

4. 프로그래밍 언어 활용

4.1 서버프로그램 구현 (NCS 5 level)

4.1.1. 서버프로그램 구현 능력 단위 정의

■ 서버프로그램 구현이란 애플리케이션 설계를 기반으로 개발에 필요한 환경을 구성하고, 프로그래밍 언어와 도구를 활용하여 공통모듈, 업무프로그램과 배치 프로그램을 구현하는 능력이다.

4.1.2. 개발환경 구축

표 4-1 응용SW엔지니어링. 프로그래밍 언어 활용, 개발환경 구축

능력단위요소	수행 준거	
개발환경 구축	1.1 응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어의 필요 사항을 검토하고 이에 따라, 개발환경에 필요한 준비를 수행할 수 있다.	
	1.2 응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어를 설치하고 설정하여 개발환경을 구축할 수 있다.	
	1.3 사전에 수립된 형상관리 방침에 따라, 운영정책에 부합하는 형상관리 환경을 구축할 수 있다.	
지식		기술

<ul style="list-style-type: none"> ○ 개발에 필요한 하드웨어 및 소프트웨어 사양 ○ 개발환경에 요구되는 아키텍처 ○ 서버 개발 프레임워크 개념 ○ 설계 산출물 해석 방법 ○ 데이터베이스 개념 ○ 네트워크 개념 ○ 보안 관련 개념 	<ul style="list-style-type: none"> ○ 컴파일러 사용 능력 ○ IDE 도구 활용 능력 ○ 네트워크 활용 기술 ○ 협업도구 활용 능력 ○ 형상관리 도구 활용 능력 ○ 프레임워크 활용 기술 ○ 프로그래밍언어 활용 기술 ○ 보안도구 활용 능력
--	---

※ 학습TIP

자신이 웹 서비스를 구축하는 개발자라고 생각하고, 개발을 위해서 무엇이 필요할지 생각하면서 학습하면 이해가 용이하다.

개발환경 구축에 사용되는 구현 도구, 형상관리 도구, 빌드 도구, 테스트 도구는 암기가 필요하고, 특히 형상관리는 굉장히 출제 빈도가 높다.

※ 실력 양성 문제

실력 양성 문제

[문제 1] 개발환경 구축을 위한 필요사항이 아닌것은?

- ① 목적에 맞는 하드웨어, 소프트웨어의 선정
- ② 개발 인력과 비용에 관련한 명세서의 작성
- ③ 해당 프로젝트의 목적과 구축 설계에 대한 명확한 이해
- ④ 개발에 사용되는 제품들의 성능과 라이선스 그리고 사용 편의성 등에 대한 내용 파악

[문제 2] 다음 중 개발을 위해 사용되는 도구 중 성격이 다른 하나는?

-
- ① CVS
 - ② Subversion
 - ③ xUnit
 - ④ Git

[문제 3] 다음 문장의 빈칸에 들어갈 단어는?

빌드도구는 개발자가 작성한 소스에 대한 빌드 및 배포를 지원하며, 프로젝트에서 사용되는 구성 요소들과 라이브러리들에 대한 () 관리를 지원하는 도구이다.

- ① 안정성
- ② 독립성
- ③ 무결성
- ④ 의존성

[문제 4] 다음중 형상관리 도구의 설명으로 가장 옳은 것은?

- ① 개발자들이 작성한 소스 및 리소스 등 산출물에 대한 버전 관리 도구이다
- ② 프로그램을 개발할 때 가장 많이 사용되는 도구로서 코드의 작성 및 편집, 디버깅 등 다양한 작업이 가능한 도구이다
- ③ 소프트웨어의 품질을 높이기 위해 사용되는 도구로써 작업결과에 대한 리포팅 및 분석 등의 작업이 가능하다
- ④ 사용되는 도구에는 Ant, Maven, Gradle 등이 있다

[문제 5] 정적 웹 서비스를 수행하는 미들웨어로서, 웹 브라우저 화면에서 요청하는 정적 파일을 제공하는 것이 아닌것은?

- ① GWS
- ② Nginx
- ③ JVM

④ IIS

[문제 6] 다음 중 개발 소프트웨어 분류에 대한 설명으로 옳지 않은 것은?

- ① 구현도구는 기능을 논리적으로 결정하기 위해 통합 모델링 언어 지원, Database 설계 지원 등 설계 및 모델링을 지원하는 도구이다.
- ② 요구 사항 관리도구는 목표 시스템의 기능과 제약 조건 등 고객의 요구 사항을 수집, 분석, 추적을 쉽게 할 수 있게 지원한다
- ③ 테스트 도구는 구현 및 개발된 모듈들에 대하여 요구 사항에 적합하게 구현되어 있는지 테스트를 지원하는 도구이다
- ④ 형상관리 도구는 산출물의 변경 사항을 버전별로 관리하여 목표 시스템의 품질 향상을 지원하는 도구이다

[문제 7] 다음 중 형상관리 대상이 되는 형상항목이 아닌것은?

- ① 소프트웨어 요구 사항 명세서
- ② 소스코드 목록 및 소스코드
- ③ 개발에 필요한 참조 문서
- ④ 실행 프로그램

[문제 8] 개발환경 구축의 순서로 올바른 것은?

- (ㄱ) 프로그램의 배포 및 라이브러리 관리를 위한 빌드(Build) 도구를 선정한다
- (ㄴ) 개발환경을 준비하기 위해서 목표 시스템의 환경을 분석한다
- (ㄷ) 개발언어 및 하드웨어 사양을 고려한 구현 도구를 선정한다
- (ㄹ) 요구 사항을 분석하고 분석/설계, 기능, 성능, 인터페이스, 테스트, 보안, 품질 요구 사항등 목록을 작성하고 해당 내용을 분석한다

- ① ㄱ -> ㄴ -> ㄷ -> ㄹ
- ② ㄴ -> ㄹ -> ㄷ -> ㄱ

③ ㄹ -> ㄴ -> ㄷ -> ㄱ

④ ㄴ -> ㄱ -> ㄷ -> ㄹ

[문제 9] 개발언어 선정을 위한 기준으로 고려해야할 항목이 아닌것은?

- ① 이식성
- ② 친밀성
- ③ 의존성
- ④ 범용성

[문제 10] 다음 문장을 설명하는 프레임워크의 특징인 것은?

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

[문제 11] 정보시스템 개발 단계에서 프로그래밍 언어 선택 시 고려할 사항으로 가장 거리가 먼 것은? (이전 기출)

- ① 개발 정보시스템의 특성
- ② 사용자의 요구사항
- ③ 컴파일러의 가용성
- ④ 컴파일러의 독창성

1-② 개발환경 구축과 개발 인력비용 명세서 작성은 무관하다.

2-③ xUnit은 테스트 도구이며 나머지는 형상관리도구이다.

3-④ Maven, Gradle과 같은 빌드도구는 라이브러리의 의존성 관리 역할도 수행한다.

4-①, ② : 구현도구 , ③ : 품질관리도구, ④ : 빌드도구

5-③ JVM은 가상환경 실행을 위한 미들웨어이고 나머지는 웹서버이다.

6-① 구현도구가 아니라 설계모델링 도구 설명이다.

7-③ 개발에 필요한 참조문서는 프로젝트의 산출물이 아니기 때문에 형상관리가 필요없다.

8-② 개발의 시작은 요구사항 분석부터이다. 그 이후 시스템 환경분석, 언어선정, 도구선정 등이 수행된다.

9-③ IT 에서 "의존성"은 대개 나쁜 의미로 사용된다. 특정 환경, 특정 사용자 그룹에 의존적이지 않은 범용성 있는 언어 선정을 고려해야 한다.

10-① 개발자가 아닌, 프레임워크(또는 컨테이너)가 애플리케이션의 흐름을 제어하는 것을 제어의 역전 또는 제어의 역흐름 이라 한다.

11-④ 컴파일러의 독창성은 고려 대상이 아니다.

4.1.3. 공통 모듈 구현

표 4-7 응용SW엔지니어링. 프로그래밍 언어 활용, 공통 모듈 구현

능력단위요소	수행 준거	
공통 모듈 구현	2.1 공통 모듈의 상세 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 업무 프로세스 및 서비스의 구현에 필요한 공통 모듈을 작성할 수 있다.	
	2.2 소프트웨어 측정지표 중 모듈간의 결합도는 줄이고 개별 모듈들의 내부 응집도를 높인 공통모듈을 구현할 수 있다.	
	2.3 개발된 공통 모듈의 내부 기능과 제공하는 인터페이스에 대해 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.	
지식		기술
<ul style="list-style-type: none"> ○ 개발환경에 요구되는 아키텍처 ○ 설계 산출물 해석 방법 ○ 프로그램 언어 활용방법 ○ 데이터베이스 개념 ○ 네트워크 개념 ○ 분산 처리 개념 ○ 소프트웨어 테스트 		<ul style="list-style-type: none"> ○ 컴파일러 사용법 ○ 개발에 필요한 프로그래밍 언어 및 도구 활용 능력 ○ 프로그램 디버깅 기법 ○ 프로그램 코드 검토 기법 ○ 형상관리 도구 활용 기법 ○ 단위테스트 도구 활용기법 ○ 인터페이스 도구/기법 (API, EAI, WebService, 등) ○ 재사용 기법 ○ 모듈화 기법

※ 학습TIP

이 장은 출제 빈도가 굉장히 높다.

재사용의 장점과 단점, 재사용을 위한 모듈화에 대한 개념 이해가 선행되어야 한다.

그리고, 모듈화의 척도인 응집도, 결합도에 대해서는 각각의 의미와 순서를 반드시 암기하고, 꼭 용어는 원문(영어)으로 암기해야한다.

※ 실력 양성 문제

실력 양성 문제

[문제 1] 다음 중 편의적 재사용(opportunistic reuse) 대한 설명으로 다른 것은?

- ① 서드파티에서 만든 컴포넌트를 구하여 사용한다
- ② 컴퍼넌트를 차후에 재사용가능하도록 전략적으로 설계해 나간다
- ③ 프로젝트를 시작할 때 재사용가능한 컴포넌트가 있는지를 찾아보고 재사용한다
- ④ 팀내에서 만든 컴포넌트를 재사용한다.

[문제 2] 다음 문장을 설명하는 모듈화의 원리인 것은?

각 모듈 자료구조를 액세스하고 수정하는 함수내에 자료 구조의 표현 내역을 은폐 한다

- ① 분할과 지배 (Divide & Conquer)
- ② 정보 은폐(information hiding)
- ③ 자료 추상화(Data Abstraction)
- ④ 모듈의 독립성(Module Independance)

[문제 3] 다음의 모듈화의 종류에서 성격이 다른 하나는 무엇인가?

- ① 함수(Function)
- ② 모듈 (Module)
- ③ 서비스(Service)

④ 컴포넌트(Component)

[문제 4] 다음 중 인라인(Inline)에 대한 설명이 올바른 것은?

- ① 프로그램 구현 시 반복되는 부분을 특정 이름을 부여하고 이름을 호출하여 실행할 수 있도록 하는 프로그래밍 기법
- ② 바이너리형태의 재사용 가능한 형태로 인터페이스에 의해 로직을 수행 할 수 있는 모듈 단위
- ③ 프로그램 구현 시 반복되는 부분을 특정 이름을 부여해 놓고 이름을 호출하여 실행 할 수 있도록 하는 프로그램 기법
- ④ 프로그램 구현시 커다란 프로그램의 일부코드로 특정한 작업을 수행하고 상대적으로 다른 코드에 비해 독립적인 모듈

[문제 5] 결합도의 특징 대한 설명으로 틀린 것은?

- ① 에러발생시 오류가 전파되어 다른 오류의 원인이 되는 리플 효과 (Ripple)를 최소화 해야 함
- ② 자료전달이 인터페이스를 통과하므로 인터페이스복잡성에 의존적임
- ③ 클래스 목적에 부합하는 같은 기능영역의 함수들로 구성되어있음
- ④ 서로 다른 상위 모듈에 의해 호출되어 처리 상 연관성이 없는 서로 다른 기능수행

[문제 6] 높은 결합도에서 낮은 결합도의 순서로 올바른 것은?

- | |
|-----------------------|
| (ㄱ) Data coupling |
| (ㄴ) Stamp coupling |
| (ㄷ) Control coupling |
| (ㄹ) External coupling |
| (ㅁ) Common coupling |

(ㄴ) Content coupling

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

[문제 7] 다음 문장의 빈칸에 들어갈 단어는?

결합도는 ()이 좋고, 응집도는 ()이 좋다.

- ① 높음, 높음
- ② 높음, 낮음
- ③ 낮음, 높음
- ④ 낮음, 낮음

[문제 8] 낮은 응집도에서 높은 응집도의 순서로 올바른 것은?

(ㄱ) 우연적
(ㄴ) 순차적
(ㄷ) 논리적
(ㄹ) 시간적
(ㅁ) 통신적
(ㅂ) 기능적
(ㅅ) 절차적

- ① ㄱ -> ㄴ -> ㄷ -> ㄹ -> ㅁ -> ㅂ -> ㅅ
- ② ㄷ -> ㄱ -> ㄹ -> ㅅ -> ㅁ -> ㄴ -> ㅂ
- ③ ㄱ -> ㄷ -> ㄹ -> ㅅ -> ㅁ -> ㄴ -> ㅂ

④ □ -> ㄱ -> ㄴ -> ㄷ -> ㄹ -> ㅁ -> ㅂ -> ㄴ

[문제 9] 다음 문장이 설명하는 재사용의 사례 인 것은?

비슷한 문제를 풀기 위한 범용적인 해법이다.

개념적이고, 개별 문제의 필요에 따라서 수정 가능하다.

추상 클래스와 인터페이스는 특정의 형태에 재사용 가능하다.

- ① 매크로(Macro)
- ② 소프트웨어 라이브러리(Software Library)
- ③ 컴포넌트(Component)
- ④ 디자인패턴(Design Pattern)

[문제 10] 소프트웨어 라이브러리(Software Library) 대한 설명으로 틀린 것은?

- ① 각종 형식으로 정보의 변환, 외부기억장치 액세스, 외부 프로그램과의 인터페이스, 정보의 조작이라고 한 일반적인 조작은, 많은 프로그램으로 필요하게 된다
- ② 성능 향상이나 출력 형식을 바꾸고자 할 때 세부사항을 조절할 수 없는 점과 라이브러리를 취득, 학습, 설정하는 데에 시간과 비용이 든다는 점이다
- ③ 코드 재작업의 매우 일반적인 예로서 라이브러리를 사용하는 것이다.
- ④ 새로운 프로그램을 쓸 때, 라이브러리의 코드를 사용해 작업을 실행하도록 할 수 있어 차바퀴의 재발명과 같이 같은 조작을 실행하는 프로그램을 또 만들어 쓸 필요는 없다

[문제 11] 소프트웨어를 재사용함으로써 얻을 수 있는 이점으로 가장 거리가 먼 것은? (이전 기출)

- ① 새로운 개발 방법론 도입 용이
- ② 생산성 증가
- ③ 소프트웨어 품질 향상
- ④ 프로젝트 문서 공유

[문제 12]한 모듈 내의 각 구성 요소들이 공통의 목적을 달성하기 위하여 서로 얼마나 관련이 있는지의 기능적 연관의 정도를 나타내는 것은? (이전 기출)

- ① cohesion
- ② coupling
- ③ structure
- ④ unity

[문제 13] 모듈의 응집도(Cohesion)에 대한 설명으로 틀린 것은? (이전 기출)

- 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.

[문제 14] 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? (이전 기출)

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

[문제 15] 소프트웨어 재사용과 관련하여 객체들의 모임, 대규모 재사용 단위로 정의되는 것

은? (이전 기출)

- ① Component
- ② Sheet
- ③ Framework
- ④ Cell

[문제 16] 효과적인 모듈화 설계 방법으로 가장 거리가 먼 것은? (이전 기출)

- ① Coupling은 강하게 Cohesion는 약하게 설계한다.
- ② Complexity와 Redundancy를 최대한 줄일 수 있도록 설계한다.
- ③ Maintenance가 용이하도록 설계한다.
- ④ Module 크기는 시스템의 전반적인 기능과 구조를 이해하기 쉬운 크기로 설계한다.

[문제 17] 결합도(Coupling)에 대한 설명으로 틀린 것은? (이전 기출)

- ① 데이터 결합도(Data Coupling)는 두 모듈이 매개변수로 자료를 전달할 때 자료구조 형태로 전달되어 이용될 때 데이터가 결합되어 있다고 한다.
- ② 내용 결합도(Content Coupling)는 하나의 모듈이 직접적으로 다른 모듈의 내용을 참조할 때 두 모듈은 내용적으로 결합되어 있다고 한다.
- ③ 공통 결합도(Common Coupling)는 두 모듈이 동일한 전역 데이터를 접근한다면 공통 결합되어 있다고 한다.
- ④ 결합도(Coupling)는 두 모듈간의 상호작용, 또는 의존도 정도를 나타내는 것이다.

[문제 18] 다음 사항과 관계되는 결합도는? (이전 기출)

- 한 모듈에서 다른 모듈의 내부로 제어 이동
- 한 모듈이 다른 모듈 내부 자료의 조회 또는 변경
- 두 모듈이 동일한 문자(Literals)의 공유

-
- ① Data Coupling
 - ② Content Coupling
 - ③ Control Coupling
 - ④ Stamp Coupling

정답

1-② 계획적 재사용(planned reuse)의 설명

2-③

3-① 함수는 구현측면의 모듈화 방법이다.

4-③, ① 매크로, ② 컴포넌트, ④ 함수

5-③ 3번은 응집도의 특징

6-① "내공외제스자"로 외우면 높은 결합도에서 낮은 결합도의 순서이다.(모듈화도가 높아지는 방향으로 외우는것이 좋다.)

영어 원문으로 자주 출제된다.

7-③ 낮은 결합도, 높은 응집도가 모듈화의 핵심 내용이다.

8-③ "우논시절통순기" 로 외우면 낮은 응집도에서 높은 응집도 순서이다. (모듈화도가 높아지는 순서로 외우는것이 좋다.) 영어 원문으로도 암기가 필요하다.

9-④ 문제의 범용적 해법(Best Practice) 를 얘기하므로, 디자인 패턴을 설명하는 예문이다.

10-③ 라이브러리는 코드 재작업이 아닌 코드 재사용의 대표적인 방법이다.

11-①

12-① coupling(결합도)는 여러 모듈 사이의 연관도를 의미한다. structure, unity는 기능적 연관과 무관하다.

13-② 2번은 순차적 응집도(Sequential cohesion)에 대한 설명이다.

14-③ 제어 신호를 전달하는 결합도는 제어(Control) 결합도이다. 함수 호출시, 제어를 위한 파라미터를 전송하도록 구현하는 형태이다.

15-① 객체들이 모여 하나의 독립적 기능을 수행할 수 있는 재사용 단위는 컴포넌트이다.

Framework는 개발 생산성 향상을 위한 틀과 라이브러리를 제공해 주는 반제품 형태의 프로그램이고, Cell, Sheet 는 재사용과 무관하다.

16-① 결합도(Coupling)는 약하게, 응집도(Cohesion)는 강하게 설계해야 한다.

17-① 1번은 스탬프 결합도(Stamp Coupling)에 대한 설명이다.

18-② 다른 모듈 내부를 직접 참조하는 결합도는 내용(Content) 결합도이다.

4.1.4. 서버 프로그램 구현

표 4-13 응용SW엔지니어링. 프로그래밍 언어 활용. 서버 프로그램 구현

능력단위요소	수행 준거	
서버 프로그램 구현	3.1	업무 프로세스 맵과 세부 업무 프로세스를 확인할 수 있다.
	3.2	세부 업무프로세스를 기반으로 프로그래밍 언어와 도구를 활용하여 서비스의 구현에 필요한 업무 프로그램을 구현할 수 있다.
	3.3	개발하고자 하는 목표 시스템의 잠재적 보안 취약성이 제거될 수 있도록 서버 프로그램을 구현할 수 있다.
	3.4	개발된 업무 프로그램의 내부 기능과 제공하는 인터페이스에 대해 테스트를 수행할 수 있다.
지식		기술
<ul style="list-style-type: none"> ○ 구현에 요구되는 아키텍처 ○ 설계 산출물 해석방법 ○ 프로그램 언어 활용방법 ○ 단위 업무 ○ 단위테스트 기술동향 및 도구 ○ 소프트웨어 테스트 ○ 결함 관리 		<ul style="list-style-type: none"> ○ 컴파일러 사용법 ○ IDE 도구 활용 능력 ○ 프로그램 디버깅 기술 ○ 프로그램 코드 검토 기술 ○ 형상관리 도구 활용 기법 ○ 단위테스트 도구 활용기법 ○ 인터페이스 도구/기법 (API, EAI, WebService, 등)

※ 학습TIP

필기에서 이 파트는 보안 취약성 식별과 API의 개념에을 다룬다.

SDLC 상 수행되는 보안 취약성 식별 기법을 알아야 하고, 특히 개발보안에 사용되는 시큐어 코딩은 반드시 암기해야 한다.

※ 실력 양성 문제

실력 양성 문제

[문제 1] 다음의 보안 취약성에 대한 정의 설명으로 옳지 않은 것은?

- ① 공격자가 결함에 대한 접근 그리고 공격자가 결함에 대한 익스플로잇 가능성을 말한다.
- ② 취약점(vulnerability)은 공격자가 시스템의 정보 보증을 높이는데 사용되는 약점이다.
- ③ 시스템 민감성 또는 결함에 대한 공격 영역을 말한다.
- ④ 취약점을 익스플로잇 하기 위해서, 공격자는 반드시 시스템의 약점에 접속할 수 있는 적어도 하나의 툴이나 기법을 가져야 한다

[문제 2] 다음 문장이 설명하는 소프트웨어 취약점은?

메모리를 다루는 데에 오류가 발생하여 잘못된 동작을 하는 프로그램 취약점

- ① 코드 인젝션
- ② 포맷 스트링 버그
- ③ 버퍼 오버플로
- ④ 심볼릭 링크 경쟁

[문제 3] 다음의 소프트웨어 취약점에 대한 설명으로 옳지 않은 것은?

- ① HTTP 헤더 Injection은 공격자가 Response 헤더 필드에 개행문자 등을 삽입함으로써 임의의 Response 헤더 필드나 Body를 추가하는 수동적 공격이다
- ② SQL 삽입공격은 유효하지 않은 데이터를 실행함으로써 야기되는 소프트웨어 버그의 부당한 사용으로 하는 공격이다.
- ③ 웹 애플리케이션에서의 사이트 간 스크립팅은 취약성이 있는 웹 사이트를 방문한 사용자의 브라우저에서 부정확한 HTML태그나 JavaScript를 동작 시키는 공격이다.
- ④ 디렉토리 접근 공격은 비공개 디렉토리의 파일에 대해서 부정하게 디렉토리 패스를 가로질러 액세스하는 공격이다.

[문제 4] 취약점 익스플로잇 절차의 Activity 순서로 올바른 것은?

- (ㄱ) 모의해킹
 - (ㄴ) 보안 요구사항 식별
 - (ㄷ) 시큐어코딩 적용
 - (ㄹ) 보안통제 수립

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

[문제 5] 다음 중 시스템의 물리적 환경 취약점이 아닌 것은?

- ① 직원
- ② 소프트웨어
- ③ 조직 내에서 관리 절차와 보안방식
- ④ CCTV

[문제 6] 다음 문장의 빈칸에 들어갈 단어는?

()는 응용 프로그램에서 사용할 수 있도록, 운영 체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스를 말한다

- ① Virtual Machine
- ② Macro
- ③ Component
- ④ API

[문제 7] 다음 Windows API중 사용자 인터페이스 집합에 포함되지 않는 것은?

- ① 공통 대화 상자 라이브러리
- ② 윈도 셸(Windows Shell)
- ③ 그래픽 장치 인터페이스 (GDI)
- ④ 공통 컨트롤 라이브러리

[문제 8] 다음 중 API의 설명으로 옳지 않은 것은?

- ① 웹 API는 웹 애플리케이션 개발에서 다른 서비스에 요청을 보내고 응답을 받기 위해 정의된 명세를 일컫는다
- ② 웹 API는 사용 시 블로그 API를 이용해서 블로그에 접속해야만 글을 올릴 수 있다.
- ③ 자바 API(Java API)는 자바를 사용하여 쉽게 구현할 수 있도록 한 클래스 라이브러리의 집합이다
- ④ 윈도우 API(Windows API, WinAPI)는 마이크로소프트 윈도 운영 체제들이 사용하는 API이다.

[문제 9] 중요한 리소스를 32비트 윈도우 시스템에서 사용할 수 있게 해주는 API가 포함된 파일은 무엇인가?

- ① gdi32.dll
- ② kernel32.dll
- ③ user32.dll
- ④ advapi32.dll

[문제 10] 다음 문장의 빈칸에 들어갈 단어는?

()란 특정사이트의 특정 서비스를 이용할 수 있는 인터페이스를 외부사이트에 누구나 가공하여 사용할 수 있도록 공개한 것으로 대표적인 Web 2.0 기술이다.

- ① Java API

② Windows API

③ Web API

④ Open API

정답

1-② 보증을 낮추어야 한다.

2-③ 버퍼는 메모리를 의미한다.

3-② 코드인젝션 설명이다.

4-②

5-④ CCTV 취약점이 아니 물리적 보안 방법이다.

6-④ API에 대한 설명이다.

7-③ GDI는 사용자 인터페이스가 아니고, 그래픽 장치 인터페이스이다.

8-② 블로그 API를 사용하면 블로그 접속 없이 글을 올릴 수 있다.

9-②

10-④ Open API에 대한 설명이다. 기상청의 날씨정보, 포털의 검색서비스 등 많은 서비스
들이 Open API를 통해 제공된다.

4.1.5. 배치 프로그램 구현

표 4-19 응용SW엔지니어링. 프로그래밍 언어 활용, 배치 프로그램 구현

능력단위요소	수행 준거	
배치 프로그램 구현	4.1 애플리케이션 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 배치 프로그램 구현 기술에 부합하는 배치 프로그램을 구현 할 수 있다.	
	4.2 목표 시스템을 구성하는 하위 시스템간의 연동 시, 안정적이고 안전하게 동작할 수 있는 배치 프로그램을 구현 할 수 있다.할 수 있다.	
	4.3 개발된 배치 프로그램을 테스트를 수행할 수 있다.	
지식		기술
<ul style="list-style-type: none"> ○ 구현에 요구되는 아키텍처 ○ 설계 산출물 해석방법 ○ 프로그램 언어 활용방법 ○ 데이터베이스 개념 ○ 네트워크 개념 ○ 분산 프로그래밍 개념 ○ 소프트웨어 테스트 		<ul style="list-style-type: none"> ○ 컴파일러 사용법 ○ 개발에 필요한 프로그래밍 언어 및 도구 활용 능력 ○ 프로그램 디버깅 기법 ○ 프로그램 코드 검토 기법 ○ 형상관리 도구 활용 기법 ○ 단위테스트 도구 활용기법 ○ 인터페이스 도구/기법 (API, EAI, WebService, 등)

※ 학습TIP

배치 프로그램은 서비스 구축에 반드시 필요한 부분은 아니지만, 배치 프로그램의 개념과 어떤 업무에 사용되는지 그리고, 배치 스케줄러가 무엇이고 어떤 역할을 하는지에 대한 이해는 필요하다.

※ 실력 양성 문제

실력 양성 문제

[문제 1] 다음 문장의 빈칸에 들어갈 단어를 넣으시오?

배치 프로그램이란 사용자와의 상호 작용 없이 일련의 작업들을 작업 단위로 묶어 ()
으로 ()하거나 정해진 규칙에 따라 ()하는 것이다

[문제 2] 배치 프로그램의 필수 요소에 대한 설명으로 틀린 것은?

- ① 대용량의 데이터를 처리할 수 있어야 한다.
- ② 심각한 오류 상황 외에도 사용자의 개입으로 처리 가능하게 동작해야 한다.
- ③ 유효하지 않은 데이터의 경우도 처리해서 비정상적인 동작 중단이 발생하지 않아야 한다.
- ④ 어떤 문제가 생겼는지, 언제 발생했는지 등을 추적할 수 있어야 한다.

[문제 3] 일괄 처리(Batch Processing)를 위해 주기적으로 발생하거나 반복적으로 발생하는 작업을 지원하는 도구는 무엇인가?

- ① 배치 트레이서(Tracer)
- ② 배치 스케줄러(Scheduler)
- ③ 배치 모니터링(Monitoring)
- ④ 배치 분석기(Analyzer)

[문제 4] 스프링 배치(Spring Batch)의 핵심 컴포넌트 중 특정 크기를 갖는 아이템 목록을 의미하는 것은?

- ① Job
- ② Step

③ Item

④ Chunk

[문제 5] 스프링 배치(Spring Batch)의 핵심 컴포넌트에 대한 설명으로 틀린 것은?

① Job Repository은 Job Execution 관련 메타데이터를 저장하는 기반 컴포넌트이다.

② Item Writer은 Item Chunk를 데이터 소스에 저장하는 컴포넌트이다.

③ JPA(Java Persistence API)는 스와핑 기능을 제공한다.

④ Job Launcher은 Job Execution을 실행하는 기반 컴포넌트이다.

[문제 6] 스프링 배치(Spring Batch)의 핵심 기능에 대한 설명으로 틀린 것은?

① 스프링 프레임워크 기반

② 작업 자동 분할과 추적성 제공

③ 자체 제공 컴포넌트

④ 견고함과 안정성

[문제 7] 쿼츠(Quartz) 스케줄러의 구성 요소가 아닌 것은?

① Task

② Trigger

③ Scheduler

④ JobDetail

[문제 8] 다음 문장이 설명하는 쿼츠(Quartz) 스케줄러의 구성 요소는?

정의한 Job 개체의 실행 스케줄을 정의하는 개체로서 Scheduler 개체에게 Job 수행 시점을 알려 주는 개체이다
--

-
- ① Task
 - ② JobDetail
 - ③ Scheduler
 - ④ Trigger

[문제 9] 배치프로그램 구현의 순서로 올바른 것은?

- (ㄱ) 애플리케이션 설계를 기반으로 배치 프로그램을 구현한다
- (ㄴ) 애플리케이션 설계를 기반으로 배치 프로그램을 확인한다
- (ㄷ) 프로그램 관리 대장의 ID와 일치하는 배치 설계를 확인한다
- (ㄹ) 배치 프로그램을 수행하기 위한 스케줄러를 등록한다

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

[문제 10] 다음중 배치프로그램 구현 시 수행할 작업이 아닌 것은?

- ① SQL을 작성한다.
- ② 데이터 접근 오브젝트(DAO)를 작성한다.
- ③ 배치 설계서를 작성한다
- ④ I/O 오브젝트(DTO/VO)를 정의한다.

1- 정기적 , 반복 수행, 일괄 처리

2-② 사용자 개입 없이 처리 가능해야 한다.

3-②

4-④

5-③ 스와핑이 아닌 페이징 기능을 제공한다.

6-② 스프링 배치의 핵심 기능은 스프링 프레임워크 기반, 자체 제공 컴포넌트, 견고함과 안정성이다.

7-① 쿼츠에서는 작업 단위를 Job이라고 한다. 쿼츠의 구성요소는 Job, Trigger, JobDetail, Job Scheduler가 있다.

8-④ Job을 실행시킬 스케줄링 조건(ex. 반복횟수, 시작시간) 등을 담고 있는 Scheduler가 이 정보를 기반으로 Job을 수행시키도록 하는데, 이를 Trigger라 한다.

9-④ 배치프로그램의 구현 순서는 먼저, 배치 프로그램 구현 대상을 확인하고, 배치프로그램의 ID를 설계하고, 마지막으로 배치 프로그램을 Job Scheduler에 등록한다.

10-③ 배치 설계서는 배치 설계 시 작성한다.

4.2 프로그래밍 언어 활용 (NCS 3 level)

4.2.1. 프로그래밍 언어 활용 능력 단위 정의

12. 프로그래밍 언어 활용이란 응용소프트웨어 개발에 사용되는 프로그래밍 언어의 기초문법을 적용하고 언어의 특징과 라이브러리를 활용하여 기본 응용소프트웨어를 구현하는 능력이다

4.2.2. 기본문법 활용

표 4-23 응용SW엔지니어링. 프로그래밍 언어 활용. 기본문법 활용

능력단위요소	수행 준거	
기본문법 활용	1.1 응용소프트웨어 개발에 필요한 프로그래밍 언어의 데이터 타입을 적용하여 변수를 사용할 수 있다	
	1.2 프로그래밍 언어의 연산자와 명령문을 사용하여 애플리케이션에 필요한 기능을 정의하고 사용할 수 있다.	
	1.3 프로그래밍 언어의 사용자 정의 자료형을 정의하고 애플리케이션에서 사용할 수 있다.	
지식		기술
<ul style="list-style-type: none"> ○ 프로그래밍 언어 기본 문법 ○ 프로그래밍 언어 문법 오류 ○ 프로그래밍 언어 기능 구현을 위한 알고리즘 ○ 디버깅 기법 ○ 코드 검토 기법 		<ul style="list-style-type: none"> ○ 컴파일러 사용 능력 ○ 프로그램 언어 활용 능력 ○ IDE 도구 활용 능력 ○ 프로그램 디버깅 기술 ○ 프로그램 코드 검토 능력

※ 학습TIP

기본 문법은 실기 시험을 위해 필수적이다. 특히, Java는 대표적인 객체지향 언어로, Python은 대표적인 스크립트 언어로 학습이 필요하다.

실기 시험의 기본이 되는 파트이므로, 비중있는 학습이 필요하고, 필기에 출제가 가능한

데이터 타입, 연산자의 유형, 연산자 우선순위 등은 암기해야 한다.

※ 실력 양성 문제

실력 양성 문제

[문제 1] 다음 문장을 설명하는 데이터 유형인 것은?

조건이 참인지 거짓인지 판단하고자 할 때 사용한다

- ① 문자 타입
- ② 문자열 타입
- ③ 부동 소수점 타입
- ④ 불린 타입

[문제 2] 다음 중 데이터 유형과 설명으로 틀린 것은?

- ① 문자 타입은 문자 하나를 저장하고자 할 때 사용한다.
- ② 문자열 타입은 나열된 여러 개의 문자를 저장하고자 할 때 사용한다.
- ③ 정수 타입은 정수 값을 저장하고자 할 때 사용한다.
- ④ 부동 소수점 타입은 소수점을 포함하는 정수 값을 저장하고자 할 때 사용한다.

[문제 3] 다음 중 변수 작성 규칙에 대한 설명으로 틀린 것은?

- ① 수집된 개발 표준 관련 문서로부터 변수 작성 규칙을 확인한다
- ② 변수는 프로그램 소스코드의 공유, 유지 관리, 표준화 등을 위해 일정한 규칙에 따라 작성된다
- ③ 개발 표준 정의서, 개발자 가이드, 디자인 정의서 등 변수 작성 규칙을 확인할 수 있는 관련 자료들을 수집한다
- ④ 수집된 자료에서 제공하는 정보가 부족한 경우에는 변수명이 변수값의 특징을 잘 반영할 수 있도록 변수명을 국제 표준화된 문서를 참조하여 설정한다.

[문제 4] 다음 중 변수명을 지정할 때에 고려사항이 아닌 것은?

- ① 변수명의 최소 길이를 확인한다.
- ② 대소문자의 구별 여부를 확인한다.
- ③ 사용할 수 있는 문자의 집합을 확인한다
- ④ 변수명으로 사용이 어려운 키워드를 포함하고 있지 않은지 확인한다.

[문제 5] Java언어의 short 타입의 크기와 범위로 올바른 것은?

- ① 1byte, -128~127
- ② 2byte, -32,768~32,767
- ③ 4byte, -2,147,483,648~2,147,483,647
- ④ 8byte, -9,223,372,036,854,775,808~9,223,372,036,854,775,807

[문제 6] 다음 연산을 표현하는 설명이 옳은 것은?

$a \% b$

- ① 양쪽의 값을 더한다
- ② 왼쪽 값을 오른쪽 값으로 나눈다.
- ③ 왼쪽 값을 오른쪽 값으로 나눈 나머지를 계산한다.
- ④ 왼쪽 값에서 오른쪽 값을 뺀다.

[문제 7] 두 개의 논리값 중 하나가 참이면 참(True)을 반환하고, 그렇지 않으면 거짓(False)을 반환하는 논리연산자는?

- ① &&
- ② >>
- ③ <<

④ ||

[문제 8] 0과 1의 각 자리에 대한 연산을 수행하며, 0 또는 1의 결과값을 가지는 연산자는?

- ① 산술 연산자(Arithmetic Operator)
- ② 시프트 연산자(Shift Operator)
- ③ 논리 연산자(Logic Operator)
- ④ 비트 연산자(Bit Operator)

[문제 9] 다음 문장이 설명하는 연산 표기법은?

피연산자를 먼저 쓰고 연산자를 맨 뒤에 쓰는 표기법 예) a b c * +
--

- ① 전위(Prefix) 표기법
- ② 중위(Infix) 표기법
- ③ 후위(Postfix) 표기법
- ④ 혼합(Mix) 표기법

[문제 10] Java언어의 연산자 우선순위중 가장 우선인 순서로 올바른 것은?

(ㄱ) &, ^, , &&, (ㄴ) (), [] (ㄷ) <<, >>, >>> (ㄹ) ?: (ㅇ) !, ~, ++, --
--

- ① ㄱ -> ㄴ -> ㄷ -> ㄹ -> ㅇ
- ② ㄴ -> ㅇ -> ㄷ -> ㄱ -> ㄹ
- ③ ㄴ -> ㅇ -> ㄷ -> ㄹ -> ㄱ

④ □ -> L -> C -> 7 -> 2

정답

1-④

2-④ 정수값이 아닌 실수값을 저장한다.

3-④

4-① 변수명의 최대 길이를 확인한다.

5-②

6-③ ex) $10 \% 3 = 1$

몫 = 3, 나머지 = 1

7-④ 2번, 3번은 시프트 연산자이고, 1번 && 연산자는 두 논리값 모두 참일 경우만 참을 반환한다.

8-④

9-③ 후위표기를 사용하면 괄호 없이 연산 우선순위 파악이 가능하고 스택을 이용한 계산기 등의 프로그램에 유용하다. (컴퓨터 입장에서 중위보다 빠르게 처리가 가능하다)

10-②

4.2.3. 언어특성 활용

표 4-36 응용SW엔지니어링. 프로그래밍 언어 활용, 언어특성 활용

능력단위요소	수행 준거	
언어특성 활용	2.1 프로그래밍 언어별 특성을 파악하고 설명할 수 있다.	
	2.2 파악된 프로그래밍 언어의 특성을 적용하여 애플리케이션을 구현할 수 있다.	
	2.3 애플리케이션을 최적화하기 위해 프로그래밍 언어의 특성을 활용할 수 있다.	
지식		기술
<ul style="list-style-type: none"> ○ 절차적 프로그래밍 언어 ○ 객체지향 프로그래밍 언어 ○ 스크립트 언어 ○ 선언형 언어 ○ 디버깅 기법 ○ 코드 검토 기법 		<ul style="list-style-type: none"> ○ 컴파일러 사용 능력 ○ 개발에 필요한 프로그래밍 언어 및 도구 활용 능력 ○ 프로그램 디버깅 기술 ○ 프로그램 코드 검토 능력

※ 학습TIP

객체지향의 클래스와 객체의 개념 그리고, 캡슐화, 정보은닉, 다형성등의 객체지향 특성은 출제 빈도가 매우 높다. 용어는 원어(영어)로 암기하고, 개념을 확실히 이해하자. 최근 인공지능으로 부각된 대표적인 스크립트 언어인 Python에 대한 이해도 필요하다.

가. 절차적 프로그래밍 언어 (*)

① 절차적 프로그래밍의 개요

(ㄴ) 절차적 프로그래밍(procedural programming)은 절차지향 프로그래밍 혹은 절차지향적 프로그래밍이라고도 불리며 프로시저 호출의 개념을 바탕으로 하고 있는 프로그래밍을 의미한다.

(ㄷ) 프로시저는 루틴, 하위프로그램, 서브루틴, 메서드, 함수라고도 하는데, 수행되어야 할 연속적인 계산 과정을 포함하고 있다.

(ㄹ) 프로그램의 아무 위치에서나 프로시저를 호출될 수 있는데, 다른 프로시저에서도 호출 가능하고 심지어는 자기 자신에서도 호출 가능하다.

② 절차적 프로그래밍의 장점

12. 절차적 프로그래밍은 복잡도가 지나치지 않고 유지보수하기 쉽기 때문에 단순한 순차적 프로그래밍이나 비구조적 프로그래밍보다 여러 상황에서 장점이 많다.

표 4-37 절차적 프로그래밍 장점

장점	설명
코드의 재사용	같은 코드를 다른장소에서 호출만으로 재사용이 가능하다
쉬운 프로그램 흐름 파악	GOTO문, JUMP문 사용이 아닌 쉬운 흐름 파악이 가능하다.
모듈화 및 구조화	재사용 단위인 모듈로 구성하여 프로그램을 구조화 한다.

③ 프로시저와 모듈성

13. 크고 복잡한 프로그램을 작성할 때 모듈성은 꼭 필요하다.

14. 모듈성은 프로시저에 어떤 형태의 입력과 출력이 이루어지는지 명확한 규칙을 정하여 구현할 수 있다.

표 4-38 모듈성의 특성

특성	설명
매개변수와 반환값	입력은 주로 인수(인자, 매개변수)의 형태로 문법적으로 지정하고, 출력은 결과값(반환값)으로 지정한다

유효범위(스코프)	프로시저가 모듈성을 잘 지킬 수 있게 해 준다. 이것은 프로시저가 다른 프로시저의 변수에 접근하거나 그 반대의 경우를 막아준다. 서로 다른 장소에서 같은 이름의 변수를 사용하여 혼동되는 것을 막아주고, 다른 프로시저의 영역을 침해하는 것을 막아준다.
명시적 허가	명시적인 허가 없이는 방금전의 자기 자신의 변수에도 접근할 수 없다.
약결합	변수가 프로그램의 부분들 사이에서 독립적 관계에 있다면 이것은 모듈성을 강하게 한다.

④ 절차적 프로그래밍 언어

✓ 절차적 프로그래밍 언어들은 절차적 프로그래밍 접근 방식을 따름으로써 프로그래머의 작업을 수월하게 한다.

표 4-39 대표적인 절차적 프로그래밍언어

언어	설명
C	정규화된 표준 C 라이브러리를 이용한 프로그래밍 언어로 유닉스 시스템의 바탕 프로그램은 모두 C로 작성되었다.
포트란	주로 과학적인 계산을 하기 위해 시작된 컴퓨터 프로그램 언어이다.
알골	알고리즘의 연구개발에 이용하기 위한 목적으로 만들어졌으며 ALGOL 계 언어의 총칭이다.
베이직	교육용으로 개발되어 언어의 문법이 쉬운 절차형 언어이다.

나. 객체지향 프로그래밍 언어 (*****)

① 객체 지향 프로그래밍 개요

✓ 객체지향은 구조적 방식의 문제점을 극복하고 인간이 사고하는 방식대로 프로그램을 개발하려는 노력으로부터 탄생한다.

✓ 객체(object)라는 실세계에 존재하거나 또는 개념적인 그 어떤 것을 표현한 것을 이용하여 프로그래밍 하는 방식을 말한다.

표 4-40 절차지향 프로그래밍과 객체지향 프로그래밍 비교

절차지향 프로그래밍	객체지향 프로그래밍
<p>✓ 모듈, 변수, 함수를 사용하여 개발하는 방법이다. 프로그램을 순차적으로 수행시키는 방법으로 자료 구조와 명령 중심으로 프로그램을 구성한다.</p>	<p>✓ 객체와 객체 간의 통신을 통해 프로그램이 구현된다.</p> <p>✓ 객체와 속성, 메소드로 구성되는데, 절차 지향에서의 모듈은 객체에 대응되고, 변수는 속성에 대응되며, 함수는 메소드에 대응된다</p>

✓ 객체 지향 프로그래밍의 특징은 기본적으로 캡슐화, 추상화, 상속, 다형성 등의 특징이 존재한다. 객체 지향 프로그래밍은 위의 특징을 이용하여 시스템의 복잡성을 제어하기 위해 서로 연관되는 기능을 하는 것이다.

표 4-41 객체지향 프로그래밍의 특징

특징	설명
<p>캡슐화 (Encapsulation)</p>	<p>✓ 속성(데이터)과 메소드(연산)을 하나로 결합하여 객체로 구성하는 것이다.</p> <p>✓ 이를 통해 변경 발생 시 오류의 파급효과가 적어지고, 소프트웨어의 재사용성이 높아지며, 인터페이스가 단순화된다.</p>
<p>정보 은닉 (Information Hidding)</p>	<p>✓ 다른 객체에게 자신의 정보를 숨기고 자신의 연산만을 통하여 접근한다.</p> <p>✓ 유지보수와 소프트웨어 확장 시 오류를 최소화할 수 있다.</p> <p>✓ 캡슐화된 클래스를 선언 시, 그 클래스를 구성하는 속성, 메소드에 대하여 public, private, protected 선언하여 정보은닉을 실현한다.</p>
<p>자료 추상화 (Abstraction)</p>	<p>공통 성질을 추출하여 슈퍼클래스를 설정하는 것</p> <p>1) 기능추상화 : 클래스 내 메소드를 정의한다. 예) obj.getName()</p> <p>2) 자료(데이터)추상화 : 객체 클래스 자체를 데이터 타입으로 사용한다. 예) String, Class</p> <p>3) 제어추상화 : 제어행위에 대한 개념화, 명령 및 이벤트를 말한다. 예) if, for, while</p>

<p>상속</p> <p>(Inheritance)</p>	<p>하위 클래스에게 자신의 속성과 메소드를 사용할 수 있도록 허용하는 것이다.</p> <p>1) 단일 상속 : 부모와 자식 클래스 간의 관계가 슈퍼클래스와 서브클래스로 유지한다.</p> <p>2) 다중 상속 : 하나의 클래스가 하나 이상의 클래스로부터 상속 받는다.</p> <p>3) 반복 상속 : 같은 조부모 클래스로부터 상속 받은 두 부모 클래스로부터 상속 받는 것이다.</p>
<p>다형성</p> <p>(Polymorphism)</p>	<p>✓ 동일한 이름의 오퍼레이션(메소드)이 각 클래스마다 다른 사양으로 정의될 수 있다는 것을 말함.</p> <p>✓ 다시 말해, 메시지에 의해 객체가 연산을 수행하게 될 때 하나의 메시지에 대해 각 객체가 가지고 있는 고유한 방법으로 응답할 수 있는 능력을 말함.</p> <p>1) Overloading : 메소드의 이름은 같으나 argument 나 retrun type 이 다른 경우이다.</p> <p>2) Overriding : Argument 와 return type 같은 경우이다.</p>

② 객체 지향 프로그래밍의 구성 요소

✓ 객체 지향 프로그래밍은 객체, 클래스, 메시지를 이용하여 개발하는 방식으로, 각 구성 요소에 대한 내용은 다음과 같다.

표 4-42 객체지향 프로그래밍 구성요소

구성요소	설명
객체(Object)	개체, 속성, 메소드로 구성된 클래스의 인스턴스를 의미한다.
클래스(Class)	객체의 타입을 정의하고 객체를 생성하는 틀을 의미한다
메시지(Message)	객체 간의 통신을 의미하며, 객체에게 어떤 행위를 하도록 지시한다.

③ 객체의 구성 요소

✓ 객체는 현실 세계 또는 개념적으로 존재하는 사물들로, 개체(Entity)를 데이터 속성(Attribute)과 메소드(Method)로 결합시킨 형태로 표현한 것을 의미하며, 각 구성 요소에 대한 내용은 다음과 같다.

표 4-43 객체 구성요소

구성요소	설명
개체(Entity)	현실 세계에 보이는 본질을 의미한다.
속성(Attribute)	자료 저장소 역할을 하며, 절차 지향 프로그래밍의 변수와 대응된다.
메소드(Method)	호출 단위를 의미하며, 절차 지향 프로그래밍의 함수와 대응된다. 메시지를 받아 실행해야 할 구체적인 동작을 수행한다.

④ 클래스(Class)

✓ 클래스는 객체 지향 프로그래밍에서 객체를 표현하는 추상 데이터 타입으로 객체를 생성하는 틀이다. 기존 언어에서 지원하던 사용자 정의 자료형 및 서브프로그램의 개념을 발전시켜 JAVA, C++ 등에서 클래스를 통해 추상화된 자료형을 제공한다.

✓ 객체지향에서 하나 이상의 유사한 객체들을 묶어 공통된 특성을 표현한 데이터 추상화를 의미하기도 한다.

⑤ 객체 지향 프로그래밍 언어

✓ 객체 지향 프로그래밍은 프로그램을 유연하고 변경이 용이하게 만들기 때문에 대규모 소프트웨어 개발에 많이 사용된다. 또한 프로그래밍을 더 배우기 쉽게 하고 소프트웨어 개발과 보수를 간편하게 하며, 보다 직관적인 코드 분석을 가능하게 하는 장점을 갖고 있다.

표 4-44 대표적인 객체 지향 프로그래밍 언어

언어	설명
C#	마이크로소프트에서 개발한 객체 지향 프로그래밍 언어로, 닷넷 프레임 워크의 한 부분으로 만들어졌다.

자바	(과거) 썬 마이크로시스템즈에서 개발된 객체지향 프로그래밍 언어.
델파이	기본적인 문법은 파스칼 문법과 같지만 파스칼에 여러 기능들이 추가되어 현재는 델파이라는 언어로 존재한다
스위프트(swift)	애플의 iOS와 macOS를 위한 프로그래밍 언어이며 오브젝티브-C와 함께 공존할 목적으로 만들어졌다.

※ 실력 양성 문제

실력 양성 문제

[문제 1] 절차 지향 프로그래밍의 설명으로 옳지 않은 것은?

- ① 프로시저 호출의 개념을 바탕으로 하고 있는 프로그래밍 패러다임을 의미하기도 한다.
- ② 객체라는 개념이 등장하기 이전에 모듈, 변수, 함수를 사용하여 개발하는 방법이다
- ③ 객체와 객체 간의 통신을 통해 프로그램이 구현된다.
- ④ 프로그램의 아무 위치에서나 프로시저를 호출될 수 있는데, 다른 프로시저에서도 호출 가능하고 심지어는 자기 자신에서도 호출 가능하다

[문제 2] 다음 중 절차적 프로그래밍 언어가 아닌 것은?

- ① 포트란
- ② 델파이
- ③ 베이직
- ④ C

[문제 3] 객체지향 프로그래밍의 특징이 아닌것은?

- ① 모듈화
- ② 상속
- ③ 다형성
- ④ 자료 추상화

[문제 4] 다음 문장을 설명하는 객체지향 프로그래밍의 특징인 것은?

불필요한 정보는 숨기고 중요한 정보만을 표현함으로써 프로그램을 간단히 만드는 것이다

- ① 자료 추상화
- ② 다중 상속
- ③ 동적 바인딩
- ④ 다형성

[문제 5] 다음 문장을 설명하는 것은?

객체 지향 프로그래밍에서 객체를 표현하는 추상 데이터 타입으로 객체를 생성하는 틀이다. 기존 언어에서 지원하던 사용자 정의 자료형 및 서브프로그램의 개념을 발전시켜 JAVA, C++ 등에서 추상화된 자료형을 제공한다

- ① 속성(Attribute)
- ② 개체(Entity)
- ③ 메소드(Method)
- ④ 클래스(Class)

[문제 6] 다음 중 객체 지향 프로그래밍 언어가 아닌 것은?

- ① C#
- ② **C**
- ③ 자바
- ④ C++

[문제 7] 다음 문장의 빈칸에 들어갈 단어는?

()란 응용 소프트웨어를 제어하는 컴퓨터 프로그래밍 언어를 가리킨다. ()는 응용 프로그램과 독립하여 사용되고 일반적으로 응용 프로그램의 언어와 다른 언어로 사용되어 최종사용자가 응용 프로그램의 동작을 사용자의 요구에 맞게 수행할 수 있도록 해준다

- ① 절차적 프로그래밍 언어
- ② 객체지향 프로그래밍 언어
- ③ 스크립트 언어
- ④ 선언형 언어

[문제 8] 다음 중 스크립트 언어가 아닌 것은?

- ① 파이썬
- ② JCL
- ③ 스위프트
- ④ 루아

[문제 9] 다음 중 스크립트 언어의 특징이 아닌 것은?

- ① 특수 분야 언어(Domain-specific language, DSL)의 형태로 자주 사용된다.
- ② 매우 빠르게 배우고 작성하기 위해 고안되었다.
- ③ 짧은 소스 코드 파일이나 REPL(Read-eval-print-loop)로 상호작용하는데 적합하다.
- ④ 상대적으로 단순한 구문과 의미를 내포한다.

[문제 10] 다음 중 선언형 프로그래밍 언어가 아닌 것은?

-
- ① OpenLaszlo
 - ② SQL
 - ③ XSL Transformations
 - ④ Oject-C

[문제 11] 객체 지향 개념 중 하나 이상의 유사한 객체들을 묶어 공통된 특성을 표현한 데이터 추상화를 의미하는 것은? (이전 기출)

- ① 메소드(method)
- ② 클래스(class)
- ③ 상속성(inheritance)
- ④ 메시지(message)

[문제 12] 객체들 간에 메시지를 주고받을 때 각 객체의 세부내용은 알 필요가 없으므로 인터페이스가 단순해지고 데이터와 데이터를 처리하는 함수를 하나로 묶는 것을 의미하는 것은? (이전 기출)

- ① abstraction
- ② class
- ③ encapsulation
- ④ integration

[문제 13] 객체지향기법에서 Encapsulation에 대한 설명으로 옳지 않은 것은? (이전 기출)

- ① 객체 간의 결합도가 높아진다.
- ② 변경 발생 시 오류의 파급효과가 적다.
- ③ 소프트웨어 재사용성이 높아진다.
- ④ 인터페이스가 단순화된다.

[문제 14] 객체지향에서 정보 은닉과 가장 밀접한 관계가 있는 것은? (이전 기출)

- ① Encapsulation
- ② Class
- ③ Method
- ④ Instance

[문제 15] 객체에 대한 설명으로 가장 옳지 않은 것은? (이전 기출)

- ① 객체는 실세계 또는 개념적으로 존재하는 세계의 사물들이다.
- ② 객체는 공통적인 특징을 갖는 클래스들을 모아둔 것이다.
- ③ 객체는 데이터를 가지며 이 데이터의 값을 변경하는 함수를 가지고 있는 경우도 있다.
- ④ 객체들 사이에 통신을 할 때는 메시지를 전송한다.

[문제 16] 객체지향 기법에서 다음 설명에 해당하는 것으로 가장 옳은 것은? (이전 기출)

- 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

[문제 17] 객체에게 어떤 행위를 하도록 지시하는 명령은? (이전 기출)

- ① Class
 - ② Instance
-

③ Object

④ Message

[문제 18] 객체지향 기법에서 객체가 메시지를 받아 실행해야 할 객체의 구체적인 연산을 정의한 것은? (이전 기출)

① Entity

② Method

③ Instance

④ Class

[문제 19] 다음 객체지향 기법에 대한 설명에 해당하는 것은?

메시지에 의해 객체가 연산을 수행하게 될 때 하나의 메시지에 대해 각 객체가 가지고 있는 고유한 방법으로 응답할 수 있는 능력이다. (이전 기출)

① Encapsulation

② Abstraction

③ Inheritance

④ Polymorphism

1-③ 3번은 객체지향 프로그래밍에 대한 설명이다.

2-② 델파이는 객체지향 프로그래밍 언어이다.

3-① 모듈화는 절차지향 프로그램의 특징이다.

객체지향 프로그래밍은 캡슐화, 추상화, 다형성, 정보은닉, 상속의 5가지 특징을 암기해야 한다.

4-① 객체의 모든 부분이 아닌, 관심부분만을 표현하는 것을 추상화라고 한다.

5-④ "객체를 생성하는 틀" 은 클래스에 대한 대표적인 표현이다.

6-② C#, JAVA, C++은 대표적인 객체지향 프로그래밍 언어이고. C는 대표적인 절차적 프로그래밍 언어이다.

7-③

8-③ 스위프트는 객체지향 프로그램 언어이면서 컴파일 언어이다.

9-① DSL은 선언형 프로그래밍의 특수분야 언어이다.

10-④ 4번 Objective-C는 Apple의 OSX, iOS 애플리케이션을 개발하는데 사용하는 객체지향 언어이다.

11-② 객체들을 묶어 사용자 정의 데이터타입을 만드는 데이터 추상화는 클래스에 대한 설명이다.

12-③ 데이터(변수)과 처리(메소드)를 하나로 묶는 것을 캡슐화 라고 한다.

13-① 객체간 결합도가 낮아진다. 필요한 데이터(변수)와 동작(메소드)을 모두 한곳에서 정의하여 처리하므로 다른 객체와 어쩔수 없이 결합해야 할 필요가 줄어든다.

14-① 캡슐화를 통해 정보은닉을 실현할 수 있다. (public, private, protected)

15-② 공통적인 특징을 갖는 클래스를 모아둔 것은 보통 패키지라고 부른다.

16-③ 정보은닉에 대한 설명이다.

17-④ 객체간 통신은 Message를 통해 수행한다.

18-② 객체안에서 동작을 수행하는 요소를 Method라고 한다.

19-④ 오버로딩, 오버라이딩의 기능을 제공하는 다형성에 대한 설명이다.

4.2.4. 라이브러리 활용

표 4-49 응용SW엔지니어링. 프로그래밍 언어 활용, 라이브러리 활용

능력단위요소	수행 준거	
라이브러리 활용	3.1 애플리케이션에 필요한 라이브러리를 검색하고 선택할 수 있다.	
	3.2 애플리케이션 구현을 위해 선택한 라이브러리를 프로그래밍 언어 특성에 맞게 구성 할 수 있다.	
	3.3 선택한 라이브러리를 사용하여 애플리케이션 구현에 적용할 수 있다.	
지식		기술
<ul style="list-style-type: none"> ○ 라이브러리 기능에 대한 이해 ○ 애플리케이션 요구사항에 대한 이해 ○ 라이브러리 적용을 위한 기술(파일 입출력, 데이터입출력, 예외 처리 등)에 대한 이해 		<ul style="list-style-type: none"> ○ 라이브러리 검색 능력 ○ 요구사항 구현에 필요한 라이브러리 선택 능력 ○ 라이브러리 설치 능력 ○ 라이브러리 적용 능력 ○ 코드 검토 능력 ○ 라이브러리를 활용한 프로토타입 작성 능력

※ 학습TIP

각 파트에서 언급되는 표준/외부 라이브러리, 모듈과 패키지등의 용어에 대한 이해가 필요하다.

그리고 import, try~catch~finally 등의 키워드는 간단한 필기 문제로 출제가능성이 높다.

※ 실력 양성 문제

실력 양성 문제

[문제 1] 라이브러리의 설명으로 옳지 않은 것은?

- ① 표준 라이브러리는 프로그래밍 언어가 기본적으로 가지고 있는 라이브러리를 의미한다
- ② 라이브러리란 필요할 때 찾아서 쓸 수 있도록 모듈화 되어 제공되는 프로그램을 말한다
- ③ 각 프로그래밍 언어의 표준 라이브러리는 여러 종류의 모듈과 패키지를 가지며, 표준라이브러리를 이용하면 별도의 파일 설치 없이 날짜와 시간 등의 기능을 이용할 수 있다
- ④ 외부 라이브러리는 누구나 개발하여 설치할 수 있으나 인터넷 등을 이용하여 공유할 수는 없다.

[문제 2] 다음 문장의 빈칸에 들어갈 단어는?

라이브러리는 ()과 ()를 총칭하며, ()이 개별 파일이라면 ()는 파일들을 모아 놓은 폴더라고 볼 수 있다.

- ① 패키지, 컴포넌트
- ② API , 모듈
- ③ 모듈, 패키지
- ④ 모듈, 컴포넌트

[문제 3] 예외처리에서 예외의 원인과 거리가 먼 것은?

- ① 구 라이브러리 사용
- ② 컴퓨터 하드웨어 문제
- ③ 사용자의 입력 실수
- ④ 받아들이기 수 없는 연산

[문제 4] 다음 문장을 설명하는 예외안전성 수준은?

프로세스 처리 중 실패 했을 경우 부작용이나 원 데이터의 손실 없이 **원상복구** 된다.

- ① 실패 투명성 (no throw guarantee)
- ② 강력한 불변의 예외 안전 (Commit or rollback semantics)
- ③ 최소 예외 안전 (no-leak guarantee)
- ④ 예외 보증을 하지 않는 안전 (No exception safety)

[문제 5] 강인한 예외안전성 수준의 순서로 올바른 것은?

(ㄱ) 실패 투명성 (no throw guarantee)
(ㄴ) 최소 예외 안전 (no-leak guarantee)
(ㄷ) 기본 예외 안전 (Basic exception safety)
(ㄹ) 강력한 불변의 예외 안전 (Commit or rollback semantics)
(ㅁ) 예외 보증을 하지 않는 안전 (No exception safety)

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

[문제 6] Java 언어에서 예외처리 시 사용하는 예약어가 아닌것은?

- ① catch
- ② finally
- ③ try
- ④ default

[문제 7] 프로토타입의 설명으로 옳지 않은 것은?

- ① 프로토타입은 '정보시스템의 미완성 버전 또는 중요한 기능들이 포함되어 있는 시스템의 초기모델'이다
- ② 프로토타입 제작 시 라이브러리를 활용보다는 요구사항에 적합하게 직접 구현한다.
- ③ 프로토타입은 사용자의 모든 요구사항이 정확하게 반영할 때까지 계속해서 개선/보완 된다
- ④ 실제로 많은 애플리케이션들이 지속적인 프로토타입의 확장과 보강을 통해 최종 설계가 승인된다

[문제 8] 다음 문장의 빈칸에 들어갈 단어는? (단답형)

() 기반 프로그래밍은 객체지향 프로그래밍의 한 형태의 갈래로 클래스가 없고, 클래스 기반 언어에서 상속을 사용하는 것과는 다르게, 객체를 원형으로 하여 복제의 과정을 통하여 객체의 동작 방식을 다시 사용할 수 있다

[문제 9] 프로토타입 기반과 언어에서 '무(無)에서(ex nihilo)' 객체를 생성하는 방법에 대한 설명중 옳지 않은 것은?

- ① 많은 프로토타입기반 언어에서 기본 객체 프로토타입인 Object 프로토타입이 있는데, 여기에 공통적으로 필요한 메서드가 들어있고, 모든 객체의 마스터 프로토타입으로 사용된다.
- ② 무에서 객체를 창조하는 방식의 유용한 한 가지는 새 객체의 슬롯 이름이 최상위의 Object 객체와 네임스페이스 충돌이 일어나지 않는다는 것이다
- ③ 이런 체계는 새 객체의 특성과 행동을 이미 있던 객체를 참조해야만 지정할 수 있다.
- ④ 무에서 객체를 생성할 수 있는 체계는 이미 있던 프로토타입을 복제하지 않고 아무것도 없는 상태에서 새로운 객체를 생성할 수 있게 해 준다.

[문제 10] 다음 문장의 빈칸에 들어갈 단어는? (단답형)

실행시간에 ()하는 프로토타입 기반의 언어들은 올바른 메서드로 동적 디스패치를 할 수 있거나, 맞는 자료 조각에 이를 때까지 객체에서 프로토타입으로 간단히 위임 포인터를 연속적으로 따라가서 찾을 수 있다

정답

1-④ 외부 라이브러리는 인터넷을 통해 공유된다.

2-③ 폴더라는 표현이 나왔으므로 패키지가 적합하다.

3-① 구 라이브러리를 사용하면 예외처리 전에 컴파일 과정에서 Error 또는 Warning 등을 통해 문제가 발견된다.

4-② 원 데이터의 손실 없는 원상복구는 "Commit or Rollback" 수준에서 가능하다.

5-① default는 Switch ~ case 조건문에서 사용되는 예약어이다.

6-④

7-② 프로토타입은 고객의 요구사항 만족보다는 기능 구현 타당성 검토를 위해 라이브러리를 이용하여 빠르게 개발한다.

8-프로토타입 - 상속이 아닌 복제를 통해 재사용 하는 프로그래밍 방식을 프로토타입 기반 프로그래밍 이라고 한다.

9-③

10-위임(delegation)