

국제인증 개요

National Instruments LabVIEW 국제인증 자격 프로그램은 3 단계로 구성되어 있습니다. 단계별로 취득할 수 있는 자격시험의 종류는 다음과 같습니다:

- Certified LabVIEW Associate Developer (CLAD)
- Certified LabVIEW Developer (CLD)
- Certified LabVIEW Architect (CLA)

이전 단계의 자격증이 있어야 상위 단계의 자격시험에 응시할 자격이 주어집니다.

CLAD 단계에서는 LabVIEW Full Development System 의 주요 기능을 완전히 이해하고 LabVIEW 모듈을 개발, 디버깅, 관리하는데 필요한 지식과 능력을 갖추어야 합니다. 일반적으로 LabVIEW Full Development System 을 대략 6 ~ 9 개월간 사용하면, CLAD 수준의 경험을 쌓을 수 있습니다.

CLD 단계에서는 중대형 LabVIEW 어플리케이션을 개발, 디버깅, 배포, 관리할 수 있는 능력을 갖추어야 합니다. LabVIEW 를 통해 중대형 어플리케이션을 총 12~18 개월 정도 개발한 경험이 있으면, CLD 수준의 전문성을 쌓을 수 있습니다.

CLA 단계에서는 여러 명의 개발자가 참여하는 환경에서 LabVIEW 어플리케이션을 설계할 수 있는 수준의 숙달된 전문성을 갖추어야 합니다. CLA 단계에서는 프로젝트 설계에서 요구하는 사항들을 관리 가능한 LabVIEW 구성요소로 구분할 수 있는 전문 기술 및 소프트웨어 개발 경험뿐만 아니라, 프로젝트 및 설정 관리 도구를 효과적으로 활용하여 전반적인 프로젝트를 파악할 수 있는 경험이 있어야 합니다. LabVIEW 를 통해 중대형 어플리케이션을 총 24 개월 정도 개발한 전문 경험이 있으면, CLA 수준에 이를 수 있습니다.



노트 CLAD 자격증이 있어야 CLD 자격시험을 응시할 수 있습니다.
CLD 자격증이 있어야 CLA 자격시험을 응시할 수 있습니다.
이 응시 자격에 대해 예외 사항은 없습니다.

자격시험 개요

제품: Windows 용 LabVIEW Full Development System 8.0 버전
(LabVIEW Full Development System 에서 사용 가능한 기능에 대한 상세설명은 ni.com/labview/how_to_buy.htm 을 참조)

시험 시간: 1 시간

시험 문항 수: 40 문항

시험 문제 형식: 다지선다형

합격 기준: 정답률 70% 이상

이 시험은 VI 와 구성요소의 이름 또는 메뉴 사용 단계에 대한 암기력보다는 어플리케이션에 대한 지식과 이해도를 검증하는데 목적을 두고 있습니다.

시험 도중에 LabVIEW 또는 다른 외부 리소스의 사용이 금지되어 있습니다. 필요에 따라 참고할 수 있도록, 적절한 시험 문항에 *LabVIEW 도움말* 스크린샷이 제공됩니다.

본 시험 내용의 일부라도 복제하거나 재생할 수 없습니다. 이를 어기면, 불합격 처리됩니다. 시험장에서 시험지를 교부 받는 경우에는 제본용 호치키스 철침을 제거(또는 시험지의 일부를 분리)하면 불합격 처리됩니다.

응시 방법

미국 및 유럽: Pearson Vue 시험장에서 CLAD 시험을 볼 수 있습니다. 컴퓨터 기반의 시험이며, 시험을 마친 직후에 결과를 확인할 수 있습니다. 일정 및 상세한 정보는 www.pearsonvue.com/ni 를 참조하십시오.

아시아: 시험지를 사용하며, 채점 및 결과 발표까지 대략 4 주가 소요됩니다. 한국에서의 시험 일정 및 상세 정보를 확인하시려면 한국 NI 교육센터에 전화(02-3451-3400) 또는 이메일(Training.Korea@ni.com)로 연락하십시오.

일반 문의사항 또는 건의사항이 있는 경우, 이메일(Training.Korea@ni.com)로 연락하십시오.

시험 주제

1. LabVIEW 프로그래밍 개념
2. LabVIEW 환경
3. LabVIEW 에서의 소프트웨어 구성
4. 프로그래밍 VI 와 함수
5. 데이터 통신 및 동기화 VI 와 함수
6. VI 서버 VI 와 함수
7. 이벤트 핸들링 VI 와 함수
8. 디자인 패턴
9. SubVI 디자인
10. 디버깅 도구와 기법
11. VI 디자인 및 문서화(스타일) 실습
12. 메모리, 성능, 정확성(determinism)

시험 주제(개요):

주제	소주제
1. LabVIEW 프로그래밍 원리	<ul style="list-style-type: none"> a. 데이터 흐름 b. 다형성
2. LabVIEW 환경	<ul style="list-style-type: none"> a. 프런트패널 윈도우, 블록다이어그램, 커넥터 팬 b. 메뉴와 팔레트 c. 설정 옵션
3. LabVIEW 에서의 소프트웨어 구성	<ul style="list-style-type: none"> a. 프런트패널 윈도우와 블록다이어그램 객체 <ul style="list-style-type: none"> i. 컨트롤, 인디케이터, IO 컨트롤, 참조 번호 ii. 터미널, 상수, 노드 iii. 팔레트, 업데이트 모드, 차트와 그래프 범례 iv. 불리언 객체의 기계적인 동작 v. 프로퍼티 노드 b. 데이터 타입과 데이터 구조 <ul style="list-style-type: none"> i. 숫자형, 문자열, 불리언, 경로 데이터 타입 ii. 배열과 클러스터 데이터 타입 iii. 웨이브폼과 타임스탬프 데이터 타입 iv. 배리언트 데이터 타입 c. 프런트패널 윈도우에서 객체와 데이터 타입 사용하기 <ul style="list-style-type: none"> i. 범위, 포맷, 형, 스케일링 ii. 컨트롤 사용자 정의하기 iii. 타입 정의와 엄격한 타입 정의 d. 프로그램 컨트롤 구조와 데이터 저장 <ul style="list-style-type: none"> i. 루프 구조(For 루프와 While 루프) <ul style="list-style-type: none"> a. 루프 경계 인덱싱 b. 시프트 레지스터 ii. 케이스 구조와 시퀀스 구조 <ul style="list-style-type: none"> a. 플랫폼 시퀀스 구조와 다층 시퀀스 구조 b. 케이스 선택자의 값과 데이터 타입 c. 데이터 전달—터널과 시퀀스 로컬 iii. 이벤트 구조 <ul style="list-style-type: none"> a. 알림자와 필터 이벤트(사용자 인터페이스) b. 컨트롤의 값(신호) 프로퍼티 c. 다이내믹 이벤트와 사용자 이벤트 iv. 수식 노드

Certified LabVIEW Associate Developer (CLAD) 시험준비 가이드

	<ul style="list-style-type: none"> v. 조건적 비활성화 구조와 다이어그램 비활성화 구조 vi. Timed 구조 vii. 로컬, 글로벌, 공유 변수
4. 프로그래밍 VI와 함수	<ul style="list-style-type: none"> a. 숫자형, 불리언, 문자열, 경로, 배리언트 b. 변환, 비교, 변경 c. 배열과 클러스터 d. 타이밍 <ul style="list-style-type: none"> i. 기다림 타이머, 틱 카운트(ms), 날짜/시간 함수 ii. Timed 구조와 관련된 시간 함수 e. ASCII, 2 진, 데이터로그, 저장(.tdm), 웨이브폼, XML, 설정 파일 I/O 포맷 f. 웨이브폼과 웨이브폼 파일 I/O g. 다이내믹 이벤트와 사용자 이벤트
5. 데이터 통신, 동기화	<ul style="list-style-type: none"> a. 로컬, 글로벌, 공유 변수 b. DataSocket c. TCP와 UDP d. 동기화 <ul style="list-style-type: none"> i. 알림자(Notifiers) ii. 큐(Queues) iii. 세마포어(Semaphores)
6. VI 서버	<ul style="list-style-type: none"> a. VI 서버 설정하기 b. 클래스 계층구조, 참조, 프로퍼티 노드, 인보크 노드 c. 다이내믹하게 VI 로드하기
7. 에러 핸들링 VI와 함수	<ul style="list-style-type: none"> a. 에러 클러스터 b. 대화 상자 & 사용자 인터페이스 VI c. 사용자 에러 코드
8. 디자인 패턴	<ul style="list-style-type: none"> a. 단순한 상태 머신 b. 사용자 인터페이스 이벤트 핸들러 c. 큐 메시지 핸들러 d. 생산자/소비자(데이터)와 생산자/소비자(이벤트) e. 기능적인 글로벌 변수
9. SubVI 디자인	<ul style="list-style-type: none"> a. SubVI 생성 방법 b. 커넥터 팬과 연결 타입 c. 다형성 SubVI d. SubVI와 관련된 옵션 e. 에러 핸들링

Certified LabVIEW Associate Developer (CLAD) 시험준비 가이드

<p>10. 디버깅 도구와 기법</p>	<p>a. 디버깅 도구</p> <ul style="list-style-type: none"> i. 에러 리스트 윈도우 ii. 실행 하이라이트 iii. 브레이크포인트와 단계별 실행 iv. 일반 프로브와 사용자 프로브 <p>b. 다양한 환경에서의 디버깅 사례와 기법</p>
<p>11. VI 디자인과 문서화</p>	<p>a. 다음에 대한 정보는 <i>LabVIEW 도움말</i>의 <i>LabVIEW 스타일 체크리스트</i> 항목을 참조하십시오.</p> <ul style="list-style-type: none"> i. 사용자 인터페이스 디자인 및 블록다이어그램 레이아웃 ii. 모듈 및 계층구조 디자인 iii. SubVI 아이콘과 커넥터 팬 레이아웃(표준) iv. VI 프로퍼티 v. VI 문서화하기
<p>12. 메모리, 성능, 정확성</p>	<p>a. 메모리 확인 및 성능 관련 도구</p> <ul style="list-style-type: none"> i. 프로파일 메모리와 성능 ii. 버퍼 할당 보이기 iii. VI 행렬 <p>b. 프로그래밍 사례</p> <ul style="list-style-type: none"> i. 데이터 흐름 강제하기 ii. 사용자 인터페이스 업데이트와 사용자 인터페이스 컨트롤에 대한 응답 iii. 데이터 타입 선택, 강제 변환, 버퍼 할당 iv. 배열, 문자열, 루프 동작 v. 로컬 및 글로벌 변수, 프로퍼티 노드, 참조

CLAD 주제 상세설명

1. LabVIEW 프로그래밍 원리:

a. 데이터 흐름

1. 데이터 흐름 정의하기
2. LabVIEW 에서 데이터 흐름의 중요성 파악하기
3. 블록다이어그램, VI, SubVI 에서 데이터 흐름 강제하는 프로그래밍 사례 확인하기
4. 데이터 흐름을 거스르는 프로그래밍 사례 확인하기

b. 다형성

1. 다형성 정의하기
2. 다형성의 장점 확인하기
3. 다형성 입력을 사용하는 VI 의 데이터 원소 중간값 또는 출력 확인하기

2. LabVIEW 환경:

a. 프런트패널 윈도우, 블록다이어그램, 커넥터 팬

1. VI 에서 프런트패널 윈도우와 블록다이어그램 객체 사이의 관계 및 커넥터 팬을 통한 연결 고리 확인하기
2. 블록다이어그램이 없는 VI 타입 확인하기
3. 커넥터 팬과 아이콘의 목적 확인하기

b. 팔레트

1. 팔레트의 타입과 기능 확인하기

c. 설정 옵션

1. 다음에 대한 설정 옵션의 효과 확인하기
 - a) 프런트패널 윈도우
 - b) 블록다이어그램
 - c) 환경

3. 프런트패널 윈도우와 블록다이어그램 객체, 데이터 타입, 변수, 소프트웨어 구성요소:

a. 프런트패널 윈도우와 블록다이어그램 객체

1. 어플리케이션의 프런트패널 윈도우에 가장 적합한 객체 선택하기
2. 프런트패널 윈도우에 있는 객체와 터미널 사이의 연관성 설명하기
3. 프런트패널 윈도우 상의 객체 또는 블록다이어그램 상수 사이에서 선택하기
4. 그래프와 차트에 대해 가장 적절한 팔레트, 범례, 업데이트 모드 선택하기
5. 불리언 컨트롤에 대해 가장 적절한 기계적인 동작 결정하기
6. 프로퍼티 노드의 적합성 결정하고 적절한 프로퍼티 선택하기

- b. 데이터 타입과 데이터 구조
 - 1. 프런트패널 윈도우와 블록 다이어그램 객체에 대해 가장 적절한 데이터 타입 선택하기
 - 2. 서로 연계된 데이터 아이템을 묶을 때 가장 적절한 방법 선택하기
 - 3. 웨브폼 데이터 타입 설명하기 및 그래프와 차트에 데이터 디스플레이할 때 사용하기
 - 4. 타임스탬프 데이터 타입 설명하기 및 측정 데이터에 대해 타임스탬프 설정할 때 사용하기
 - 5. 배리언트 데이터 타입을 사용하기에 가장 적합한 어플리케이션 확인하기
- c. 프런트패널 윈도우 객체와 데이터 타입 사용하기
 - 1. 데이터 아이템을 표현하기에 가장 적합한 형, 범위, 포맷, 정밀도, 스케일링 결정하기
 - 2. 컨트롤을 사용자 정의해야 할 상황 및 시나리오 확인하고 설명하기
 - 3. 타입 정의와 엄격한 타입 정의 구분하기
 - 4. 타입 정의 또는 엄격한 타입 정의가 유용한 어플리케이션 확인하고 설명하기
 - 5. 데이터 아이템을 표현하는데 타입 정의 또는 엄격한 타입 정의의 필요 여부 확인하기
- d. 프로그램 컨트롤 구조와 데이터 저장
 - 1. 가장 적합한 프로그램 컨트롤 구조 선택하고 적용하기
 - 2. 프로그램 컨트롤 구조에 대한 데이터 저장 매커니즘 선택하고 구현하기
 - 3. 루프 구조 구성요소의 기능 확인하고 설명하기
 - 4. 가장 적절한 루프 구조로 While 루프 또는 For 루프 선택하기
 - 5. 자동 인덱싱 설명하기 및 각 루프 구조 타입에서 인덱싱의 활성화/비활성화 효과 결정하기
 - 6. 특정 횟수만큼 실행이 반복되거나 루프가 종료한 후, 자동 인덱싱을 사용하는 루프의 데이터 값 결정하기
 - 7. 데이터 저장 원소로서 시프트 레지스터의 용도 및 초기화 설명하기
 - 8. 특정 횟수만큼 실행이 반복되거나 루프가 종료한 후, 시프트 레지스터의 데이터 값 결정하기
 - 9. 시퀀스 구조와 케이스 구조의 장단점 확인하고, 둘 사이에서 선택하기
 - 10. 데이터 흐름 및 데이터 전달 관점에서 플랫폼 시퀀스 구조와 다층 시퀀스 구조의 장단점 확인하기
 - 11. 케이스 구조의 선택자 터미널에 연결하기에 가장 적절한 데이터 타입 선택하기
 - 12. 케이스 구조의 두 출력 터널 타입 확인하고, 각 타입의 장단점 확인하기
 - 13. 이벤트 기반의 프로그래밍에서 이벤트 구조의 장점 확인하기
 - 14. 이벤트 구조의 구성요소 확인하기
 - 15. 이벤트를 생성할 수 있는 다양한 방법 확인하기

16. 이벤트 구조가 처리할 수 있는 다양한 이벤트 확인하기
17. 사용자 인터페이스 이벤트의 두 가지 타입 확인하기
18. 사용자 인터페이스 이벤트에 대해 프런트패널 윈도우를 잠글 때의 효과 이해하기
19. 어플리케이션에서 다이내믹 이벤트가 필요한 경우 확인하고 설명하기
20. 다이내믹 이벤트를 등록 및 등록 취소하는 기법 적용하기
21. 어플리케이션에서 사용자 이벤트가 필요한 경우 확인하고 설명하기
22. 사용자 이벤트를 등록, 생성, 제거하는 기법 적용하기
23. 어플리케이션에 대해 가장 적절한 이벤트 매커니즘 결정하기
24. 식 노드의 구성요소 확인하고 스크립트 변수와 입출력 터미널 사이의 관계 확인하기
25. 식 노드를 사용하는 어플리케이션의 출력 결정하기
26. 조건적 비활성화 구조와 다이어그램 비활성화 구조의 차이점 확인하기
27. 조건적 비활성화 구조 또는 다이어그램 비활성화 구조를 사용하는 어플리케이션의 출력 결정하기
28. 조건적 비활성화 구조 또는 다이어그램 비활성화 구조가 어플리케이션에서 가장 적합한지 결정하기
29. 가장 적절한 Timed 구조 선택하기
30. 서로 다른 Timed 구조 구성요소의 입력과 출력 확인하고 설정하기
31. Timed 구조의 우선순위 설정하기
32. 우선 순위가 설정된 여러 Timed 구조로 구성된 어플리케이션의 출력 결정하기
33. 공유, 로컬, 글로벌 변수 사이의 차이점 확인하기

4. 프로그래밍 VI, 함수, 프로퍼티

a. VI와 익스프레스 VI

1. 다음 리스트의 VI와 함수를 사용하는 어플리케이션 안에 있는 데이터 원소의 출력 또는 중간값 결정하기
2. 다음 리스트의 VI와 함수 사용하기, 특정 기능을 완성하는데 가장 적절한 VI 또는 함수 결정하기

이 섹션에 적용되는 VI와 함수의 리스트:

- a) 숫자형-숫자형, 변환, 데이터 조작, 비교 팔레트
- b) 불리언-불리언 팔레트
- c) 문자열-문자열, 문자열/숫자 변환, 문자열/배열.경로 변환 팔레트
- d) 경로-파일 I/O 팔레트에 있는 경로 함수
- e) 배리언트-클러스터&배리언트 팔레트에 있는 배리언트 함수
- f) 배열-배열 팔레트
- g) 클러스터-클러스터&배리언트 팔레트에 있는 클러스터 함수
- h) 타이밍-타이밍 및 Timed 구조 팔레트

- i) 파일 I/O-파일 I/O 및 XML 팔레트
- j) 웨이브폼-웨이브폼 팔레트
- k) 이벤트-이벤트 팔레트

5. 데이터 통신 및 동기화 VI 와 함수

a. 함수, VI, 익스프레스 VI

1. 어플리케이션 사이에서 데이터를 전송하는 가장 적절한 방법 선택하기
2. 데이터 통신을 위한 로컬, 글로벌, 공유 변수의 장단점 확인하고 사용하기
3. 다양한 TCP 와 UDP 를 사용한 통신 방법 확인하고 설명하기
4. 알림자와 큐 사이의 차이 확인하고 설명하기
5. 여러 블록 다이어그램 섹션 사이에서 데이터를 전송하는 가장 적절한 방법 선택하기
6. 데이터 보호 및 동기화를 위해 세마포어를 사용하는 어플리케이션 확인하기
7. 다음 리스트의 VI 와 함수를 사용하는 어플리케이션 안에 있는 데이터 원소의 출력 또는 중간값 결정하기
8. 다음 리스트의 VI 와 함수 사용하기, 특정 기능을 완성하는데 필요한 가장 적절한 VI 또는 함수 결정하기

이 섹션에 적용되는 VI 와 함수의 리스트:

- a) DataSocket-DataSocket 팔레트
- b) TCP 와 UDP-TCP 와 UDP 팔레트
- c) 알림자-알림자 동작 팔레트
- d) 큐-큐 동작 팔레트
- e) 세마포어-세마포어 팔레트

6. VI 서버

a. 설정

1. VI 서버를 설정할 때 적절한 셋팅 적용하기

b. 클래스 계층구조, 참조, 프로퍼티 노트, 인보크 노트, VI 다이내믹하게 로드하기

1. VI 를 다이내믹하게 로드하고 실행하는 다양한 메소드 및 메소드를 지원하는 참조 타입, 프로퍼티, 메소드 확인하기
2. 주어진 클래스 계층구조에 대해 프로퍼티와 메소드 상속을 이해하고, VI 가 계층구조에서 상위 또는 하위 클래스에 대한 참조를 가져오도록 만드는 참조 타입 사용하기
3. VI 를 다이내믹하게 로드하고 실행하는데 가장 적절한 메소드 결정하기

7. 이벤트 핸들링 VI 와 함수

- a. 에러 클러스터와 와이어
 - 1. 에러 와이어를 받는 에러 클러스터와 터미널의 구성요소 확인하기
 - 2. 에러와 경고의 차이점 확인하기
 - 3. *LabVIEW 도움말의 LabVIEW Style Checklist* 토픽에 부합하는 VI 디자인하기 (예를 들어, While 루프의 컨트롤, Case 구조의 에러 핸들링, 커넥터 팬에서의 적절한 터미널 사용을 위해 에러 확인하기)
- b. 에러 및 대화 상자 VI
 - 1. 주어진 VI 또는 SubVI 에서 에러를 처리하거나, 사용자에게 알리거나, 에러가 있는 VI 를 호출하기에 가장 적합한 위치 확인하기
 - 2. **대화 상자 & 사용자 인터페이스** 팔레트의 VI 와 함수를 활용하여 지정한 에러 핸들링 및 리포트 기능을 완성하는데 가장 적합한 VI 또는 함수 결정하기
- c. 사용자 에러 코드
 - 1. 사용자 에러 코드를 정의하는 방법과 숫자형 범위를 확인하고, 사용자 에러 코드를 사용하여 VI 에서 에러 생성하기

8. 디자인 패턴

- a. 디자인 패턴 선택하기:
 - 1. 디자인 패턴 확인하고, 장단점 설명하고, 다른 디자인 패턴과 비교하기
 - 2. 주어진 어플리케이션 요구조건에 대해, 다음 중에서 가장 적절한 디자인 패턴 선택하기
 - a) 단순한 상태 머신
 - b) 사용자 인터페이스 이벤트 핸들러
 - c) 큐 메시지 핸들러
 - d) 생산자/소비자(데이터)
 - e) 생산자/소비자(이벤트)
 - f) 기능적인 글로벌 변수

9. SubVI 디자인

- a. SubVI 생성 방법
 - 1. SubVI 생성 방법과 그 장단점 확인하고 설명하기
- b. 커넥터 팬과 연결 타입
 - 1. *LabVIEW 도움말의 LabVIEW Style Checklist* 토픽에 따라 가장 적절한 커넥터 팬 선택하고 터미널 할당하기
 - 2. 필수, 권장, 옵션 터미널 확인하기

3. 주어진 요구조건을 기반으로, 필수, 권장, 옵션으로 연결이 설정된 터미널 확인하기
- c. 다형성 SubVI
 1. 다형성 SubVI 디자인이 가장 적합한 디자인인지 평가하기
 2. 다형성 SubVI 을 개발할 때의 장단점과 제한사항 확인하기
- d. SubVI 와 관련된 옵션
 1. 실행 및 윈도우 셋팅 확인하고, 각 셋팅의 의미 설명하기
 2. SubVI 의 인스턴스 전체 또는 단일 인스턴스에 대해 가장 적절한 실행 및 윈도우 셋팅 선택하고 적용시키기
- e. 에러 핸들링
 1. *LabVIEW 도움말의 LabVIEW Style Checklist* 토픽에 따라 SubVI 에 대한 에러 핸들링 적용시키기

10. 디버깅 도구와 기법

- a. 디버깅 도구
 1. LabVIEW 의 에러 핸들링 및 에러 경고 방법을 결정하는 VI 프로퍼티 셋팅 확인하고, 그 의미를 설명하기
 2. 깨진 **실행** 버튼이 나타나는 VI 의 에러를 확인하고, **에러 리스트** 윈도우를 사용하여 에러의 원인 찾기
 3. 데이터 흐름을 추적할 때 및 다른 도구와 함께 디버깅을 할 때 실행 하이라이트의 용도 설명하기
 4. VI 및 SubVI 를 디버깅할 때 브레이크포인트, 실행 지연, 단계별 실행 도구의 용도 설명하고 적용시키기
 5. 데이터 값을 디스플레이할 때 프로브 도구, 인디케이터, 일반 프로브, 조건적 프로브, 사용자 프로브 활용하기
- b. 다양한 환경에서의 디버깅 사례와 기법
 1. 주어진 에러 조건에 대해, 에러를 디버깅하는 가장 적절한 방법 선택하기
 2. 주어진 블록다이어그램을 실행하면 에러 조건이 발생할지 구분하기

11. VI 디자인과 문서화

- a. 다음에 대해 LabVIEW 도움말의 LabVIEW Style Checklist 토픽 활용하고 적용시키기:
 1. 사용자 인터페이스 디자인 및 블록다이어그램 레이아웃
 2. 모듈 및 계층구조 디자인
 3. SubVI 아이콘과 커넥터 팬 레이아웃(표준)
 4. VI 프로퍼티
 5. VI 문서화하기

12. 메모리, 성능, 정확성

a. 메모리 확인 및 성능 관련 도구

1. 메모리 및 성능 이슈를 확인하는데 가장 적절한 도구 선택하기

b. 프로그래밍 사례

1. 데이터 흐름을 깨뜨리는 블록다이어그램 코드 구분하기 및 데이터 흐름을 강제하는 기법 확인하기
2. 사용자 인터페이스 응답을 지연시키는 블록다이어그램 코드 구분하기 및 응답 속도를 향상하는 기법 확인하고 업데이트하기
3. 강제 변환을 제한하고, 버퍼링 할당하고, 속도 및 메모리 재사용을 최적화하는데 가장 적절한 데이터 타입 선택하기
4. 메모리와 성능을 제한하는 배열, 문자열, 루프 동작을 확인하고, 성능을 최적화하는 기법과 방법 확인하기
5. 로컬 및 글로벌 변수, 프로퍼티 노드, 참조를 사용할 때 동반될 수 있는 경합 조건, 메모리 및 성능 이슈 확인하고 이를 최적화할 수 있는 기법 사용하기

CLAD 자격 시험 준비를 위한 참조문헌

자격 시험을 준비할 때 사용할 수 있는 참조문헌은 다음과 같습니다:

CLAD 대비 교육과정:

- [NI 국제인증 자격시험 설명회](#) (교육센터에서 실시하는 무료세미나)

CLAD Sample Exam:

- [CLAD Sample Exam](#)

교육과정/길라잡이:

- [Online LabVIEW Basics](#)
- [LabVIEW Introduction Course – Three Hours](#)
- [LabVIEW Introduction Course – Six Hours](#)
- [National Instruments LabVIEW Basics I](#) 및 [LabVIEW Basics II](#) 교육과정:
 - 강사 수업
 - 교육과정 매뉴얼 개인 학습
- [National Instruments LabVIEW Intermediate I](#) 및 [Intermediate II](#) 교육과정:
 - 강사 수업
 - 교육과정 매뉴얼 개인 학습

웹 리소스:

- [LabVIEW Development Guidelines](#)
- 무료 연습용 [LabVIEW Fundamentals Exam](#)
- [National Instruments Academic Web](#)
- [National Instruments Developer Zone](#)
- [National Instruments Developer \(LabVIEW\) Zone](#)
- [National Instruments LabVIEW Zone](#)
- [National Instruments LabVIEW Support](#)
- [LabVIEW Manuals Online](#) (최신 매뉴얼)
- 무료 길라잡이:
 - [LabVIEW Review](#) (Colorado School of Mines)
 - [LabVIEW Tutorial](#) (University of Sydney)
 - [LabVIEW for Dummies](#)® (Illinois Institute of Technology)
 - [LabVIEW Tutorial](#) (University of Buffalo)
 - [LabVIEW Tutorial Series](#) (University of Western Australia)