

Рязанский станкостроительный колледж РГРТУ

Основы алгоритмизации и программирования

Тема 9. Разработка оконного приложения.

Использование списков

Рязань 2022

Оглавление

Компонент ListBox.....	3
Практическая работа №19	6

Компонент `ListBox`

Данный компонент представляет собой список, в котором пользователь может выбирать нужный элемент. Основные свойства компонента приведены в таблице.

Таблица 1. Основные свойства элемента `ListBox`

Свойство	Комментарий
<code>ItemCollection Items</code>	Элементы списка — коллекция строк
<code>object SelectedItem</code>	Возвращает или задает первый элемент в текущем выделении или возвращает значение <i>NULL</i> , если выделение является пустым.
<code>int SelectedIndex</code>	Получение или установка индекса первого элемента в текущем выделенном фрагменте или возврат значения минус один (-1), если выделенный фрагмент пуст.
<code>SelectionMode</code>	Возвращает или задает поведение выбора для списка <i>ListBox</i> . <i>Single</i> - Пользователь может выбрать только один элемент. <i>Multiple</i> - Можно выбрать несколько элементов, не удерживая клавишу <i>Ctrl</i> , <i>Shift</i> или <i>Alt</i> . <i>Extended</i> - Пользователь может выбрать несколько последовательных элементов, удерживая нажатой клавишу <i>SHIFT</i> .
События	Комментарий
<code>SelectionChanged</code>	Возникает при изменении текущего выделения

Как мы видим выше основное свойство списка `Items` — элементы списка, которое в свою очередь является тоже объектом и соответственно содержит свойства и методы. Это означает, что если мы хотим работать со списком, необходимо использовать свойства и методы элемента `Items`, который имеет тип `ItemCollection`.

Чтобы использовать элемент `Items`, рассмотрим класс `ItemCollection`.

Таблица 2. Основные свойства объекта `ItemCollection`

Свойство	Комментарий
<code>int Count</code>	Получает количество записей в коллекции. Доступно только для чтения.
<code>object this[int index]</code>	Возвращает из коллекции элемент по указанному индексу. Используется как индексатор к текущему объекту. Для списка <i>ListBox</i> используется следующим образом (<i>ListBox1.Items[Index]</i>).
Методы	Комментарий
<code>int Add (object newItem)</code>	Добавляет элемент в конец списка. Возвращает номер добавленной строки.
<code>void Clear ()</code>	Очищает список.
<code>bool Contains (object containItem)</code>	Возвращает значение, которое показывает, находится ли указанный элемент в этой коллекции. <i>true</i> , если элемент найден в коллекции; в противном случае <i>false</i> .
<code>void CopyTo (Array array, int index)</code>	Копирует элементы из коллекции в массив, начиная с определенного индекса массива.
<code>int IndexOf (object item)</code>	Возвращает индекс в данной коллекции, в которой расположен указанный элемент. Если элемента нет - значение -1.
<code>void Insert (int</code>	Вставляет элемент в коллекцию по указанному индексу.

insertIndex, object insertItem)	
void Remove (object removeItem)	Удаляет указанный элемент из списка.
void RemoveAt (int removeIndex)	Удаляет элемент с указанным номером.
События	Комментарий
CollectionChanged	Происходит при изменении коллекции.
CurrentChanged	Происходит после изменения текущего элемента
CurrentChanging	Происходит, когда текущий элемент меняется.

Рассмотрим пример использования списка для решения задачи: задано n чисел, найти сумму.

//Добавить запись в список из поля ввода

Ссылка: 1

```
private void btnAdd_Click(object sender, RoutedEventArgs e)
{
    listBox1.Items.Add(tbValue.Text);
}
```

//Удалить выделенную запись

Ссылка: 1

```
private void btnDelete_Click(object sender, RoutedEventArgs e)
{
    listBox1.Items.RemoveAt(listBox1.SelectedIndex);
}
```

//Очистить список

Ссылка: 1

```
private void btnClear_Click(object sender, RoutedEventArgs e)
{
    listBox1.Items.Clear();
}
```

//Заполнить заданным кол. значений

Ссылка: 1

```
private void Заполнить_Click(object sender, RoutedEventArgs e)
{
    //Создаем объект рандом
    Random rnd = new Random();
    //Получаем кол. значений
    int i, n = Convert.ToInt32(tbKolItems.Text);
    //Заполняем в список n случ. значений
    for (i = 1; i <= n; i++)
        listBox1.Items.Add(rnd.Next(200));
}
```

//Расчет по заданию

Ссылка: 1

```
private void btnCalc_Click(object sender, RoutedEventArgs e)
{
    int i, sum;
    sum = 0;
    //Складываем элементы списка
    for (i = 0; i < listBox1.Items.Count; i++)
        sum = sum + Convert.ToInt32(listBox1.Items[i]);
    //Выводим ответ
    tbRez.Text = sum.ToString();
}
```

Практическая работа №19

Использование списков

Начальные условия

1. Размер шрифта 12 пт.
2. Для формирования группы чисел использовать список (ListBox).
3. Заполнять список значениями по одному или указанным количеством случайных значений.
4. Реализовать удаление из списка выбранного значения.
5. Обеспечить защиту полей ввода, от ввода некорректных значений.
6. Обеспечить неизменяемость границ основного окна.
7. Добавить иконку в заголовок программы и исполняемый файл. Заполнить заголовок программы.
8. Заблокировать ввод данных в поле вывода информации.
9. Предусмотреть в программе две кнопки «**Выход**» и «**О программе**», где вывести ФИО разработчика, номер работы и формулировку задания.
10. Оформить программу комментариями

Варианты заданий (Базовый уровень)

1. Ввести n целых чисел. Найти сумму чисел > 5 . Результат вывести на экран.
2. Ввести n целых чисел (>0 или <0). Найти произведение чисел. Результат вывести на экран.
3. Ввести n целых чисел. Найти разницу чисел. Результат вывести на экран.
4. Ввести n целых чисел. Вычислить для чисел > 0 функцию \sqrt{x} . Результат обработки каждого числа вывести на экран.
5. Ввести n целых чисел. Найти произведение чисел < 3 . Результат вывести на экран.
6. Ввести n целых чисел. Найти сумму чисел < 15 . Результат вывести на экран.
7. Ввести n целых чисел. Вычислить для чисел < 0 функцию x^2 . Результат обработки каждого числа вывести на экран.
8. Ввести n целых чисел. Вычислить косинус (\cos) суммы чисел < 3 . Результат вывести на экран.
9. Ввести n целых чисел (>0 или <0). Найти произведение чисел. Результат вывести на экран.
10. Ввести n целых чисел. Вычислить для чисел > 0 функцию \sqrt{x} . Результат обработки каждого числа вывести на экран.
11. Ввести n целых чисел (>0 или <0). Найти разницу чисел. Результат вывести на экран.
12. Ввести n целых чисел. Найти сумму чисел >15 . Результат вывести на экран.
13. Ввести n целых чисел. Найти произведение чисел > 2 . Результат вывести на экран.
14. Ввести n целых чисел. Найти сумму чисел < 8 . Результат вывести на экран.
15. Ввести n целых чисел. Вычислить \sin суммы чисел > 3 . Результат вывести на экран.
16. Найти сумму n целых, четных, случайных чисел, распределенных в диапазоне от 0 до n .
17. Ввести n целых чисел. Найти разницу четных из них чисел. Результат вывести на экран.
18. Ввести n целых чисел. Найти сумму чисел кратных 5. Результат вывести на экран.