

# LabVIEW 準開発者認定試験 (CLAD)

## 認定プログラムと試験の概要

### 認定プログラムの概要

ナショナルインスツルメンツ LabVIEW 認定プログラムは、以下の 3 段階の認定レベルで構成されています。

- LabVIEW 準開発者認定試験 (CLAD)
- LabVIEW 開発者認定試験 (CLD)
- LabVIEW 設計者認定試験 (CLA)

各レベルの認定は、次のレベルの認定を受けるための前提条件になります。

CLAD 認定者は、LabVIEW 開発システムで提供されるコア機能と機能範囲を幅広く、完全に理解しています。さらに、それらの知識を応用して小規模な LabVIEW モジュールの開発、デバッグ、保守を行える能力を保有しています。CLAD 認定者の標準的な経験レベルは、LabVIEW 開発システムの使用経験がおおよそ 6～9 か月です。

CLD 認定者は、中～大規模な LabVIEW アプリケーションの開発、デバッグ、展開、保守の経験者です。CLD 認定者は、中～大規模の LabVIEW アプリケーション開発において累積で 12～18 か月程度の経験を持つプロフェッショナルです。

CLA 認定者は、複数の開発者を組織する環境での LabVIEW アプリケーションの設計に習熟しています。CLA 認定者は、プロジェクトの仕様を管理可能な LabVIEW コンポーネントに細分化するための技術的専門知識とソフトウェア開発経験を保有するだけでなく、プロジェクト管理ツールや構成管理ツールを効果的に活用してプロジェクトの完成を指揮するための経験を持ちます。CLA 認定者は、中～大規模の LabVIEW アプリケーション開発において累積で 24 か月程度の経験を持つプロフェッショナルです。



**メモ:** CLAD 認定は CLD 試験を受けるための前提条件です。

CLD 認定は CLA 試験を受けるための前提条件です。

それぞれの試験において、この要件は例外なく適用されます。

# LabVIEW 準開発者認定試験 (CLAD)

## 認定プログラムと試験の概要

### 試験の概要

対象製品: LabVIEW 8.0 開発システム Windows 用 (LabVIEW 開発システムで提供される機能の詳細については、[ni.com/labview/how\\_to\\_buy.htm](http://ni.com/labview/how_to_buy.htm) を参照してください)。

試験時間: 1 時間

出題数: 40

出題形式: 多肢選択式

合格基準: 70%

試験は、アプリケーションの知識を確認するものであり、メニューのステップや VI、コンポーネントの名前の記憶を問うものではありません。

試験中に LabVIEW や他の外部資料を使用することは一切禁止されています。参考用として、または必要に応じて、『LabVIEW ヘルプ』のスクリーンショットが試験中に提供されます。

試験の完全性を維持するため、試験問題のコピーや複製は禁止されています。規定に違反があると不合格となります。ペーパー方式で試験が行われる地域では、ホッチキスを取り外すと不合格となり、評価を受けられません。

### お申し込みの流れ

米国および欧州: CLAD 試験は Pearson Vue 試験センターで受験できます。試験はコンピュータ方式で行われ、試験結果は試験の終了直後にご確認いただけます。詳細および日程については、[www.pearsonvue.com](http://www.pearsonvue.com) を参照してください。

日本: 試験はオンライン方式、またはマークシート方式で実施いたします。試験結果はオンライン方式では終了直後にご確認いただけます。マークシート方式は 2 週間以内にご連絡いたします。詳細および日程については、最寄りのナショナルインスツルメンツ営業窓口までお問い合わせください。

試験全般に関するご質問は E メールをご利用ください。  
[certification@ni.com](mailto:certification@ni.com).

LabVIEW 準開発者認定試験 (CLAD)  
認定プログラムと試験の概要

試験内容

1. LabVIEW プログラミングの概念
2. LabVIEW 環境
3. LabVIEW でのソフトウェア構築
4. プログラミング VI と関数
5. データ通信と同期 VI、同期関数
6. VI サーバ VI、VI サーバ関数
7. エラー処理 VI、エラー処理関数
8. デザインパターン
9. サブ VI の設計
10. デバッグのツールとテクニック
11. VI の設計と文書化 (スタイル) の手法
12. メモリ、パフォーマンス、確定性

# LabVIEW 準開発者認定試験 (CLAD)

## 認定プログラムと試験の概要

### 試験内容 (概要)

テーマ	項目
1. LabVIEW プログラミング手法	<ul style="list-style-type: none"><li>a. データフロー</li><li>b. 多態性</li></ul>
2. LabVIEW 環境	<ul style="list-style-type: none"><li>a. フロントパネルウィンドウ、ブロックダイアグラム、コネクタペーン</li><li>b. メニューとパレット</li><li>c. 構成オプション</li></ul>
3. LabVIEW でのソフトウェア構築	<ul style="list-style-type: none"><li>a. フロントパネルウィンドウとブロックダイアグラムオブジェクト<ul style="list-style-type: none"><li>i. 制御器、表示器、IO 制御器、Refnum</li><li>ii. 端子、定数、ノード</li><li>iii. チャートとグラフのパレット、更新モード、凡例</li><li>iv. ブールオブジェクトの機械的動作</li><li>v. プロパティノード</li></ul></li><li>b. データタイプとデータ構造<ul style="list-style-type: none"><li>i. 数値データタイプ、文字列データタイプ、ブールデータタイプ、パスデータタイプ</li><li>ii. 配列データタイプとクラスタデータタイプ</li><li>iii. 波形データタイプとタイムスタンプデータタイプ</li><li>iv. バリエーションデータタイプ</li></ul></li><li>c. フロントパネルウィンドウでのオブジェクトおよびデータタイプの操作<ul style="list-style-type: none"><li>i. 範囲、形式、表現、スケーリング</li><li>ii. 制御器のカスタマイズ</li><li>iii. タイプ定義と指定タイプ定義</li></ul></li><li>d. プログラム制御ストラクチャとデータストレージ<ul style="list-style-type: none"><li>i. ループ構造 (For ループと While ループ)<ul style="list-style-type: none"><li>a. ループ境界での指標付け</li><li>b. シフトレジスタ</li></ul></li><li>ii. ケースストラクチャとシーケンスストラクチャ<ul style="list-style-type: none"><li>a. フラットシーケンスストラクチャとスタックシーケンスストラクチャ</li><li>b. ケースセレクトの値とデータタイプ</li><li>c. データ受け渡しトンネルとシーケンスローカル</li></ul></li><li>iii. イベントストラクチャ<ul style="list-style-type: none"><li>a. 通知およびフィルタイベント (ユーザインタ</li></ul></li></ul></li></ul>

## LabVIEW 準開発者認定試験 (CLAD)

### 認定プログラムと試験の概要

	<ul style="list-style-type: none"> <li>フェース)</li> <li>b. 制御器の値 (信号) プロパティ</li> <li>c. ダイナミックイベントとユーザイベント</li> <li>iv. フォーミュラノード</li> <li>v. 条件無効ストラクチャとダイアグラム無効ストラクチャ</li> <li>vi. タイミングストラクチャ</li> <li>vii. ローカル変数、グローバル変数、シェア変数</li> </ul>
4. プログラミング VI と関数	<ul style="list-style-type: none"> <li>a. 数値、ブール、文字列、パス、バリエーション</li> <li>b. 変換、比較、操作</li> <li>c. 配列とクラスタ</li> <li>d. タイミング <ul style="list-style-type: none"> <li>i. 待機タイマ、ティックカウント (ms) 関数、日付/時間関数</li> <li>ii. タイミングストラクチャ関連のタイミング関数</li> </ul> </li> <li>e. ASCII 形式、バイナリ形式、データログ形式、ストレージ形式 (.tdm)、波形形式、XML 形式、構成ファイル I/O 形式</li> <li>f. 波形と波形ファイル I/O</li> <li>g. ダイナミックイベントとユーザイベント</li> </ul>
5. データ通信、同期	<ul style="list-style-type: none"> <li>a. ローカル変数、グローバル変数、シェア変数</li> <li>b. DataSocket</li> <li>c. TCP と UDP</li> <li>d. 同期 <ul style="list-style-type: none"> <li>i. ノーティファイア</li> <li>ii. キュー</li> <li>iii. セマフォ</li> </ul> </li> </ul>
6. VI サーバ	<ul style="list-style-type: none"> <li>a. VI サーバの構成</li> <li>b. クラス階層、リファレンス、プロパティノード、インボークノード</li> <li>c. VI の動的ロード</li> </ul>
7. エラー処理 VI、エラー処理関数	<ul style="list-style-type: none"> <li>a. エラークラスタ</li> <li>b. ダイアログ &amp; ユーザインタフェース VI</li> <li>c. カスタムエラーコード</li> </ul>
8. デザインパターン	<ul style="list-style-type: none"> <li>a. シンプルステートマシン</li> <li>b. ユーザインタフェースイベントハンドラ</li> <li>c. キューメッセージハンドラ</li> <li>d. 生産者/消費者 (データ) と生産者/消費者 (イベント)</li> <li>e. 機能的グローバル変数</li> </ul>
9. サブ VI の設計	<ul style="list-style-type: none"> <li>a. サブ VI 作成方式</li> </ul>

## LabVIEW 準開発者認定試験 (CLAD)

### 認定プログラムと試験の概要

	<ul style="list-style-type: none"><li>b. コネクタペーンと接続タイプ</li><li>c. 多態性サブ VI</li><li>d. サブ VI 関連オプション</li><li>e. エラー処理</li></ul>
10. デバッグのツールとテクニック	<ul style="list-style-type: none"><li>a. デバッグツール<ul style="list-style-type: none"><li>i. エラーリストウィンドウ</li><li>ii. 実行のハイライト</li><li>iii. ブレークポイントとシングルステップ</li><li>iv. 一般プローブとカスタムプローブ</li></ul></li><li>b. さまざまな状況のデバッグ手法とテクニック</li></ul>
11. VI の設計と文書化	<ul style="list-style-type: none"><li>a. 以下の項目について、『LabVIEW ヘルプ』の「LabVIEW スタイルチェックリスト」トピックを参照してください。<ul style="list-style-type: none"><li>i. ユーザインタフェースの設計とブロックダイアグラムのレイアウト</li><li>ii. モジュール化と階層設計</li><li>iii. サブ VI のアイコンとコネクタペーンのレイアウト (標準)</li><li>iv. VI プロパティ</li><li>v. VI の文書化</li></ul></li></ul>
12. メモリ、パフォーマンス、確定性	<ul style="list-style-type: none"><li>a. メモリ問題やパフォーマンス問題を特定するためのツール<ul style="list-style-type: none"><li>i. パフォーマンスおよびメモリのプロファイル</li><li>ii. バッファ割り当ての表示</li><li>iii. VI 複雑度測定</li></ul></li><li>b. プログラミング手法<ul style="list-style-type: none"><li>i. データフローの確立</li><li>ii. ユーザインタフェースの更新とユーザインタフェース制御器への応答</li><li>iii. データタイプの選択、強制、バッファ割り当て</li><li>iv. 配列、文字列、ループ動作</li><li>v. ローカル変数、グローバル変数、プロパティノード、リファレンス</li></ul></li></ul>

# LabVIEW 準開発者認定試験 (CLAD)

## 認定プログラムと試験の概要

### CLAD 試験内容の詳細

#### 1. LabVIEW プログラミング手法

##### a. データフロー

1. データフローを定義します。
2. LabVIEW でのデータフローの重要性を特定します。
3. ブロックダイアグラム、VI、サブ VI でのデータフローを確立するプログラム手法を特定します。
4. データフローを遮断するプログラム手法を特定します。

##### b. 多態性

1. 多態性を定義します。
2. 多態性の利点を特定します。
3. 多態性入力を使用する VI でのデータ要素の出力または中間値を識別します。

#### 2. LabVIEW 環境

##### a. フロントパネルウィンドウ、ブロックダイアグラム、コネクタペーン

1. VI のフロントパネルウィンドウとブロックダイアグラムオブジェクトの関係とコネクタペーンによるそれらの接続を特定します。
2. ブロックダイアグラムを持たない VI のタイプを特定します。
3. コネクタペーンとアイコンの目的を特定します。

##### b. パレット

1. パレットのタイプと機能を特定します。

##### c. 構成オプション

1. 以下の項目に対する構成オプションの影響を特定します。
  - a) フロントパネルウィンドウ
  - b) ブロックダイアグラム
  - c) 環境

#### 3. フロントパネルウィンドウとブロックダイアグラムオブジェクト、データタイプ、変数、ソフトウェア構成

##### a. フロントパネルウィンドウとブロックダイアグラムオブジェクト

1. アプリケーションのフロントパネルウィンドウに最も適したオブジェクトを選択します。
2. フロントパネルウィンドウのオブジェクトとその端子の間の接続を説明します。
3. フロントパネルウィンドウのオブジェクトかブロックダイアグラム定数のいずれかを選択します。
4. グラフおよびチャートに最も適したパレット、凡例、更新モードを選択します。
5. ブール制御器に最も適した機械的動作を識別します。

## LabVIEW 準開発者認定試験 (CLAD)

### 認定プログラムと試験の概要

6. プロパティノードの妥当性を識別し、適切なプロパティを選択します。
- b. データタイプとデータ構造
  1. フロントパネルウィンドウとブロックダイアグラムオブジェクトに最も適したデータタイプを選択します。
  2. 関連データ項目のグループ化に最も適したメソッドを選択します。
  3. 波形データタイプについて説明し、このデータタイプを使用してデータをグラフとチャートに表示します。
  4. タイムスタンプデータタイプについて説明し、このデータタイプを使用して計測データにタイムスタンプを付けます。
  5. バリエーションデータタイプが最も適した選択であるアプリケーションを特定します。
- c. フロントパネルウィンドウのオブジェクトとデータタイプの操作
  1. 特定のデータ項目を表すのに最も適した表現、範囲、形式、精度、スケールリングを識別します。
  2. 制御器のカスタマイズが必要になるシナリオを特定し、説明します。
  3. タイプ定義と指定タイプ定義の違いを識別します。
  4. タイプ定義または指定タイプ定義の使用で利点のあるアプリケーションを特定し、説明します。
  5. 特定のデータ項目を表すためにタイプ定義または指定タイプ定義が必要かどうかを識別します。
- d. プログラム制御ストラクチャとデータストレージ
  1. 最も適したプログラム制御ストラクチャを選択し、適用します。
  2. プログラム制御ストラクチャのデータストレージメカニズムを選択し、実装します。
  3. ループ構造コンポーネントの機能を特定し、説明します。
  4. 最も適したループ構造として While ループまたは For ループを選択します。
  5. 自動指標付けについて説明し、それぞれのループ構造のタイプで指標付けをオンまたはオフする効果を特定します。
  6. 自動指標付けを使用するループにおける、設定回数の反復動作後またはループの停止時のデータ値を識別します。
  7. データストレージ要素としてのシフトレジスタの使用と初期化について説明します。
  8. 設定回数の反復動作後またはループ停止時のシフトレジスタのデータ値を識別します。
  9. シーケンスストラクチャとケースストラクチャの利点と欠点を特定し、いずれかを選択します。
  10. データフローとデータの受け渡しに関してフラットシーケンスストラクチャとスタックシーケンスストラクチャの利点と欠点を特定します。
  11. ケースストラクチャのセレクト端子に配線する最も適したデータタイプを選択します。

## LabVIEW 準開発者認定試験 (CLAD)

### 認定プログラムと試験の概要

12. ケースストラクチャにおける 2 種類の出力トンネルを特定し、それぞれのタイプの利点と欠点を特定します。
13. イベント駆動型プログラミングにおけるイベントストラクチャの利点を特定します。
14. イベントストラクチャの構成要素を特定します。
15. イベントが生成される可能のある各種の方法を特定します。
16. イベントストラクチャで処理可能な各種イベントを特定します。
17. ユーザインタフェースイベントの 2 つのタイプを特定します。
18. ユーザインタフェースイベントに対してフロントパネルウィンドウをロックする影響を認識します。
19. アプリケーションでのダイナミックイベントの必要性を特定し、説明します。
20. ダイナミックイベントの登録と登録解除のテクニックを適用します。
21. アプリケーションでのユーザイベントの必要性を特定し、説明します。
22. ユーザイベントの登録、登録解除、破棄のテクニックを適用します。
23. 特定アプリケーションに最も適したイベントメカニズムを識別します。
24. フォーミュラノードの構成要素を特定し、スクリプト変数と出力端子の間の関係を特定します。
25. フォーミュラノードを使用するアプリケーションの出力を識別します。
26. 条件無効ストラクチャとダイアグラム無効ストラクチャの違いを特定します。
27. 条件無効ストラクチャまたはダイアグラム無効ストラクチャを使用するアプリケーションの出力を識別します。
28. 特定のアプリケーションに対して条件無効ストラクチャまたはダイアグラム無効ストラクチャが最も適しているかどうかを識別します。
29. 最も適したタイミングストラクチャを選択します。
30. さまざまなタイミングストラクチャの構成要素の入力と出力と特定し構成します。
31. タイミングストラクチャの優先度を構成します。
32. 優先度が設定された複数のタイミングストラクチャで構成されるアプリケーションの出力を識別します。
33. シェア変数、ローカル変数、グローバル変数の違いを特定します。

#### 4. プログラミング VI、関数、プロパティ

##### a. VI と Express VI

1. 以下の一覧の VI および関数を使用するアプリケーションでのデータ要素の出力または中間値を識別します。
2. 以下の一覧の VI および関数を使用する場合に、指定された機能を完成するために最も適した VI または関数を識別します。

このセクションに適用される VI および関数の一覧

## LabVIEW 準開発者認定試験 (CLAD)

### 認定プログラムと試験の概要

- a) 数値-「数値」パレット、「変換」パレット、「データ操作」パレット、「比較」パレット
- b) ブール-「ブール」パレット
- c) 文字列-「文字列」パレット、「文字列数値変換」パレット、「文字列/配列/パス変換」パレット
- d) パス-「ファイル I/O」パレットのパス関数
- e) バリエント-「クラスタ&バリエント」パレットのバリエント関数
- f) 配列-「配列」パレット
- g) クラスタ-「クラスタ&バリエント」パレットのクラスタ関数
- h) タイミング-「タイミング」パレットと「タイミングストラクチャ」パレット
- i) ファイル I/O-「ファイル I/O」パレットと「XML」パレット
- j) 波形-「波形」パレット
- k) イベント-「イベント」パレット

## 5. データ通信と同期 VI、同期関数

### a. 関数、VI、Express VI

1. アプリケーション間のデータのやり取りに最も適したメソッドを選択します。
2. データ通信にローカル変数、グローバル変数、またはシェア変数を使用する場合のそれぞれの利点と欠点を特定します。
3. TCP と UDP を使用した異なる通信手法を特定し、説明します。
4. ノーティファイアとキューの違いを特定し、説明します。
5. 複数のブロックダイアグラムセクション間のデータのやり取りに最も適したメソッドを選択します。
6. データの保護と同期にセマフォを使用するアプリケーションを特定します。
7. 以下の一覧の VI および関数を使用するアプリケーションでのデータ要素の出力または中間値を識別します。
8. 以下の一覧の VI および関数を使用する場合に、指定された機能の完成に必要なとされる最も適した VI または関数を識別します。

このセクションに適用される VI および関数の一覧

- a) DataSocket-「DataSocket」パレット
- b) TCP と UDP-「TCP」パレットと「UDP」パレット
- c) ノーティファイア-「ノーティファイア操作」パレット
- d) キュー-「キュー操作」パレット
- e) セマフォ-「セマフォ」パレット

## 6. VI サーバ

### a. 構成

1. VI サーバの構成に適した設定を適用します。

## LabVIEW 準開発者認定試験 (CLAD)

### 認定プログラムと試験の概要

- b. クラス階層、リファレンス、プロパティノード、インボークノード、VIの動的ロード
  - 1. VI を動的にロードして実行するさまざまなメソッド、リファレンスタイプ、プロパティ、そのメソッドをサポートするメソッドについて特定します。
  - 2. クラス階層に基づき、プロパティ、メソッド継承を認識し、リファレンスタイプキャスト VI を使用して階層の上位または下位のクラスのリファレンスを取得します。
  - 3. VI を動的にロードして実行する場合に最も適したメソッドを識別します。
  
- 7. エラー処理 VI、エラー処理関数
  - a. エラークラスとエラー配線
    - 1. エラークラスとエラー配線を接続可能な端子の構成要素を特定します。
    - 2. エラーと警告の違いを特定します。
    - 3. 『LabVIEW ヘルプ』の「LabVIEW スタイルチェックリスト」トピックに準拠した VI を設計します。たとえば、エラーチェックングを使用した While ループの制御、ケースストラクチャによるエラー処理、コネクタペーンの適切な端子の使用などです。
  - b. エラーとダイアログ VI
    - 1. 与えられた VI またはサブ VI に基づき、エラーを処理してユーザまたは呼び出し VI にエラーを通知するために最も適した場所を特定します。
    - 2. 「ダイアログ&ユーザインタフェース」パレットの VI および関数を使用する場合に、指定されたエラー処理とレポートの機能を完成するために最も適した VI または関数を識別します。
  - c. カスタムエラーコード
    - 1. カスタムエラーコードを定義するための数値範囲とメソッドを特定します。また、カスタムエラーコードを使用して VI からエラーを生成します。
  
- 8. デザインパターン
  - a. デザインパターンの選択
    - 1. デザインパターンを特定し、その利点と欠点を説明し、そのデザインパターンと他のデザインパターンを比較します。
    - 2. 与えられたアプリケーション要件に基づいて、最も適したデザインパターンを以下から選択します。
      - a) シンプルステートマシン
      - b) ユーザインタフェースイベントハンドラ

## LabVIEW 準開発者認定試験 (CLAD)

### 認定プログラムと試験の概要

- c) キューメッセージハンドラ
- d) 生産者/消費者 (データ)
- e) 生産者/消費者 (イベント)
- f) 機能的グローバル変数

## 9. サブ VI の設計

### a. サブ VI の作成メソッド

1. サブ VI の作成に使用されるメソッドと、それぞれのメソッドの利点と欠点を特定し、説明します。

### b. コネクタペーンと接続タイプ

1. 『LabVIEW ヘルプ』の「LabVIEW スタイルチェックリスト」トピックの推奨に従って、最も適したコネクタペーンを選択し、端子を割り当てます。
2. 端子の種類として、必須、推奨、任意を特定します。
3. 与えられた要件に基づいて、必要、推奨、または任意として設定する端子を特定します。

### c. 多態性サブ VI

1. 多態性サブ VI 設計が最も適した選択かどうかを評価します。
2. 多態性サブ VI の作成における利点、欠点、制約を特定します。

### d. サブ VI 関連オプション

1. 実行設定とウィンドウ設定を特定し、それぞれの設定の意味を説明します。
2. サブ VI のすべてのインスタンスまたは単一のインスタンスに対して最も適した実行設定とウィンドウ設定を選択し、適用します。

### e. エラー処理

1. 『LabVIEW ヘルプ』の「LabVIEW スタイルチェックリスト」トピックの推奨に従ってサブ VI にエラー処理を適用します。

## 10. デバッグのツールとテクニック

### a. デバッグツール

1. LabVIEW のエラー/警告の処理方法を決定する VI プロパティ設定の意味を特定し、説明します。
2. 壊れた「実行」ボタンとなる VI のエラーを特定し、エラーリストウィンドウを使用して原因を識別します。
3. データフローの追跡を目的とし、他のデバッグツールとの組み合わせの補助ツールとして実行のハイライトを使用することについて説明します。
4. VI およびサブ VI のデバッグツールとしてブレークポイント、実行中断、シングルステップを説明し、適用します。
5. プロブツール、表示器、一般プロブ、条件付きプロブ、カスタムプロブを使用してデータ値を表示します。

## LabVIEW 準開発者認定試験 (CLAD)

### 認定プログラムと試験の概要

- b. さまざまな状況のデバッグ手法とテクニック
  - 1. 与えられたエラー状況に基づいて、エラーのデバッグに最も適したメソッドを選択します。
  - 2. 与えられたブロックダイアグラムがエラー状態になるかどうかを識別します。

### 11. VI の設計と文書化

- a. 以下の項目について、『LabVIEW ヘルプ』の「LabVIEW スタイルチェックリスト」トピックを適用します。
  - 1. ユーザインタフェースの設計とブロックダイアグラムのレイアウト
  - 2. モジュール化と階層設計
  - 3. サブ VI のアイコンとコネクタペーンのレイアウト (標準)
  - 4. VI プロパティ
  - 5. VI の文書化

### 12. メモリ、パフォーマンス、確定性

- a. メモリ問題やパフォーマンス問題を特定するためのツール
  - 1. メモリ問題やパフォーマンス問題を特定するために最も適したツールを選択します。
- b. プログラミング手法
  - 1. データフローを遮断するブロックダイアグラムコードと、データフローを確立するテクニックを特定します。
  - 2. ユーザインタフェースの応答や更新の速度を低下させる可能性のあるブロックダイアグラムコードを特定し、応答を向上させるテクニックを特定します。
  - 3. 強制とバッファ割り当てを制限し、速度とメモリ再利用を最適化するために最も適したデータタイプを選択します。
  - 4. メモリとパフォーマンスを制限する配列、文字列、ループ操作を特定し、パフォーマンスを最適化するテクニックとメソッドを特定します。
  - 5. ローカル変数、グローバル変数、プロパティノード、リファレンスの使用に関連した競合状態、メモリとパフォーマンスの問題を特定し、これらの使用を最適化するテクニックを使用します。

# LabVIEW 準開発者認定試験 (CLAD)

## 認定プログラムと試験の概要

### CLAD の準備

#### トレーニング/チュートリアル

- 無料トレーニング教材
  - [Online LabVIEW Basics](#)
  - [LabVIEW Introduction Course – 3 時間](#)
  - [LabVIEW Introduction Course - 6 時間](#)
  
- 有料トレーニング教材
  - [ナショナルインスツルメンツ LabVIEW 初級 I](#) および [LabVIEW 初級 II](#) コース
    - 講師によるトレーニング
    - コーステキストを使用したセルフペーストトレーニング
  - [ナショナルインスツルメンツ LabVIEW 中級 I](#) および [中級 II](#) コース
    - 講師によるトレーニング
    - コーステキストを使用したセルフペーストトレーニング
  - [ナショナルインスツルメンツ LabVIEW 上級コース \(オンライン\)](#)

#### ウェブリソース

- [LabVIEW 開発ガイドライン](#)
- 無料演習 [LabVIEW Fundamentals Exam](#)
- [アカデミック](#)
- [NI Developer Zone](#)
- [NI LabVIEW Zone](#)
- [ナショナルインスツルメンツ LabVIEW サポート](#)
- [LabVIEW マニュアル オンライン](#) (最新マニュアル)