

## [프로그래밍 언어 활용>서버프로그램 구현>개발환경 구축]

[기출 예상 문제]

1. 다음은 무엇에 대한 설명인가?

애플리케이션 설계를 기반으로 개발에 필요한 환경을 구성하고, 프로그래밍 언어와 도구를 활용하여 공통 모듈, 업무프로그램과 배치 프로그램을 구현하는 능력이다.

- ① 서버프로그램 구현    ② 단위 모듈 구현
- ③ 데이터베이스 구현    ④ 인터페이스 구현

[기출 예상 문제]

2. 개발 환경 구성을 위한 도구들의 분류가 아닌 것은?

- ① 구현 도구                      ② 테스트 도구
- ③ 인터페이스 도구            ④ 형상관리 도구

[기출 예상 문제]

3. 다음 보기 중 개발 도구 분류 종류가 다른 하나는?

- ① CVS                              ② Subversion
- ③ Eclipse                        ④ Git

[기출 예상 문제]

4. 형상관리 도구 (Configuration Management Tool)에 대한 설명으로 옳은 것은?

- ① 작성한 코드에 대한 빌드 및 배포를 수행하며, 각각의 구성 요소들과 모듈들에 대한 의존성 관리를 지원하는 도구이다.
- ② 코드의 작성과 디버깅, 수정 등과 같은여러가지 작업이 가능하며 프로그램을 개발할 때 가장 많이 사용되는 도구이다.
- ③ 코드의 기능 검증과 전체의 품질을 높이기 위해 사용되는 도구이다.
- ④ 개발자들이 작성한 코드와 리소스 등 산출물에 대한 버전관리를 위해 사용된다.

[기출 예상 문제]

5. 웹 서비스를 동적 (Dynamic)으로 제공하기 위해 Resin, Tomcat 등 과 서비스에 관련된 애플리케이션이 설치되는 서버는 무엇인가?

- ① 파일(File) 서버
- ② 웹(Web) 서버
- ③ 데이터베이스(Database) 서버
- ④ 웹 애플리케이션(Application) 서버

## [프로그래밍 언어 활용>서버프로그램 구현>개발환경 구축]

[기출 예상 문제]

6. 소프트웨어의 개발을 위한 전체 과정에서 발생하는 모든 항목의 변경 사항을 관리하기 위한 활동은 무엇인가?

- ① 운영체제(OS)                      ② 형상관리
- ③ DBMS                                ④ 미들웨어

[기출 예상 문제]

7. 소프트웨어 환경 구성 이 아닌 것은?

- ① 운영체제(OS)                      ② 시스템 인터페이스
- ③ 미들웨어                            ④ DBMS

[기출 예상 문제]

8. 다음 중 미들웨어(Middleware)의 종류가 아닌 것은?

- ① Web Server                        ② WAS
- ③ JVM                                  ④ Oracle

[기출 예상 문제]

9. 형상관리의 특징이 아닌 것은?

- ① 소프트웨어 변경 시 버전별로 관리하며 소프트웨어 뿐만 아니라 형상 대상의 항목을 모두 관리한다.
- ② 형상관리는 프로젝트의 전 단계에서 수행이며, 사업 계획 단계부터 유지 보수 단계에서도 수행된다.
- ③ 프로젝트 또는 기업에서 필요로 하는 내부 리포지토리 구성을 위한 도구이다.
- ④ 형상관리를 수행하여 소프트웨어 개발 과정에서 발생하는 여러 가지 문제점 발생 시 해당 요인을 추적한다.

[기출 예상 문제]

10. 형상관리 절차의 순서로 알맞은 것은?

ㄱ. 형상 검사	ㄴ. 형상상태보고
ㄷ. 변경제어	ㄹ. 형상식별

- ① ㄹ-ㄷ-ㄴ-ㄱ                      ② ㄱ-ㄷ-ㄴ-ㄹ
- ③ ㄱ-ㄷ-ㄹ-ㄴ                      ④ ㄴ-ㄹ-ㄷ-ㄱ

## [프로그래밍 언어 활용>서버프로그램 구현>개발환경 구축]

[기출 예상 문제]

11. 개발환경 구축 순서로 알맞은 것은?

- ㄱ. 개발 환경 구성을 위한 필요 도구 설계
- ㄴ. 개발 대상에 따른 적절한 개발 언어 선정
- ㄷ. 프로젝트의 요구사항 분석
- ㄹ. 개발 언어와 하드웨어를 고려한 구현도구 구축
- ㅁ. 빌드와 테스트 도구 구축

- ① ㄹ-ㄱ-ㄷ-ㄴ-ㄱ
- ② ㄹ-ㄴ-ㄷ-ㄱ-ㅁ
- ③ ㄷ-ㄱ-ㄴ-ㄹ-ㅁ
- ④ ㄱ-ㄷ-ㄴ-ㄹ-ㅁ

[기출 예상 문제]

12. 다음 중 개발언어 선정 기준 특성 중 설명으로 옳지 않은 것은?

- ① 효율성 : 개발 대상을 효율적으로 구현이 가능한가
- ② 이식성 : 여러가지 디바이스(Device)와 운영체제(OS)에 적용가능한가
- ③ 적정성 : 다수의 시스템에서 사용중이며 많은 과거의 사례가 존재하는가
- ④ 친밀성 : 대중의 프로그래머가 사용 가능한 언어인가

[기출 예상 문제]

13. 다음은 설명하는 프레임워크의 특징은 무엇인가?

프레임워크 코드가 전체 애플리케이션의 처리 흐름을 제어하여 특정한 이벤트가 발생할 때 다형성(Polymorphism)을 통해 애플리케이션이 확장한 메소드를 호출함으로써 제어가 프레임워크로부터 애플리케이션으로 거꾸로 흐르게 한다.

- ① 모듈화(modularity)
- ② 재사용성(reusability)
- ③ 확장성(extensibility)
- ④ 제어의 역흐름(inversion of control)

## [프로그래밍 언어 활용>서버프로그램 구현>공통 모듈 구현]

[기출 예상 문제]

1. 다음 빈 칸에 들어갈 말로 가장 적절한 것은?

편의적 재사용(opportunistic reuse)이란 프로젝트를 시작할 때 재사용가능한 컴포넌트가 있는지를 찾아보고 재사용하는 것을 말한다. 편의적 재사용은 다음으로 더 세분화 할 수 있다. (ㄱ)이란 팀내에서 만든 컴포넌트를 재사용하는 것으로 어디까지나 편의상이며 계획적인 것이 아니기 때문에, 인터페이스의 조정 등에 추가적인 비용이 발생할수 있다. (ㄴ) 서드파티에서 만든 컴포넌트를 구하여 사용하는 것으로 유상인 경우, 조달비용을 자신이 직접 개발할 때 드는 비용의 20% 이하로 잡는 것이 일반적이다.

- ① (ㄱ) 계획적 재사용 (ㄴ) 외부 재사용
- ② (ㄱ) 외부 재사용 (ㄴ) 내부 재사용
- ③ (ㄱ) 내부 재사용 (ㄴ) 외부 재사용
- ④ (ㄱ) 내부 재사용 (ㄴ) 계획적 재사용

[기출 예상 문제]

2. 재사용 프로그래밍 기법이 아닌 것은?

- ① 메타 프로그래밍      ② 절차적 프로그래밍
- ③ 제네릭 프로그래밍    ④ 객체지향 프로그래밍

[기출 예상 문제]

3. 재사용의 사례 중 다음을 설명하는 것은?

- 비슷한 문제를 풀기 위한 범용적인 해법이다.
- 개념적이고, 개별 문제의 필요에 따라 수정 가능하다.
- 추상 클래스와 인터페이스는 특정의 패턴에 재사용 가능하다.

- ① 프레임워크(Framework)
- ② 디자인패턴(Design pattern)
- ③ 객체지향 프로그래밍
- ④ 소프트웨어 라이브러리(Software Library)

[이전 기출 문제]

4. 좋은 모듈이 되기 위한 응집도와 결합도에 대한 설명으로 옳은 것은?

- ① 모듈의 응집도와 결합도 모두가 높아야 한다.
- ② 모듈의 응집도는 높아야 하고 결합도는 낮아야 한다.
- ③ 모듈의 응집도는 낮아야 하고 결합도는 높아야 한다.
- ④ 모듈의 응집도와 결합도 모두가 낮아야 한다.

## [프로그래밍 언어 활용>서버프로그램 구현>공통 모듈 구현]

[기출 예상 문제]

5. 모듈화의 원리의 설명으로 옳지 않은 것은?

- ① 분할과 지배 (Divide & Conquer): 복잡한 문제를 분해, 모듈 단위로 문제 해결한다.
- ② 정보 은폐(information hiding): 어렵거나 변경 가능성이 있는 모듈을 타모듈로부터 은폐한다.
- ③ 모듈의 독립성(Module Independence): 높은 결합도와 낮은 응집도를 가진다.
- ④ 자료 추상화(Data Abstraction): 각 모듈 자료구조를 액세스하고 수정하는 함수내에 자료 구조의 표현내역을 은폐한다.

[기출 예상 문제]

6. 모듈화의 종류의 구현 측면이 아닌 것은?

- ① 서비스(Service)      ② 매크로(Macro)
- ③ 함수(Function)      ④ 인라인(Inline)

[이전 기출 문제]

7. 결합도(coupling)가 강한 순서대로 옳게 나열된 것은?

- ① 내용 결합도>공통 결합도>제어 결합도>스탬프 결합도>데이터 결합도
- ② 공통 결합도>내용 결합도>제어 결합도>데이터 결합도>스탬프 결합도
- ③ 데이터 결합도>내용 결합도>제어 결합도>공통 결합도>스탬프 결합도
- ④ 공통 결합도>내용 결합도>제어 결합도>스탬프 결합도>데이터 결합도

[이전 기출 문제]

8. 두 모듈이 동일한 자료구조를 조회하는 경우의 결합성이며 자료구조의 어떠한 변화, 즉 포맷이나 구조의 변화는 그것을 조회하는 모든 모듈 및 변화되는 필드를 실제로 조회하지 않는 모듈에까지도 영향을 미치게 되는 결합성은?

- ① data coupling      ② stamp coupling
- ③ control coupling      ④ content coupling

[이전 기출 문제]

9. 한 모듈과 다른 모듈간의 상호 의존도 또는 두 모듈사이의 연관 관계를 의미하는 것은?

- ① 신뢰도    ② 충실도    ③ 응집도    ④ 결합도

## [프로그래밍 언어 활용>서버프로그램 구현>공통 모듈 구현]

[이전 기출 문제]

10. 모듈을 이루고 있는 각 요소들이 공통의 목적을 달성하기 위하여 얼마나 관련이 있는가를 나타내는 것을 무엇이라고 하는가?

- ① 결합도(coupling)    ② 응집도(cohesion)
- ③ 구조도(structure)    ④ 일치도(unity)

[이전 기출 문제]

11. 응집력이 강한 것부터 약한 순서로 옳게 나열된 것은?

- ① sequential → functional → procedural → coincidental → logical
- ② procedural → coincidental → functional → sequential → logical
- ③ functional → sequential → procedural → logical → coincidental
- ④ logical → coincidental → functional → sequential → procedural

[이전 기출 문제]

12. 응집도의 단계에서 모듈진행 요소들의 서로 관계되어지고 순서대로 진행되는 단계는 무엇인가?

- ① 통신적 단계                      ② 시간적 단계
- ③ 순차적 단계                      ④ 절차적 단계

## [프로그래밍 언어 활용>서버프로그램 구현>서버 프로그램 구현]

[기출 예상 문제]

1. 소프트웨어 취약점 중에서 메모리를 다루는 데에 오류가 발생하여 잘못된 동작을 하는 프로그램 취약점은 무엇인가?

- ① 허상 포인터(Dangling pointer)
- ② 심볼릭 링크 경쟁
- ③ 버퍼 오버플로
- ④ 코드 인젝션

[기출 예상 문제]

2. SW개발보안가이드 유형(시큐어코딩)에 대한 설명으로 옳지 않은 것은?

- ① 입력 데이터 검증 및 표현: 타입 변환 오류, 자원 (메모리 등)의 부적절한 반환 등과 같이 개발자가 범할 수 있는 코딩 오류로 인해 유발되는 보안 약점
- ② 캡슐화: 중요한 데이터 또는 기능성을 불충분하게 캡슐화하였을 때 인가되지 않는 사용자에게 데이터 누출이 가능해지는 보안 약점
- ③ 에러 처리: 에러 처리하지 않거나, 불충분하게 처리하여 에러 정보에 중요 정보(시스템 등)가 포함될 때 발생할 수 있는 보안 약점
- ④ 보안 기능 인증, 접근 제어, 기밀성, 암호화, 권한 관리 등을 적절하지 않게 구현시 발생할 수 있는 보안 약점

[기출 예상 문제]

3. 소프트웨어 취약점 분류에서 입력 확인 오류가 아닌 것은?

- ① 디렉토리 접근공격
- ② HTTP 헤더 인젝션
- ③ HTTP 응답 스플리팅
- ④ FTP 바운스 공격

[기출 예상 문제]

4. 다음을 설명하는 것은 무엇인가?

- 응용 프로그램에서 사용할 수 있도록, 운영 체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스를 말한다.  
- 주로 파일 제어, 창 제어, 화상 처리, 문자 제어 등을 위한 인터페이스를 제공한다.

- ① Web Server                      ② Kernel
- ③ Shell                              ④ API

[기출 예상 문제]

5. 다음 중 단일 유닉스 규격(SUS, Single UNIX Specification)에 등록된 운영 체제 목록이 아닌 것은?

- ① Windows API                      ② HP/UX
- ③ 맥 OS X                              ④ 솔라리스

## [프로그래밍 언어 활용>서버프로그램 구현>서버 프로그램 구현]

[기출 예상 문제]

6. 다음을 설명하는 것은 무엇인가?

- 마이크로소프트 윈도우 운영 체제들이 사용하는 API 이다.
- C/C++ 프로그램에서 직접 운영 체제와 상호작용 할 수 있도록 만들어졌으며, 그보다 더 낮은 수준의 제어는 Ntdll.dll을 사용한 낮은 수준의 DLL로 가능하다.

- ① SUS
- ② Window API
- ③ Web API
- ④ Java API



## [프로그래밍 언어 활용>서버프로그램 구현>배치 프로그램 구현]

[기출 예상 문제]

1. 다음을 설명하는 것은 무엇인가?

사용자와의 상호작용 없이 일련의 작업들을 작업 단위로 묶어 정기적으로 반복 수행하거나 정해진 규칙에 따라 일괄 처리하는 것이다.

- ① 절차 지향 프로그램    ② 제네릭 프로그램
- ③ 객체 지향 프로그램    ④ 배치 프로그램

[기출 예상 문제]

2. 배치 프로그램의 필수 요소에 대한 설명이다. 틀린 것은?

- ① 주어진 시간 내에 처리를 완료할 수 있어야하고, 동시에 동작하고 있는 다른 애플리케이션을 방해하지 말아야 한다.
- ② 대용량의 데이터를 처리할 수 있어야 한다.
- ③ 유효하지 않은 데이터의 경우에는 처리하지 않는다.
- ④ 심각한 오류 상황 외에는 사용자의 개입 없이 동작해야 한다.

[기출 예상 문제]

3. 스프링 배치(Spring Batch)의 핵심 기능의 설명으로 옳지 않은 것은?

- ① 선언적 생략과 처리 실패 후 재시도 설정을 제공한다.
- ② Spring Framework에 플러그인(Plug-in)되어 수행하는 Job과 실행 스케줄을 정의하는 Trigger를 분리하여 유연성을 제공한다.
- ③ 스프링의 DI, AOP 및 다양한 엔터프라이즈 지원 기능을 사용한다.
- ④ 배치 처리(데이터베이스나 파일로부터 데이터를 읽거나 쓰는 등)시 공통적으로 필요한 컴포넌트를 제공한다.

**[정답] [프로그래밍 언어 활용>서버프로그램 구현]**

## 1. 개발환경 구축

1	2	3	4	5	6	7	8	9	10	11	12	13							
1	3	3	4	4	2	2	4	3	1	3	3	4							

## 2. 공통 모듈 구현

1	2	3	4	5	6	7	8	9	10	11	12								
3	2	2	2	3	1	1	2	4	2	3	4								

### 3. 서버 프로그램 구현

[illegible]

#### 4. 배치 프로그램 구현

[illegible]

## [정답 및 해설] [프로그래밍 언어 활용>서버프로그램 구현>개발환경 구축]

### 개발환경 구축 1. ①

서버프로그램 구현에 대한 설명이다.

### 개발환경 구축 2. ③

개발 도구 분류에는 구현 도구, 테스트 도구, 형상 관리 도구, 빌드 도구가 있다.

### 개발환경 구축 3. ③

- 구현 도구 : Eclipse, Visual Studio Code, IntelliJ, NetBeans 등
- 테스트 도구 : xUnit, Spring Test
- 형상 관리 도구 : CVS, Subversion, Git 등
- 빌드 도구 : Ant, Maven, Gradle 등

### 개발환경 구축 4. ④

- 구현도구: 코드의 작성과 디버깅, 수정 등과 같은 여러가지 작업이 가능하며 프로그램을 개발할 때 가장 많이 사용되는 도구이다.
- 빌드 도구: 작성한 코드에 대한 빌드 및 배포를 수행하며, 각각의 구성 요소들과 모듈들에 대한 의존성 관리를 지원하는 도구이다.
- 테스트 도구: 코드의 기능 검증과 전체의 품질을 높이기 위해 사용되는 도구이다.
- 형상관리 도구: 개발자들이 작성한 코드와 리소스 등 산출물에 대한 버전관리를 위해 사용된다.

### 개발환경 구축 5. ④

- 웹(Web) 서버: HTTP를 이용한 요청/응답(Request/Response)를 처리하기 위한 미들웨어를 설치 되는 서버이다. 주로 웹 브라우저 화면에서 요청하는서비스의 처리하기 위해 HTML, CSS, 이미지 등을 제공하는 웹 서버 애플리케이션의 처리를 담당한다.
- 웹 애플리케이션(Application) 서버: 웹 서비스를 동적(Dynamic)으로 제공하기 위해 WAS(Web Application Server)의 형태인 Resin, Tomcat등 과 서비스에 관련된 애플리케이션이 설치되는 서버이다. 데이터베이스(Database) 서버 - 데이터의 수집, 저장을 위한 용도로 MySql, Oracle, MS-SQL 등 데이터베이스의 SW에 설치되어 있다.
- 파일(File) 서버: 파일저장 하드웨어로 물리적 저장 장치를 활용한 서버이다. 대용량HDD, SSD 등 장치가 주요요소이다.

### 개발환경 구축 6. ②

형상관리에 대한 설명이다.

### 개발환경 구축 7. ②

소프트웨어 환경 구성은 운영체제, 미들웨어, DBMS 이다.

## [정답 및 해설] [프로그래밍 언어 활용>서버프로그램 구현>개발환경 구축]

### 개발환경 구축 8. ④

④ Oracle - DBMS의 종류이다.  
미들웨어(Middleware)의 종류에는 Web Server, WAS, JVM 등이 있다.

### 개발환경 구축 9. ③

형상관리의 특징은 버전관리, 전단계 수행, 추적성 확보이다.

- 버전관리: 소프트웨어 변경 시 버전별로 관리하며 소프트웨어뿐만 아니라 형상 대상의 항목을 모두 관리한다.

- 전단계 수행: 형상관리는 프로젝트의 전 단계에서 수행이며, 사업계획 단계부터 유지 보수 단계에서도 수행된다.

- 추적성 확보: 형상관리를 수행하여 소프트웨어 개발 과정에서 발생 하는 여러 가지 문제점 발생 시 해당 요인을 추적한다.

### 개발환경 구축 10. ①

형상관리 절차의 순서

: 형상식별 > 변경 제어 > 형상상태보고 > 형상검사

### 개발환경 구축 11. ③

개발환경 구축 순서

1. 프로젝트의 요구사항 분석
2. 개발 환경 구성을 위한 필요 도구 설계
3. 개발 대상에 따른 적절한 개발 언어 선정
4. 개발 언어와 하드웨어를 고려한 구현도구 구축
5. 빌드와 테스트 도구 구축

### 개발환경 구축 12. ③

③ 적정성이 아닌 범용성에 대한 설명이다.

- 적정성: 개발하고자 하는 시스템이나 응용 프로그램의 목적에 적합하는가

### 개발환경 구축 13. ④

제어의 역흐름(inversion of control)에 대한 설명이다.

## [정답 및 해설] [프로그래밍 언어 활용>서버프로그램 구현>공통 모듈 구현]

### 공통 모듈 구현 1. ③

편의적 재사용은 내부 재사용과 외부 재사용으로 세분화되어집니다. 상세설명은 보기를 참고하세요.  
(ㄱ) 내부 재사용      (ㄴ) 외부 재사용

### 공통 모듈 구현 2. ②

재사용 프로그래밍 기법으로는 객체지향 프로그래밍, 제네릭프로그래밍, 자동프로그래밍, 메타프로그래밍이 있다.

### 공통 모듈 구현 3. ②

디자인 패턴 (Design pattern)에 대한 설명이다.

### 공통 모듈 구현 4. ②

좋은 모듈이 되기 위해서는 모듈의 응집도는 높아야 하고 결합도는 낮아야 한다.  
- 응집도: 모듈의 독립성을 나타내는 개념으로 하나의 모듈 내부 처리 요소들간에 기능적 연관도를 측정하는 척도이다. 높을수록 독립성이 강하다.  
- 결합도: 소프트웨어 구조에서 모듈간 연관성을 측정하는 척도이며 모듈간 상호의존성을 밀한다. 낮을수록 좋다.

### 공통 모듈 구현 5. ③

모듈화의 원리

- 분할과 지배 (Divide & Conquer): 복잡한 문제를 분해, 모듈 단위로 문제 해결한다.
- 정보 은폐 (information hiding): 어렵거나 변경 가능성이 있는 모듈을 타모듈로부터 은폐한다.
- 모듈의 독립성 (Module Independence): 낮은 결합도와 높은 응집도를 가진다.
- 자료 추상화 (Data Abstraction): 각 모듈 자료구조를 액세스하고 수정하는 함수 내에 자료 구조의 표현 내역을 은폐한다.

### 공통 모듈 구현 6. ①

모듈화의 종류

- 설계적 측면: 모듈 (Module), 컴포넌트 (Component), 서비스 (Service)
- 구현 측면: 매크로 (Macro), 함수 (Function), 인라인 (Inline)

### 공통 모듈 구현 7. ①

결합도가 강한 순서

: 내용 결합도 > 공통 결합도 > 제어 결합도 > 스탬프 결합도 > 데이터 결합도

## [정답 및 해설] [프로그래밍 언어 활용>서버프로그램 구현>공통 모듈 구현]

### 공통 모듈 구현 8. ②

스탬프 결합도(stamp coupling)에 대한 설명이다.

### 공통 모듈 구현 9. ④

결합도에 대한 설명이다.

- 결합도: 모듈내부가 아닌 외부의 모듈과의 연관도 (모듈간의 상호의존성)이다. 소프트웨어 구조에서 모듈간의 관련성을 측정하는 척도이다.

### 공통 모듈 구현 10. ②

응집도에 대한 설명이다.

- 응집도: 정보은닉개념의 확장개념으로, 하나의 모듈은 하나의 기능을 수행하는 집적성을 지칭한다. 모듈의 독립성을 나타내는 개념으로 모듈내부 구성원간의 연관도이다.

### 공통 모듈 구현 11. ③

응집력이 강한 순서

: functional (기능적) > sequential (순차적) > procedural (절차적) > logical(논리적) > coincidental(우연적)

### 공통 모듈 구현 12. ④

절차적 단계에 대한 설명이다.

## [정답 및 해설] [프로그래밍 언어 활용>서버프로그램 구현>서버 프로그램 구현]

### 서버 프로그램 구현 1. ③

버퍼 오버플로에 대한 설명이다.

### 서버 프로그램 구현 2. ①

① 입력 데이터 검증 및 표현은 프로그램 입력값에 대한 검증 누락 또는 부적절한 검증, 데이터의 잘못된 형식 지정으로 인해 발생할 수 있는 보안 약점이다. 코드 오류가 타입 변환 오류, 자원(메모리 등)의 부적절한 반환 등과 같이 개발자가 범할 수 있는 코딩 오류로 인해 유발되는 보안 약점이다.

### 서버 프로그램 구현 3. ④

FTP 바운스 공격은 소프트웨어 취약점 분류에서 권한 혼동 버그 분류이다.

### 서버 프로그램 구현 4. ④

API(Application Programming Interface, 응용 프로그램 프로그래밍 인터페이스)에 대한 설명이다.

### 서버 프로그램 구현 5. ①

규격에 등록된 운영 체제 목록

- HP/UX
- 맥 OS X
- SCO
- 솔라리스

### 서버 프로그램 구현 6. ②

Window API에 대한 설명이다.

## [정답 및 해설] [프로그래밍 언어 활용>서버프로그램 구현>배치 프로그램 구현]

### 배치 프로그램 구현 1. ④

배치 프로그램에 대한 설명이다.

### 배치 프로그램 구현 2. ③

③ 유효하지 않은 데이터의 경우도 처리해서 비정상적인 동작 중단이 발생하지 않아야 한다.

배치프로그램 필요 요소

- 대용량 데이터: 대용량의 데이터를 처리할 수 있어야 한다.
- 자동화: 심각한 오류 상황 외에는 사용자의 개입 없이 동작해야 한다.
- 견고함: 유효하지 않은 데이터의 경우도 처리해서 비정상적인 동작 중단이 발생하지 않아야 한다.
- 안정성: 어떤 문제가 생겼는지, 언제 발생했는지 등을 추적할 수 있어야 한다.
- 성능: 주어진 시간 내에 처리를 완료할 수 있어야 하고, 동시에 동작하고 있는 다른 애플리케이션을 방해하지 말아야 한다.

### 배치 프로그램 구현 3. ②

② Quartz 스케줄러(Scheduler)에 대한 설명이다.

스프링 배치(Spring Batch)의 핵심 기능

- 스프링 프레임워크 기반: 스프링의 DI, AOP 및 다양한 엔터프라이즈 지원 기능을 사용한다.
- 자체 제공 컴포넌트: 배치 처리(데이터베이스나 파일로부터 데이터를 읽거나 쓰는 등)시 공통적으로 필요한 컴포넌트를 제공한다.
- 견고함과 안정성: 선언적 생략과 처리 실패 후 재시도 설정을 제공한다.