

**Рязанский станкостроительный колледж РГРТУ**

## **Основы алгоритмизации и программирования**

**Тема 9. Разработка оконного приложения.**

Управляющие и дополнительные элементы

**Рязань 2022**

## Оглавление

|   |   |
|---|---|
| Меню .....                              | 3 |
| Компоненты меню.....                    | 3 |
| Компонент Menu – главное меню .....     | 3 |
| Компонент MenuItem – элемент меню ..... | 3 |
| Управление событиями в программе.....   | 4 |
| Всплывающие подсказки .....             | 5 |
| Всплывающая подсказка ToolTip .....     | 5 |
| Всплывающая подсказка Popup .....       | 5 |
| Практическая работа №20 .....           | 7 |

## Меню

Список команд по работе с программой, предлагаемых на выбор пользователя - одно из самых старых и универсальных средств организации интерфейса компьютерных программ.

Присутствия меню в главном окне программы, является правилом хорошего тона. Отказываясь от включения меню в проект своей программы, автор игнорирует опыт и навыки пользователей, заставляет их отказываться от стиля работы, к которому они привыкли. Хорошее меню - это не просто список команд, это еще и многофункциональная "шпаргалка" по работе с программой, "шпаргалка", которая всегда под рукой (как известно, читать справочные файлы пользователи не очень любят). Достаточно провести мышью по строке меню вверху экрана, и можно выяснить набор функций программы, комбинации "горячих клавиш".

Теперь - о том, что же можно помещать в меню. **В меню помещаются все команды, которые будут присутствовать в данной программе.** Даже программа, которая ничего не делает, достойна меню как минимум из двух пунктов: *Файл*, и *Справка*, включающий подпункт *О программе* (ведь потенциальные пользователи должны знать, куда отправлять свои денежки!).

Желательна минимальная конфигурация меню. Пункт *Файл*, где находится команда *Выход*, даже если программа непосредственно с файлами не работает, очень желателен, я бы сказал - необходим. Просто *Файл* - настолько привычный элемент меню, что все основные команды (начать новую работу, сохранить результаты, выйти из программы) пользователи ищут прежде всего там. Пункт *Справка*, включающий подпункт *О программе* я рекомендую включать в любое меню по двум причинам: во-первых, пользователь сразу видит, что в программе есть справочная система и есть куда обратиться в случае возникновения затруднений; во-вторых, если в строке меню содержится только один пункт - *Файл*, то это выглядит не очень красиво. Добавление еще одного пункта уравнивает картину.

## Компоненты меню

### Компонент *Menu* – главное меню

Этот компонент представляет собой главное меню программы. Этот компонент обычно размещается в верхней части формы. Обычно на форму помещается один компонент *Menu*.

Разместить заготовку меню можно так, в данном случае с привязкой к верхней части окна:

```
<Menu VerticalAlignment="Top">  
:  
</Menu>
```

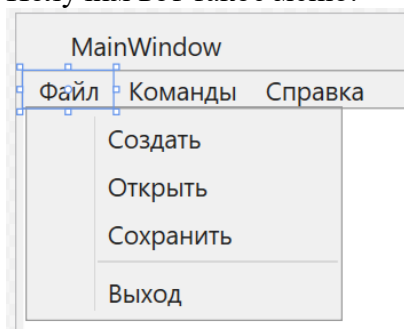
### Компонент *MenuItem* – элемент меню

После добавления к форме программы компонента *Menu* его необходимо настроить. Т.е. добавить элементы меню. Для добавления пунктов меню используется элемент *MenuItem*. Путем использования элемента *MenuItem* и вложения его друг в друга строится иерархическая система меню. **Обратите внимание, что меню строится вручную в окне разметки интерфейса.**

Построим простое меню, используя окно разметки интерфейса:

```
<Menu VerticalAlignment="Top">
  <MenuItem Header="Файл">
    <MenuItem Header="Создать"/>
    <MenuItem Header="Открыть"></MenuItem>
    <MenuItem Header="Сохранить"></MenuItem>
    <Separator></Separator>
    <MenuItem Header="Выход"></MenuItem>
  </MenuItem>
  <MenuItem Header="Команды">
  </MenuItem>
  <MenuItem Header="Справка"></MenuItem>
</Menu>
```

Получим вот такое меню:



Пункты меню можно отделять друг от друга с помощью разделителей. Для этого используйте элемент *Separator*.

```
<MenuItem Header="Сохранить"></MenuItem>
<Separator></Separator>
<MenuItem Header="Выход"></MenuItem>
```

Рассмотрим основные свойства элемента *MenuItem*.

**Таблица 1. Основные свойства элементов меню (MenuItem)**

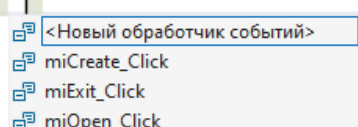
| Свойство                      | Комментарий   |
|-------------------------------|---|
| <code>object</code> Header    | Получает или задает элемент, задающий подпись элементу управления.                    |
| <code>object</code> Icon      | Возвращает или задает значок, появляющийся в элементе меню.                           |
| <code>bool</code> IsCheckable | Вызывает значение показывающее, можно ли отметить флажком элемент меню.               |
| <code>bool</code> IsChecked   | Возвращает или задает значение, показывающее, установлен ли флажок для элемента меню. |
| События                       | Комментарий   |
| <code>Click</code>            | Происходит при нажатии элемента управления  |

### Управление событиями в программе

Для выполнения одного действия в программе может использоваться несколько управляющих элементов, например, действие «**Выход**» может осуществляться через меню «**Выход**» или кнопку «**Выход**», т.е. выход из программы можно осуществить двумя способами. В таком случае необходимо, чтобы оба элемента ссылались на одно событие, т.е. два раза один и тоже код на два обработчика события писать не нужно.

Пример привязки события к двум элементам, смотри ниже. Сначала создаем меню и все события привязываем к элементам меню. После этого уже созданные события привязываем к другим управляющим элементам, в который при создании события можно выбрать уже существующие события:

```
<MenuItem Header="Файл">
  <MenuItem Header="Сохранить"></MenuItem>
  <Separator></Separator>
  <MenuItem Header="Выход" Click="miExit_Click"></MenuItem>
</MenuItem>
</Menu>
<Button Content="Выход" Click="
```

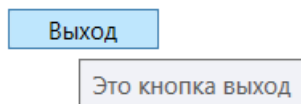


## Всплывающие подсказки

### Всплывающая подсказка *ToolTip*

Элемент *ToolTip* представляет всплывающую подсказку при наведении на какой-нибудь элемент. Для определения всплывающей подсказки у элементов уже есть свойство *ToolTip*, которому можно задать текст, отображаемый при наведении:

```
<Button Content="Выход" ToolTip="Это кнопка выход"/>
```



Всплывающие подсказки можно применять не только кнопкам, но и ко всем другим элементам управления.

Поскольку *ToolTip* является элементом управления содержимого, то в него можно встроить другие элементы для создания более богатой функциональности, например, картинки.

### Всплывающая подсказка *Popup*

Элемент *Popup* также представляет всплывающее окно, только в данном случае оно имеет другую функциональность. Если *ToolTip* отображается автоматически при наведении и также автоматически скрывается через некоторое время, то в случае с *Popup* все эти действия нам надо задавать вручную.

Так, чтобы отразить при наведении мыши на элемент всплывающее окно, нам надо соответственным образом обработать событие *MouseEnter*.

Второй момент, который надо учесть, это установка свойства *StaysOpen="False"*. По умолчанию оно равно *True*, а это значит, что при отображении окна, оно больше не исчезнет, пока мы не установим явно значение этого свойства в *False*.

При потере фокуса подсказка исчезает.

Итак, создадим всплывающее окно:

```

<Button Content="Выход" MouseEnter="Выход_MouseEnter" HorizontalAlignment=
<Popup x:Name="popup1" StaysOpen="False" Placement="Mouse"
AllowsTransparency="True" >
    <TextBlock TextWrapping="Wrap" Background="LightPink" Opacity="0.8" >
        Это кнопка выход
    </TextBlock>
</Popup>

```

И обработчик наведения курсора мыши на кнопку в коде с#:

```

private void Выход_MouseEnter(object sender, MouseEventArgs e)
{
    popup1.IsOpen = true;
}

```

И при наведении указателя мыши на элемент появится всплывающее окно с сообщением.

Выход

Это кнопка выход

## Практическая работа №20

### Программирование циклических алгоритмов (While)

#### Начальные условия

1. Размер шрифта 12 пт.
2. Обеспечить защиту полей ввода, от ввода некорректных значений.
3. Использовать для отображения информации элемент **TextBox**.
4. Указывать диапазон вводимых значений.
5. Реализовать подсказки на элементах интерфейса программы **ToolTip**.
6. Массивы и таблицы не использовать.
7. Обеспечить неизменяемость границ основного окна.
8. Использовать меню **Menu**.
9. Добавить иконку в заголовок программы и исполняемый файл. Заполнить заголовок программы.
10. Заблокировать ввод данных в поле вывода информации.
11. Предусмотреть в программе две кнопки «**Выход**» и «**О программе**», где вывести ФИО разработчика, номер работы и формулировку задания.
12. Оформить программу комментариями.

#### Варианты заданий (Базовый уровень)

1. Вычислить сумму целых случайных чисел, распределенных в диапазоне от 5 до 10, пока эта сумма не превышает некоторого числа  $K$ . Вывести на экран сгенерированные числа, значение суммы, и количество сгенерированных чисел.
2. Найти произведение  $n$  целых случайных чисел  $X$ ., распределенных в диапазоне от 0 до  $n$ . Вывести на экран на одной строке сгенерированные числа, на другой строке результат.
3. Найти сумму  $n$  целых, случайных чисел, распределенных в диапазоне от 0 до  $n$ . Вывести на экран на одной строке сгенерированные числа, на другой строке результат
4. Генерировать случайные числа  $X$ , распределенные в диапазоне от -5 до 5 и вычислять для чисел  $> 0$   $\sqrt{X}$  , а для чисел  $< 0$  функцию  $x^2$ . Вычисления прекратить, когда подряд появится два одинаковых случайных числа. На экран необходимо выводить сгенерированное число и результат расчета функции на разных строках.
5. Вычислить сумму целых случайных чисел, распределенных в диапазоне от 55 до 70, пока эта сумма не превышает некоторого числа  $K$ . Вывести на экран сгенерированные числа, значение суммы, и количество сгенерированных чисел.
6. Найти сумму  $n$  целых, четных, случайных чисел, распределенных в диапазоне от 0 до  $n$ . Вывести на экран на одной строке сгенерированные числа, на другой строке результат.
7. Генерировать случайные числа  $X$ , распределенные в диапазоне от -1 до 6 и вычислять для чисел  $> 0$   $\sqrt{X}$  , а для чисел  $< 0$  функцию  $x^2$ . Вычисления прекратить, когда подряд появится два одинаковых случайных числа. На экран необходимо выводить сгенерированное число и результат расчета функции на разных строках.
8. Найти сумму  $n$  целых, нечетных, случайных чисел, распределенных в диапазоне от 0 до  $n$ . Вывести на экран на одной строке сгенерированные числа, на другой строке результат.
9. Вычислить сумму целых случайных чисел, распределенных в диапазоне от 2 до 10, пока эта сумма не превышает некоторого числа  $K$ . Вывести на экран сгенерированные числа, значение суммы, и количество сгенерированных чисел.
10. Вычислить разницу целых случайных чисел, распределенных в диапазоне от 2 до 10, пока эта разница не станет меньше некоторого числа  $K$  ( $K < 0$ ). Вывести на экран сгенерированные числа, значение суммы, и количество сгенерированных чисел.
11. Генерировать случайные числа  $X$ , распределенные в диапазоне от -4 до 7 и вычислять для чисел  $> 0$   $\sqrt{X}$  , а для чисел  $< 0$  функцию  $x^2$ . Вычисления прекратить, когда подряд

- появится два одинаковых случайных числа. На экран необходимо выводить сгенерированное число и результат расчета функции на разных строках.
12. Найти минимум из  $n$  целых случайных чисел  $X$ , распределенных в диапазоне от 10 до 40. Вывести на экран на одной строке сгенерированные числа, на другой строке результат.
  13. Вычислить сумму целых случайных чисел, распределенных в диапазоне от -7 до 3, пока эта сумма не превышает некоторого числа  $K$ . Вывести на экран сгенерированные числа, значение суммы, и количество сгенерированных чисел.
  14. Генерировать случайные числа  $X$ , распределенные в диапазоне от -5 до 4 и вычислять для чисел  $> 0$   $\sqrt{X}$ , а для чисел  $< 0$  функцию  $x^2$ . Вычисления прекратить, когда подряд появится два одинаковых случайных числа. На экран необходимо выводить сгенерированное число и результат расчета функции на разных строках.
  15. Найти сумму  $n$  целых, случайных чисел (ДСЧ), распределенных в диапазоне от 0 до  $n$  и меньше  $n/2$ . Вывести на экран на одной строке сгенерированные числа, на другой строке результат.
  16. Найти максимум из  $n$  целых случайных чисел  $X$ , распределенных в диапазоне от 0 до  $n$ . Вывести на экран на одной строке сгенерированные числа, на другой строке результат.
  17. Сформировать  $n$  целых чисел в диапазоне 2-14. Найти сумму чисел  $< 8$ . Результат вывести на экран.