자와의 인터페이스 역할을 하는 클라이언트(Client) 그리고 클라이언트와 통신하여 서비스를 제공하는 서버(Server)로 구성된다.
- 서버는 사용 목적에 따라 웹 서버, 웹 애플리케이션 서버, 데이터베이스 서버, 파일 서버 등으로 나뉜다.

웹 서버 (Web Server)	클라이언트로부터 직접 요청을 받아 처리하는 서버로, 저장량의 정적 파일들을 제공
웹 애플리케이션 서버(WAS; Web Application Server)	사용자에게 동적 서비스를 제공하기 위해 웹 서버로부터 요청을 받아 데이터 가공 작업을 수행하거나, 웹 서버와 데이터베이스 서버 또는 웹 서버와 파일 서버 사이에서 인터페이스 역할을 수행하는 서버
데이터베이스 서버(DB Server)	데이터베이스와 이를 관리하는 DBMS를 운영하는 서버
파일 서버 (File Server)	데이터베이스에 저장하기에는 비효율적이거나, 서비스 제공을 목적으로 유지하는 파일들을 저장하는 서버

1. 다음은 클라이언트에게 서비스를 제공하는 서버에 대한 설명이다. 괄호(①, ②)에 들어갈 가장 적합한 서버를 쓰시오.

데이터베이스 서버	데이터베이스와 이를 관리하는 DBMS를 운영하는 서버
(①)	클라이언트로부터 직접 요청을 받아 처리하는 서버로, 저장량의 정적 파일들을 제공
파일 서버	데이터베이스에 저장하기에는 비효율적이거나, 서비스 제공을 목적으로 유지하는 파일들을 저장하는 서버
(②)	사용자에게 동적 서비스를 제공하기 위해 웹 서버로부터 요청을 받아 데이터 가공 작업을 수행하거나, 웹 서버와 데이터베이스 서버 또는 웹 서버와 파일 서버 사이에서 인터페이스 역할을 수행하는 서버

답

① :

② :

정답 1. ① 웹 서버(Web Server)

② 웹 애플리케이션 서버(WAS; Web Application Server)

[핵심 071] 개발 환경 구축 - 소프트웨어 환경

- 소프트웨어 환경은 클라이언트와 서버 운영을 위한 시스템 소프트웨어와 개발에 사용되는 개발 소프트웨어로 구성된다.
- 시스템 소프트웨어에는 운영체제(OS), 웹 서버 및 WAS 운영을 위한 서버 프로그램, DBMS 등이 있다.
- 개발 소프트웨어에는 요구사항 관리 도구, 설계/모델링 도구, 구현 도구, 빌드 도구, 테스트 도구, 형상 관리 도구 등이 있다.

요구사항 관리 도구	요구사항의 수집과 분석, 추적 등을 편리하게 도와주는 소프트웨어
설계/모델링 도구	UML(통합 모델링 언어)을 지원하며, 개발의 전 과정에서 설계 및 모델링을 도와주는 소프트웨어
구현 도구	개발 언어를 통해 애플리케이션의 실제 구현을 지원하는 소프트웨어
빌드 도구	구현 도구를 통해 작성된 소스의 빌드 및 배포, 라이브러리 관리를 지원하는 소프트웨어
테스트 도구	모듈들이 요구사항에 적합하게 구현되었는지 테스트하는 소프트웨어
형상 관리 도구	산출물들을 버전별로 관리하여 품질 향상을 지원하는 소프트웨어

1. 소프트웨어 개발에 사용되는 개발 소프트웨어 중 다음은 어떤 도구에 대한 설명인지 쓰시오.

- 산출물들을 버전별로 관리하여 품질 향상을 지원하는 소프트웨어이다.
- 종류에는 Git, CVS, Subversion, Mercurial 등이 있다.

답 :

정답 1. 형상 관리 도구

[핵심 072] 모듈(Module)

- 모듈화를 통해 분리된 시스템의 각 기능들로, 서브루틴, 서브시스템, 소프트웨어 내의 프로그램, 작업 단위 등과 같은 의미로 사용된다.
 - 모듈화 : 소프트웨어의 성능을 향상시키거나 시스템의 수정 및 재사용, 유지 관리 등이 용이하도록 시스템의 기능들을 모듈 단위로 분해하는 것으로 모듈 간 결합도의 최소화, 응집도의 최대화가 목표임
- 모듈의 기능적 독립성은 소프트웨어를 구성하는 각 모듈의 기능이 서로 독립됨을 의미하는 것으로, 모듈이 하나의 기능만을 수행하고 다른 모듈과의 과도한 상호작용을 배제함으로써 이루어진다.
- 독립성이 높은 모듈일수록 모듈을 수정하더라도 다른 모듈들에게는 거의 영향을 미치지 않으며, 오류가 발생해도 쉽게 발견하고 해결할 수 있다.
- 모듈의 독립성은 결합도(Coupling)와 응집도(Cohesion)에 의해 측정되며, 독립성을 높이려면 모듈의 결합도는 약하게, 응집도는 강하게, 모듈의 크기는 작게 만들어야 한다.

2020년 1회 기사 실기

1. 소프트웨어 패키징이란 모듈별로 생성한 실행 파일들을 묶어 배포용 설치 파일을 만드는 것을 말한다. 괄호(①, ②)에 들어갈 가장 적합한 용어를 쓰시오.

소스 코드는 향후 관리를 고려하여 모듈화하여 패키징한다. 모듈화는 모듈 간 (①)의 최소화와 모듈 내 요소들의 (②)는 최대화 하는 것이 목표이다.

답

- ① :
② :

정답 1. ① 결합도(Coupling) ② 응집도(Cohesion)

[핵심 073] 결합도(Coupling)

- 모듈 간에 상호 의존하는 정도 또는 두 모듈 사이의 연관 관계를 의미한다.
- 다양한 결합으로 모듈을 구성할 수 있으나 결합도가 약할수록 품질이 높고, 강할수록 품질이 낮다.
- 결합도가 강하면 시스템 구현 및 유지보수 작업이 어렵다.

- 결합도의 종류에는 자료 결합도, 스탬프 결합도, 제어 결합도, 외부 결합도, 공통 결합도, 내용 결합도가 있으며 결합도의 정도는 다음과 같다.

자료 결합도	스탬프 결합도	제어 결합도	외부 결합도	공통 결합도	내용 결합도
결합도 약함 ← → 결합도 강함					
자료 결합도 (Data Coupling)	<ul style="list-style-type: none"> 모듈 간의 인터페이스가 자료 요소로만 구성될 때의 결합도 어떤 모듈이 다른 모듈을 호출하면서 매개 변수나 인수로 데이터를 넘겨주고, 호출 받은 모듈은 받은 데이터에 대한 처리 결과를 다시 돌려주는 방식 				
스탬프(검인) 결합도 (Stamp Coupling)	모듈 간의 인터페이스로 배열이나 레코드 등의 자료 구조가 전달될 때의 결합도				
제어 결합도 (Control Coupling)	어떤 모듈이 다른 모듈 내부의 논리적인 흐름을 제어하기 위해 제어 신호를 이용하여 통신하거나 제어 요소(Function Code, Switch, Tag, Flag)를 전달하는 결합도				
외부 결합도 (External Coupling)	어떤 모듈에서 선언한 데이터(변수)를 외부의 다른 모듈에서 참조할 때의 결합도				
공통(공유) 결합도 (Common Coupling)	공유되는 공통 데이터 영역을 여러 모듈이 사용할 때의 결합도				
내용 결합도 (Content Coupling)	한 모듈이 다른 모듈의 내부 기능 및 그 내부 자료를 직접 참조하거나 수정할 때의 결합도				

2020년 1, 2회 기사 필기

1. 다음의 설명에서 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

- ()는 모듈 간에 상호 의존하는 정도 또는 두 모듈 사이의 연관 관계를 의미한다.
- ()가 높으면 시스템 구현 및 유지보수 작업이 어렵다.
- 모듈 간의 ()를 약하게 하면 모듈 독립성이 향상된다.

답 :

2. 다음은 두 모듈 사이의 연관 관계를 나타내는 결합도의 종류에 대한 설명이다. ①~③의 설명에 해당하는 결합도를 쓰시오.

- ① () 결합도 : 모듈 간의 인터페이스로 배열이나 레코드 등의 자료 구조가 전달될 때의 결합도
 ② () 결합도 : 한 모듈이 다른 모듈의 내부 기능 및 그 내부 자료를 직접 참조하거나 수정할 때의 결합도
 ③ () 결합도 : 모듈 간의 인터페이스가 자료 요소로만 구성될 때의 결합도

답

- ① :
 ② :
 ③ :

정답 1. 결합도(Coupling)

2. ① 스탬프(Stamp) ② 내용(Content) ③ 자료(Data)

[핵심 074] 응집도(Cohesion)

- 정보 은닉 개념을 확장한 것으로, 명령어나 호출문 등 모듈의 내부 요소들의 서로 관련되어 있는 정도, 즉 모듈이 독립적인 기능으로 정의되어 있는 정도를 의미한다.
- 다양한 기준으로 모듈을 구성할 수 있으나 응집도가 강할수록 품질이 높고, 약할수록 품질이 낮다.
- 응집도의 종류에는 기능적 응집도, 순차적 응집도, 교환(통신)적 응집도, 절차적 응집도, 시간적 응집도, 논리적 응집도, 우연적 응집도가 있으며 응집도의 정도는 다음과 같다.

기능적 응집도	순차적 응집도	교환적 응집도	절차적 응집도	시간적 응집도	논리적 응집도	우연적 응집도
------------	------------	------------	------------	------------	------------	------------

응집도 강함 ← → 응집도 약함

기능적 응집도 (Functional Cohesion)	모듈 내부의 모든 기능 요소들이 단일 문제와 연관되어 수행될 경우의 응집도
순차적 응집도 (Sequential Cohesion)	모듈 내 하나의 활동으로부터 나온 출력 데이터를 그 다음 활동의 입력 데이터로 사용할 경우의 응집도
교환(통신)적 응집도 (Communication Cohesion)	동일한 입력과 출력을 사용하여 서로 다른 기능을 수행하는 구성 요소들이 모였을 경우의 응집도

절차적 응집도 (Procedural Cohesion)	모듈이 다수의 관련 기능을 가질 때 모듈 안의 구성 요소들이 그 기능을 순차적으로 수행할 경우의 응집도
시간적 응집도 (Temporal Cohesion)	특정 시간에 처리되는 몇 개의 기능을 모아 하나의 모듈로 작성할 경우의 응집도
논리적 응집도 (Logical Cohesion)	유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소들로 하나의 모듈이 형성되는 경우의 응집도
우연적 응집도 (Coincidental Cohesion)	모듈 내부의 각 구성 요소들이 서로 관련 없는 요소로만 구성된 경우의 응집도

2020년 1, 2회 기사 필기

1. 다음 지문에 제시된 응집도를 높은 응집도에서 낮은 응집도 순으로 나열하시오. 단, 기호로 쓰시오.

- ㉠ 기능적 응집도 ㉡ 시간적 응집도 ㉢ 절차적 응집도
 ㉣ 우연적 응집도

답 : () → () → () → ()

2. 다음은 응집도의 종류에 대한 설명이다. 괄호(①~③)에 들어갈 가장 적합한 응집도를 쓰시오.

(①) 응집도	모듈 내 하나의 활동으로부터 나온 출력 데이터를 그 다음 활동의 입력 데이터로 사용할 경우의 응집도
기능적 응집도	모듈 내부의 모든 기능 요소들이 단일 문제와 연관되어 수행될 경우의 응집도
(②) 응집도	특정 시간에 처리되는 몇 개의 기능을 모아 하나의 모듈로 작성할 경우의 응집도
논리적 응집도	유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소들로 하나의 모듈이 형성되는 경우의 응집도
(③) 응집도	모듈이 다수의 관련 기능을 가질 때 모듈 안의 구성 요소들이 그 기능을 순차적으로 수행할 경우의 응집도

답

- ① :
 ② :
 ③ :

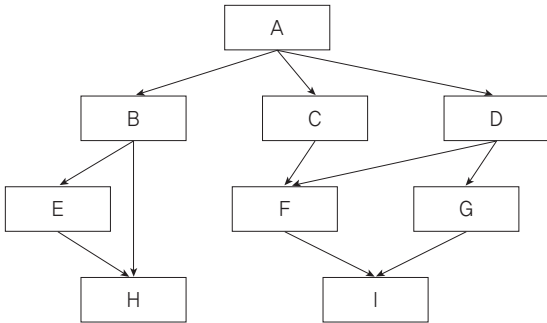
정답 1. ㉠ → ㉡ → ㉢ → ㉣

2. ① 순차적(Sequential) ② 시간적(Temporal) ③ 절차적(Procedural)

[핵심 075] 팬인(Fan-In) / 팬아웃(Fan-Out)

- 팬인은 어떤 모듈을 제어(호출)하는 모듈의 수를 나타낸다.
- 팬아웃은 어떤 모듈에 의해 제어(호출)되는 모듈의 수를 나타낸다.
- 팬인과 팬아웃을 분석하여 시스템의 복잡도를 알 수 있다.
- 팬인이 높다는 것은 재사용 측면에서 설계가 잘 되어있다고 볼 수 있으나, 단일 장애점이 발생할 수 있으므로 중점적인 관리 및 테스트가 필요하다.

예제 다음의 시스템 구조도에서 각 모듈의 팬인(Fan-In)과 팬아웃(Fan-Out)을 구하시오.



해설

- 팬인(Fan-In) : A는 0, B · C · D · E · G는 1, F · H · I는 2
- 팬아웃(Fan-Out) : H · I는 0, C · E · F · G는 1, B · D는 2, A는 3

1. (㉠)은 어떤 모듈을 제어하는 모듈의 수를 나타내고, (㉡)은 어떤 모듈에 의해 제어되는 모듈의 수를 나타낸다. 괄호 (㉠, ㉡)에 들어갈 가장 적합한 용어를 쓰시오.

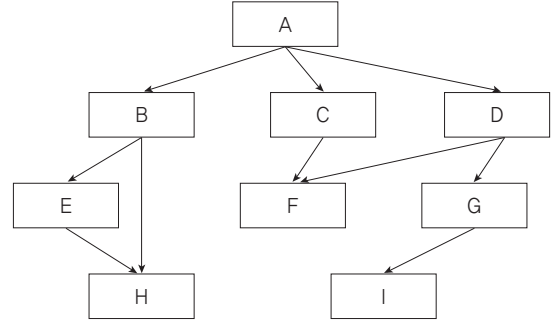
답

① :

② :

2020년 1회 기사 실기

2. 다음의 모듈 관계를 표현한 시스템 구조도를 참고하여 팬인(Fan-In)이 2 이상인 모듈을 모두 쓰시오.



답 :

해설 2. 팬인(Fan-In)은 위에서 특정 모듈로 들어오는(In) 선(Line)의 수라고 생각하면 됩니다. 그러므로 2개의 선이 들어오는 F, H 모듈이 정답이 됩니다.

정답 1. ① 팬인(Fan-In) ② 팬아웃(Fan-Out) 2. F, H

[핵심 076] 공통 모듈

- 여러 프로그램에서 공통적으로 사용할 수 있는 모듈을 의미한다.
- 자주 사용되는 계산식이나 매번 필요한 사용자 인증과 같은 기능들이 공통 모듈로 구성될 수 있다.
- 공통 모듈을 구현할 때는 다른 개발자들이 해당 기능을 명확히 이해할 수 있도록 다음의 명세 기법을 준수해야 한다.

정확성 (Correctness)	시스템 구현 시 해당 기능이 필요하다는 것을 알 수 있도록 정확히 작성
명확성 (Clarity)	해당 기능을 이해할 때 중의적으로 해석되지 않도록 명확하게 작성
완전성 (Completeness)	시스템 구현을 위해 필요한 모든 것을 기술
일관성 (Consistency)	공통 기능들 간 상호 충돌이 발생하지 않도록 작성
추적성 (Traceability)	기능에 대한 요구사항의 출처, 관련 시스템 등의 관계를 파악할 수 있도록 작성



1. 다음 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

- ()는 자주 사용되는 계산식이나 매번 필요한 사용자 인증과 같은 기능들에 주로 사용된다.
- 인터넷 쇼핑몰 사이트에서 사이트의 첫 페이지에서 '로그인'을 할 수도 있고, 상품 구매 버튼을 누른 후에 나오는 '로그인' 창에서 할 수도 있다. 두 개의 '로그인'은 발생하는 위치는 다르지만 동일한 기능을 갖고 있어 ()로 구성하기에 적합하다.

답:

2020년 1.2회 기사 필기

2. 공통 모듈에 대한 명세 기법 중 해당 기능에 대해 일관되게 이해되고 한 가지로 해석될 수 있도록 작성하는 원칙을 쓰시오.

답:

정답 1. 공통 모듈 2. 명확성(Clarity)

[핵심 077] 재사용(Reuse)

- 비용과 개발 시간을 절약하기 위해 이미 개발된 기능들을 파악하고 재구성하여 새로운 시스템 또는 기능 개발에 사용하기 적합하도록 최적화 시키는 작업이다.
- 재사용되는 대상은 외부 모듈과의 결합도는 낮고, 응집도는 높아야 한다.
- 재사용 규모에 따른 분류

함수와 객체	클래스나 메소드 단위의 소스 코드를 재사용함
컴포넌트	컴포넌트 자체에 대한 수정 없이 인터페이스를 통해 통신하는 방식으로 재사용함
애플리케이션	공통된 기능들을 제공하는 애플리케이션을 공유하는 방식으로 재사용함

1. 다음은 재사용에 대한 설명이다. 괄호(①~③)에 들어갈 가장 적합한 용어를 쓰시오.

재사용은 비용과 개발 시간을 절약하기 위해 이미 개발된 기능들을 파악하고 재구성하여 새로운 시스템 또는 기능 개발에 사용하기 적합하도록 최적화 시키는 작업으로 재사용되는 대상은 외부 모듈과의 (①)는 낮고, (②)는 높아야 한다. (①)는 모듈 간에 상호 의존하는 정도를, (②)는 내부 요소들의 서로 관련되어 있는 정도를 나타낸다.

답

- ① :
② :

정답 1. ① 결합도 ② 응집도

[핵심 078] DBMS 접속(Connection)

- 사용자가 데이터를 사용하기 위해 응용 시스템을 이용하여 DBMS에 접근하는 것을 의미한다.
- 응용 시스템은 사용자로부터 매개 변수를 전달받아 SQL을 실행하고 DBMS로부터 전달받은 결과를 사용자에게 전달하는 매개체 역할을 수행한다.
- 인터넷을 통해 구동되는 웹 응용 프로그램은 웹 응용 시스템을 통해 DBMS에 접근한다.
- DBMS 접속 기술 : DBMS에 접근하기 위해 사용하는 API 또는 API의 사용을 편리하게 도와주는 프레임워크 등을 의미

JDBC(Java DataBase Connectivity)	Java 언어로 다양한 종류의 데이터베이스에 접속하고 SQL문을 수행할 때 사용되는 표준 API
ODBC(Open DataBase Connectivity)	데이터베이스에 접근하기 위한 표준 개방형 API로, 개발 언어에 관계없이 사용할 수 있음
MyBatis	<ul style="list-style-type: none"> • JDBC 코드를 단순화하여 사용할 수 있는 SQL Mapping 기반 오픈 소스 접속 프레임워크 • SQL을 거의 그대로 사용할 수 있어 SQL 친화적인 국내 환경에 적합하여 많이 사용됨

※ 프레임워크(Framework) : 본래 '뼈대', '골조'를 의미하는 용어로, 소프트웨어에서는 특정 기능을 수행하기 위해 필요한 클래스나 인터페이스 등을 모아둔 집합체를 가리킴

- 동적 SQL(Dynamic SQL)
 - 다양한 조건에 따라 SQL 구문을 동적으로 변경하여 처리할 수 있는 SQL 처리 방식이다.
 - 동적 SQL은 사용자로부터 SQL문의 일부 또는 전부를 입력받아 실행할 수 있다.



- 동적 SQL은 정적 SQL에 비해 속도가 느리지만, 상황에 따라 다양한 조건을 첨가하는 등 유연한 개발이 가능하다.

※ 정적 SQL : SQL 코드를 변수에 담지 않고 코드 사이에 직접 기술한 SQL문

1. 다음의 설명에 가장 부합하는 DBMS 접속 기술을 쓰시오.

- 데이터베이스에 접근하기 위한 표준 개방형 API로, 개발 언어에 관계없이 사용할 수 있다.
- 프로그램 내 이 기술의 문장을 사용하여 MS-Access, DBase, DB2, Excel, Text 등 다양한 데이터베이스에 접근할 수 있다.

답 :

2. JDBC 코드를 단순화하여 사용할 수 있는 SQL Mapping 기반 오픈 소스 접속 프레임워크로 SQL을 거의 그대로 사용할 수 있어 SQL 친화적인 국내 환경에 적합하여 많이 사용되는 DBMS 접속 기술을 쓰시오.

답 :

정답 1. ODBC(Open DataBase Connectivity) 2. MyBatis

[핵심 079] 서버 개발

- 웹 애플리케이션의 로직을 구현할 서버 프로그램을 제작하여 웹 애플리케이션 서버(WAS)에 탑재하는 것을 의미한다.
- 웹 애플리케이션 서버에 구현된 서버 프로그램은 웹 서버로부터 받은 요청을 처리하여 결과를 반환하는 역할을 수행한다.
- 서버 개발에 사용되는 프로그래밍 언어에는 Java, JavaScript, Python, PHP, Ruby 등이 있다.
- 각 프로그래밍 언어에는 해당 언어로 서버 프로그램을 개발할 수 있도록 지원하는 프레임워크가 있다.

• 서버 개발 프레임워크

프레임워크	특징
spring	JAVA를 기반으로 만들어진 프레임워크로, 전자정부 표준 프레임워크의 기반 기술로 사용되고 있다.
Node.js	JavaScript를 기반으로 만들어진 프레임워크로, 비동기 입·출력 처리와 이벤트 위주의 높은 처리 성능을 갖고 있어 실시간으로 입·출력이 빈번한 애플리케이션에 적합하다.
Django	Python을 기반으로 만들어진 프레임워크로, 컴포넌트의 재사용과 플러그인화를 강조하여 신속한 개발이 가능하도록 지원한다.
Codeigniter	PHP를 기반으로 만들어진 프레임워크로, 인터페이스가 간편하며 서버 자원을 적게 사용한다.
Ruby on Rails	Ruby를 기반으로 만들어진 프레임워크로, 테스트를 위한 웹 서버를 지원하며 데이터베이스 작업을 단순화, 자동화시켜 개발 코드의 길이가 짧아 신속한 개발이 가능하다.

• 프레임워크의 특징

모듈화 (Modularity)	프레임워크는 캡슐화를 통해 모듈화를 강화하고 설계 및 구현의 변경에 따른 영향을 최소화함으로써 소프트웨어의 품질을 향상시킨다.
재사용성 (Reusability)	프레임워크는 재사용 가능한 모듈들을 제공함으로써 개발자의 생산성을 향상시킨다.
확장성 (Extensibility)	프레임워크는 다형성(Polymorphism)을 통한 인터페이스 확장이 가능하여 다양한 형태와 기능을 가진 애플리케이션 개발이 가능하다.
제어의 역흐름 (Inversion of Control)	개발자가 관리하고 통제해야 하는 객체들의 제어 권한을 프레임워크에 넘김으로써 생산성을 향상시킨다.

1. 다음은 서버 개발 프레임워크들의 특징을 설명한 것이다. 제시된 특징에 부합되도록 괄호(①, ②)에 들어갈 가장 알맞은 프레임워크를 <보기>에서 골라 쓰시오.

- (①) : JavaScript를 기반으로 만들어진 프레임워크로, 비동기 입·출력 처리와 이벤트 위주의 높은 처리 성능을 갖고 있어 실시간으로 입·출력이 빈번한 애플리케이션에 적합하다.



- (②) : 테스트를 위한 웹 서버를 지원하며 데이터베이스 작업을 단순화, 자동화시켜 개발 코드의 길이가 짧아 신속한 개발이 가능하다.

<보기>

스프링(Spring), 이클립스(Eclipse), 엑스 코드(Xcode), 메이븐(Maven), 장고(Django), 루비 온 레일즈(Ruby on Rails), 마이바티스(MyBatis), 코드이그나이터(Codeigniter), Node.js

답

- ① :
② :

2. 프레임워크(Framework)의 특성 중 제어의 역흐름(Inversion of Control)에 대해 간략히 서술하시오.

답 :

정답 1. ① Node.js ② 루비 온 레일즈(Ruby on Rails)

2. 제어의 역흐름은 개발자가 관리하고 통제해야 하는 객체들의 제어 권한을 프레임워크에 넘김으로써 생산성을 향상시키는 특성이다.

[핵심 080] 서버 개발 과정

- 서버 개발은 DTO/VO, SQL, DAO, Service, Controller를 각각 구현하는 과정을 통해 이루어진다.
- 구현 순서는 개발자가 임의로 변경할 수 있다.
- 개발하려는 서버 프로그램의 목적, 개발 언어, 규모 등의 이유로 통합되거나 세분화될 수 있다.
- 모든 과정에서 보안 약점이 발생하지 않도록 소프트웨어 개발 보안 가이드를 참고한다.

DTO (Data Transfer Object)/ VO(Value Object) 구현	<ul style="list-style-type: none"> • 데이터 교환을 위해 사용할 객체를 만드는 과정이다. • 변수 및 객체를 송·수신할 데이터의 자료형(Data Type)에 알맞게 생성한다. • 알고리즘 등의 로직은 구현하지 않고, 변수와 데이터를 저장하고 반환하는 메소드만 구현한다.
---	---

SQL 구현	<ul style="list-style-type: none"> • 데이터의 삽입, 변경, 삭제 등의 작업을 수행할 SQL문을 생성하는 과정이다. • SQL문은 소스 코드 내에 직접 입력하거나, 별도의 XML 파일로 저장하여 관리한다. • XML 파일로 SQL문을 관리하는 경우 중복되는 SQL문을 최소화할 수 있고, 유지보수가 간편해진다.
DAO (Data Access Object) 구현	데이터베이스에 접근하고, SQL을 활용하여 데이터를 실제로 조작하는 코드를 구현하는 과정이다.
Service 구현	사용자의 요청에 응답하기 위한 로직을 구현하는 과정이다.
Controller 구현	사용자의 요청에 적절한 서비스를 호출하여, 그 결과를 사용자에게 반환하는 코드를 구현하는 과정이다.

1. 서버 개발은 DTO/VO, SQL, DAO, Service, Controller를 각각 구현하는 과정을 통해 이루어지는데, 서버 개발 구현 과정 중 데이터베이스에 접근하고, SQL을 활용하여 데이터를 실제로 조작하는 코드를 구현하는 과정은 무엇인지 쓰시오.

답 :

정답 1. DAO(Data Access Object) 구현

[핵심 081] 배치 프로그램(Batch Program)

- 사용자와의 상호 작용 없이 여러 작업들을 미리 정해진 일련의 순서에 따라 일괄적으로 처리하는 것을 의미한다.
- 배치 프로그램이 자동으로 수행되는 주기에 따라 정기 배치, 이벤트성 배치, On-Demand 배치로 구분된다.
- 배치 프로그램의 필수 요소 : 대용량 데이터, 자동화, 견고성, 안정성/신뢰성, 성능
- 배치 스케줄러(Batch Scheduler) : 일괄 처리(Batch Processing) 작업이 설정된 주기에 맞춰 자동으로 수행되도록 지원해주는 도구

• 주요 배치 스케줄러

스프링 배치 (Spring Batch)	<ul style="list-style-type: none"> • Spring Source 사와 Accenture 사가 2007년 공동 개발한 오픈소스 프레임워크이다. • 로그 관리, 추적, 트랜잭션 관리, 작업 처리 통계, 작업 재시작 등의 다양한 기능을 제공한다. • 주요 구성 요소 : Job, Job Launcher, Step, Job Repository
Quartz	<ul style="list-style-type: none"> • 스프링 프레임워크로 개발되는 응용 프로그램들의 일괄 처리를 위한 다양한 기능을 제공하는 오픈 소스 라이브러리이다. • 주요 구성 요소 : Scheduler, Job, JobDetail, Trigger
Cron	리눅스의 스케줄러 도구로 crontab 명령어를 통해 작업을 예약할 수 있다.

1. 다음 설명에 가장 부합하는 용어를 쓰시오.

사용자와의 상호 작용 없이 여러 작업들을 미리 정해진 일련의 순서에 따라 일괄적으로 처리하는 것을 의미하며, 수행하는 주기에 따라 정기, 이벤트성, On-Demand로 구분할 수 있다.

답 :

2. 배치 프로그램과 관련된 소프트웨어 중 다음과 같은 특징을 갖는 소프트웨어를 쓰시오.

- 데이터베이스나 파일의 데이터를 교환하는데 필요한 컴포넌트들을 제공한다.
- 로그 관리, 추적, 트랜잭션 관리, 작업 처리 통계, 작업 재시작 등의 다양한 기능을 제공한다.
- 주요 구성 요소에는 Job, Job Launcher, Step, Job Repository가 있다.

답 :

정답 1. 배치 프로그램(Batch Program)

2. 스프링 배치(Spring Batch)