

ФИЗИКО-МАТЕМАТИЧЕСКОЕ И ИНЖЕНЕРНО-ПЕДАГОГИЧЕСКОЕ ОБРАЗОВАНИЕ: ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ РАЗВИТИЯ

1. МЕЖПРЕДМЕТНЫЕ СВЯЗИ ФИЗИКИ И ИНФОРМАТИКИ



ПРИКЛАДНОЕ ИСПОЛЬЗОВАНИЕ ЭЛЕМЕНТА DATA GRID VIEW НА WINDOWS FORMS

Абчинец Диана (УО МГПУ им. И. П. Шамякина, Беларусь)

Научный руководитель – В. В. Давыдовская, канд. физ.-мат. наук, доцент

Windows Forms приложение представляет собой программу, основанную на технологии .NET, которая использует формы для отображения информации пользователю и обработки пользовательского ввода. Форма представляет собой экранный объект, обычно прямоугольной формы, и может быть стандартным диалоговым окном, многодокументным интерфейсом (MDI) или поверхностью для отображения графической информации.

DataGridView является мощным и гибким инструментом для отображения данных в табличном формате. При перетаскивании этого объекта на форму, как и в случае с другими объектами, появляется панель навигации. Она выполняет следующие функции: перемещение по записям, добавление, удаление и сохранение записей [1; 2].

Для настройки пользовательского поведения в приложениях DataGridView может быть расширен несколькими способами. Например, можно программно определить алгоритмы сортировки и создать собственные типы ячеек. Внешний вид легко настраивается путем задания значений его свойств. Он также может работать без привязки к определенному источнику данных, источником могут быть хранилища различных типов [3].

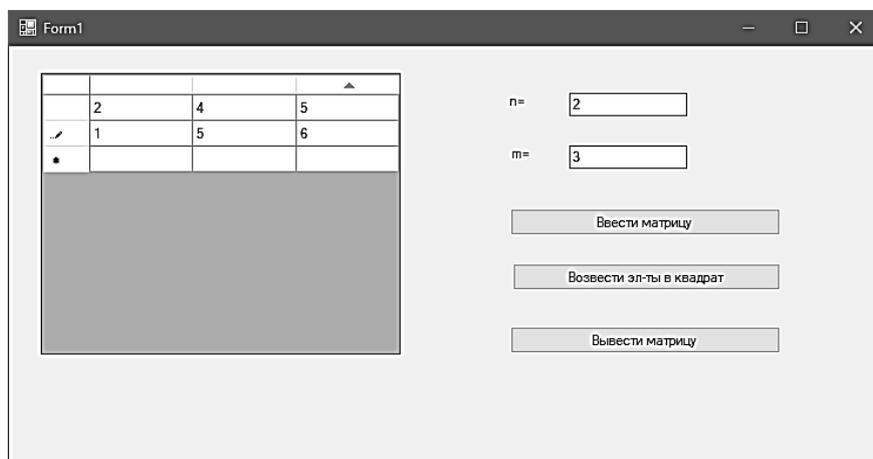


Рисунок 1 – Пример использования DataGridView при обработке матриц на Windows Forms

Рассмотрим пример использования элемента DataGridView на форме при обработке матричных данных на Windows Forms (рисунок 1).

Фрагмент кода приведен на рисунке 2, в данном примере один элемент DataGridView используется как для ввода исходных данных, так и для вывода результата.

```
private void button1_Click(object sender, EventArgs e)
{
    N = Convert.ToInt32(textBox1.Text);
    M = Convert.ToInt32(textBox2.Text);
    dataGridView1.RowCount = N;
    dataGridView1.ColumnCount = M;
    A=new int[N,M];
}

private void button2_Click(object sender, EventArgs e)
{
    for (i = 0; i <= N - 1; i++)
    {
        for (j = 0; j <= M - 1; j++)
            A[i, j] = Convert.ToInt32(dataGridView1[j, i].Value);
    }
    for (i = 0; i <= N-1; i++)
    {
        for (j = 0; j <= M-1; j++)
            A[i, j] = A[i, j] * A[i, j];
    }
}

private void button3_Click(object sender, EventArgs e)
{
    for (i = 0; i <= N-1; i++)
    {
        for (j = 0; j <= M-1; j++)
            dataGridView1[j, i].Value=A[i,j];
    }
}
}
```

Рисунок 2 – Фрагмент кода по работе с элементом DataGridView

Элемент управления DataGridView можно использовать для отображения представлений небольшого объема данных только для чтения либо можно масштабировать его для отображения редактируемого представления очень больших наборов данных.

Список использованной литературы

1. Гросс, К. С. # 2008 и платформа NET 3.5 Framework: вводный курс / К. Гросс. – М. : Вильямс, 2009. – 466 с.
2. Ишкова, Э. А. С#. Начала программирования / Э. А. Ишкова. – М.: Бинум, 2013. – 333 с.
3. # 2008 и платформа .NET 3.5 для профессионалов / К. Нейгел [и др.]. – М. : Диалектика, 2009. – 1738 с.