**Лабораторна робота № 2**

Багаторазові повторення і цикли

2.1. Вступ

Середовище LabVIEW містить наступні структури: цикл While (за умовою) і цикл For (з фіксованим числом ітерацій). Цикл While (за умовою) працює до тих пір, поки не виконається логічне умова виходу з циклу. Цикл While аналогічний циклам Do і Repeat Until, використовуваним в текстових мовах програмування.

Цикл While знаходиться в палітрі Functions »Pogramming» Structures. Після того як цикл обраний в палітрі Functions (Функцій), слід за допомогою курсору виділити частину блок-діаграми, яку необхідно помістити в цикл. Після відпускання кнопки миші, виділена область блок-діаграми поміщається в тіло циклу. Додавання об'єктів блок-діаграми в тіло циклу здійснюється приміщенням або перетягуванням об'єктів.

Дані можуть надходити в цикл While (або виходити з нього) через термінали вхідних / вихідних даних циклу. Термінали вхідних / вихідних даних циклу передають дані з структур і в структури. Термінали вхідних / вихідних даних циклу відображаються у вигляді суцільних прямокутників на кордоні області циклу While. Прямокутник приймає колір типу даних, що передаються по терміналу. Дані виходять з циклу по його завершенні. У разі якщо дані надходять в цикл While через термінал вхідних / вихідних даних циклу, виконання циклу починається при надходженні даних в термінал.

Цикл For (з фіксованим числом ітерацій) виконує повторювані операції над потоком даних певну кількість разів.

Цикл For відрізняється від циклу While тим, що завершує роботу, виконавши задане максимальне число ітерацій N. Цикл While завершує роботу при виконанні заданої умови виходу з циклу.

2.2. Створіть ВП, який генерує випадкові числа до тих пір, поки одна з них не виявиться рівним значенню, введеному в елемент управління. При цьому повинно відображатися кількість ітерацій, виконане циклом.

2.2.1. Створіть лицьову панель, розмістивши на ній елементи управління і відображення, як показано на рис. 2.1.

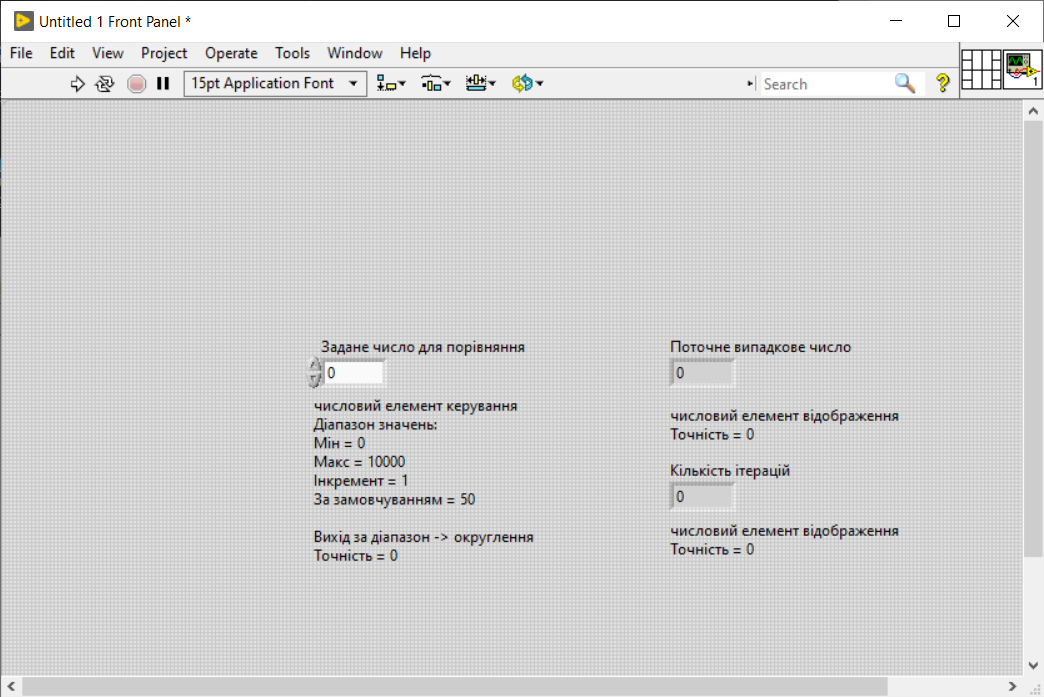


рис.2.1

Створіть блок-діаграму, як показано на рисунку 2.2.

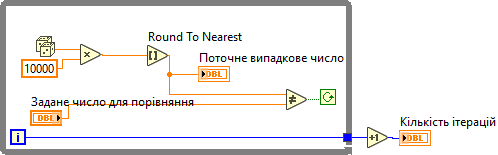


рис.2.2

Помістіть на блок-діаграму функцію Random Number (Генератор випадкових чисел), розташовану на палітрі Функцій в розділі Functions »Programming» Numeric. Ця функція генерує випадкові числа в межах від 0 до 1.

Помістіть на блок-діаграму функцію Multiply (Множення), розташовану в палітрі Функцій в розділі Functions »Programming» Numeric. Ця функція множить поточне значення з виходу функції Random Number (Генератор випадкових чисел) на 10000.

Створіть константу. Для цього слід навести курсор на поле введення даних функції Multiply (Множення), клацнути по ньому правою кнопкою миші і вибрати в контекстному меню пункт Create »Constant. За допомогою інструменту ВВЕДЕННЯ ТЕКСТУ надайте їй значення 10000.

Помістіть на блок-діаграму функцію Round To Nearest (Округлення до найближчого цілого), розташовану в палітрі Функцій в розділі Functions »Programming» Numeric. Ця функція буде округляти отримане в межах від 0 до 10000 випадкове число до найближчого цілого числа.

Помістіть на блок-діаграму функцію Not Equal? (≠), розташовану в палітрі Функцій в розділі Functions »Programming» Comparison. Ця функція призначена для порівняння випадкового числа з числом, введеним в елемент управління Заданий число для порівняння. Якщо значення не рівні, функція видає значення TRUE.

Помістіть на блок-діаграму цикл While, розташований в палітрі Функцій в розділі Functions »Programming» Structures. Наведіть курсор на термінал умови виходу, клацніть по ньому правою кнопкою миші і виберіть пункт Continue if True (Продовження якщо Істина).

Підключіть термінал лічильника ітерацій до кордону області циклу While. На кордоні циклу з'явиться синій прямокутник. Термінал вихідних даних циклу приєднаний до функції збільшення. При виконанні циклу лічильник ітерацій від нього бере зріст дорівнює 1.

Після завершення циклу значення лічильника ітерацій передається на вихід через термінал виходу циклу. Поза тілом циклу значення лічильника ітерацій збільшується на одиницю для відображення кількості виконаних ітерацій.

Помістіть на блок-діаграму функцію Increment (Приріст на 1), розташовану в палітрі Функцій в розділі Functions »Programming» Numeric. Ця функція додає 1 до значення лічильника ітерацій після завершення виконання циклу.

Збережіть ВП під ім'ям Підрахунок ітерацій.vi.

2.2.2. Запуск В.П.

Перейдіть на лицьову панель і змініть значення елемента Заданий число для порівняння. Запустіть ВП. Змініть значення елемента Заданий число для порівняння і запустіть ВП знову. При цьому елемент Поточне випадкове число оновлюється після кожної ітерації циклу, тому що його термінал даних розташований всередині тіла циклу. Значення ж елемента Кількість ітерацій оновлюється після завершення циклу, тому що термінал даних цього елемента розташований поза тілом циклу.

Щоб подивитися, як ВП оновлює значення елементів відображення інформації, необхідно запустити ВП в режимі анімації. Для цього слід натиснути на нструментальной панелі кнопку Highlight Execution, показану зліва. Режим налагодження анімує потік даних, що проходять по блокдіаграмме. Таким чином, є можливість спостерігати зміни значень на кожному етапі їх генерації.

Змініть значення елемента Заданий число для порівняння таким чином, щоб воно зі збільшенням на 1 виходило за встановлений діапазон значень від 0 до 10000.

Запустіть ВП. LabVIEW автоматично призведе нове значення до найближчого значення в зазначеному діапазоні вхідних даних елемента. Закрийте ВП.

2.3. Створіть ВП - найпростіший генератор випадкових чисел і відобразіть результат у вигляді графіка. Зовнішній вигляд лицьової панелі ВП і блок-діаграма представлені на рис.2.3. і рис.2.4 відповідно.

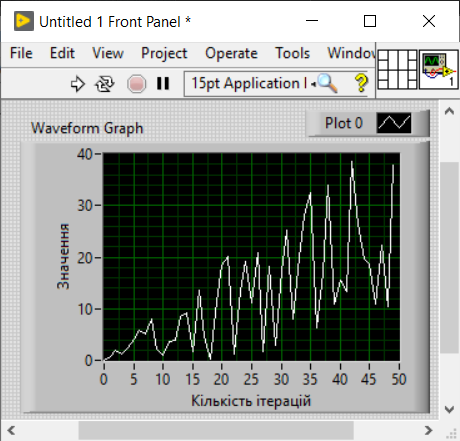


рис.2.3

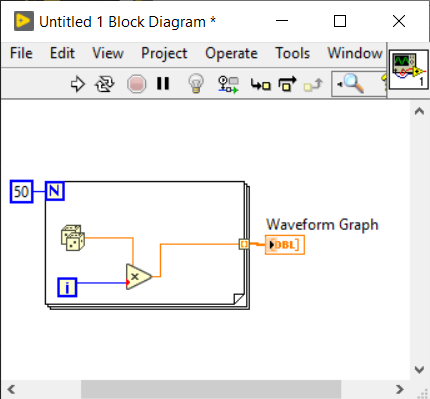


рис.2.4

2.4. Контрольні питання

2.4.1. Цикл While і його застосування в LabVIEW.

2.4.2. Цикл For і його застосування в LabVIEW.