Министерство образования Саратовской области

Государственное автономное профессиональное образовательное учреждение Саратовской области

«Новоузенский агротехнологический техникум»

Специальность:09.02.07 Информационные системы и программирование

ОТЧЕТ

по учебной практике профессионального модуля

ПМ 01. Разработка модулей программного обеспечения для компьютерных систем

Студента (ки) группы П-31 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя)

Руководитель практики от техникума Бекбулатов Р.Ж. Мажирин А.А. Тиханин В.Н.

(Ф.И.О.)

Перечень общих компетенций

|  |  |
| --- | --- |
| Код | Наименование общих компетенций |
| ОК 01. | Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. |
| ОК 02. | Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности. |
| ОК 03. | Планировать и реализовывать собственное профессиональное и личностное развитие. |
| ОК 04. | Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. |
| ОК 05. | Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста. |
| ОК 06. | Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения. |
| ОК 07. | Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях. |
| ОК 08. | Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности. |
| ОК 09. | Использовать информационные технологии в профессиональной деятельности. |
| ОК 10. | Пользоваться профессиональной документацией на государственном и иностранном языках. |
| ОК 11. | Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной среде. |

|  |  |
| --- | --- |
| Код | Наименование профессиональных компетенций |
| ПК 1.1 | Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. |
| ПК 1.2 | Разрабатывать программные модули в соответствии с техническим заданием. |
| ПК 1.3 | Выполнять отладку программных модулей с использованием специализированных программных средств. |
| ПК 1.4 | Выполнять тестирование программных модулей. |
| ПК 1.5 | Осуществлять рефакторинг и оптимизацию программного кода. |
| ПК 1.6 | Разрабатывать модули программного обеспечения для мобильных платформ. |

Перечень профессиональных компетенций

Государственное автономное профессиональное образовательное учреждение Саратовской области

«Новоузенский агротехнологический техникум»

АТТЕСТАЦИОННЫЙ ЛИСТ ПО УЧЕБНОЙ ПРАКТИКЕ

ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Разработка модулей программного обеспечения для компьютерных систем

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О. обучающегося)

За время прохождения учебной практики обучающийся подготовлен/не подготовлен (нужное подчеркнуть) к формированию следующих компетенций

**Общие компетенции:**

|  |  |
| --- | --- |
| ОК 01. | Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. |
| ОК 02. | Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности. |
| ОК 03. | Планировать и реализовывать собственное профессиональное и личностное развитие. |
| ОК 04. | Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. |
| ОК 05. | Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста. |
| ОК 06. | Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения. |
| ОК 07. | Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях |
| ОК 08. | Использовать средства физической культуры для сохранения и укрепления  здоровья в процессе профессиональной деятельности и поддержания необходимого  уровня физической подготовленности. |
| ОК 09. | Использовать информационные технологии в профессиональной деятельности. |
| ОК 10. | Пользоваться профессиональной документацией на государственном и иностранном языках. |
| ОК 11. | Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной среде |

**Профессиональные компетенции:**

|  |  |
| --- | --- |
| ПК 1.1 | Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. |
| ПК 1.2 | Разрабатывать программные модули в соответствии с техническим заданием. |
| ПК 1.3 | Выполнять отладку программных модулей с использованием специализированных программных средств. |
| ПК 1.4 | Выполнять тестирование программных модулей. |
| ПК 1.5 | Осуществлять рефакторинг и оптимизацию программного кода. |
| ПК 1.6 | Разрабатывать модули программного обеспечения для мобильных платформ. |

Оценка по результатам практики: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики от техникума Бекбулатов Р.Ж. Мажирин А.А. Тиханин В.Н.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание № 1

На выполнение практического занятия.

По учебной практике ПМ 01. Разработка модулей программного обеспечения для компьютерных систем

Виды работ:Создание интерфейсов посредством визуального проектирования.

Тема: Особенности платформы WPF

Цель: Разработать классическое приложение Windows Presentation Foundation (WPF)

Норма времени: 10 ч.

Место проведения: учебная лаборатория.

Оснащенность рабочего места: ПК, проектор, экран, локальная сеть, программное обеспечение общего и профессионального назначения.

Литература для самоподготовки:

* <https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/getting-started/walkthrough-my-first-wpf-desktop-application?view=netframeworkdesktop-4.8>
* <https://metanit.com/sharp/wpf/1.php>

Задание:

1. Создать автономное приложение для Windows

### Структура проекта

* **Dependencies** - это узел содержит сборки dll, которые добавлены в проект по умолчанию. Эти сборки как раз содержат классы библиотеки .NET, которые будет использовать C#
* **App.xaml** задает ресурсы приложения и ряд конфигурационных настроек в виде кода XAML. В частности, в файле **App.xaml** задается файл окна программы, которое будет открываться при запуске приложения. Если вы откроете этот файл, то можете найти в нем строку **StartupUri="MainWindow.xaml"** - то есть в данном случае, когда мы запустим приложение, будет создаваться интерфейс из файла **MainWindow.xaml**.

**App.xaml.cs** - это файл кода на C#, связанный с файл **App.xaml**, который также позволяет задать ряд общих ресурсов и общую логику для приложения, но в вид кода C#.

* **AssemblyInfo.cs** содержит информацию о создаваемой в процессе компиляции сборке
* **MainWindow.xaml** представляет визуальный интерфейс окна приложения в виде кода XAML.

**MainWindow.xaml.cs** - это файл логики кода на C#, связанный с окном **MainWindow.xaml**.

По умолчанию эти файлы открыти в текстовом редакторе Visual Studio. Причем файл **MainWindow.xaml** имеет два представления: визуальное - в режиме WYSIWIG отображает весь графический интерфейс данного окна приложения, и под ним декларативное объявление интерфейса в XAML. Если мы изменим декларативную разметку, например, определим там кнопку, то эти изменения отображаться в визуальном представлении. Таким образом, мы сможем сразу же получить представление об интерфейсе окна приложения.

Задание № 2

На выполнение практического занятия.

По учебной практике ПМ 01. Разработка модулей программного обеспечения для компьютерных систем

Виды работ:Разработка модуля с использованием текстовых компонентов.

Тема: Текстовые элементы управления

Цель: Отработать создание простого текстового редактора

Норма времени: 10 ч.

Место проведения: учебная лаборатория.

Оснащенность рабочего места: ПК, проектор, экран, локальная сеть, программное обеспечение общего и профессионального назначения.

Литература для самоподготовки:

* <https://www.wpf-tutorial.com/ru/55/элементы-управления-текстом/введение-в-элементы-управления-текстом-wpf>
* <https://metanit.com/sharp/wpf/5.5.php>

Задание:

1. Реализовать следующие объекты в модуле:

### Label

### TextBlock

### RichTextBox

* Главной особенностью меток является поддержка мнемонических команд-клавиш быстрого доступа, которые передают фокус связанному элементу. Например,

|  |  |
| --- | --- |
| 1  2 | <Label Target="{Binding ElementName=TextBox1}">\_привет</Label>  <TextBox Name="TextBox1" Margin="0 30 0 0" Height="30" Width="100"></TextBox> |

* Ключевым свойством здесь является свойство Text, которое задает текстовое содержимое. Причем в случае <TextBlock>Текст1</TextBlock> данное свойство задается неявно. С помощью таких свойств, как FontFamily, TextDecorations и др., мы можем настроить отображение текста. Однако мы можем задать и более сложное форматирование, например:

<TextBlock TextWrapping="Wrap">

    <Run FontSize="20" Foreground="Red" FontWeight="Bold">О</Run>

    <Run FontSize="16" Foreground="LightSeaGreen">негин был, по мненью многих...</Run>

</TextBlock>

* Если для отображения потокового документа в WPF предназначены элементы FlowDocumentPageViewer, FlowDocumentScrollViewer и FlowDocumentReader, то для его редактирования применяется элемент **RichTextBox**.

Чтобы загрузить в RichTextBox файл, мы можем воспользоваться методом XamlReader.Load, который загружает xaml-содержимое. Однако чтобы загрузить иное содержимое, например, файлы rtf или простой текст, нам надо воспользоваться классом **TextRange**, который сам преобразует документ из формата rtf в объект FlowDocument.

Задание № 3

На выполнение практического занятия.

По учебной практике ПМ 01. Разработка модулей программного обеспечения для компьютерных систем

Виды работ:Построение событийно-управляемого интерфейса.

Тема: Элементы управления

Цель: Реализация элементов управления и их свойств

### Норма времени: 10 ч.

Место проведения: учебная лаборатория.

Оснащенность рабочего места: ПК, проектор, экран, локальная сеть, программное обеспечение общего и профессионального назначения.

Литература для самоподготовки:

* <https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/events/how-to-add-an-event-handler-using-code?view=netdesktop-6.0>
* <https://metanit.com/sharp/wpf/6.php>

Задание:

1. Реализовать следующие объекты в модуле:

* Элементы управления содержимым
* Специальные контейнеры
* Декораторы
* Элементы управления списками
* Текстовые элементы управления
* Элементы, основанные на диапазонах значений
* Элементы для работ с датами

1. Все элементы управления наследуются от общего класса **System.Window.Controls.Control** и имеют ряд общих свойств.

### System.Threading.DispatcherObject

В основе WPF лежит модель STA (Single-Thread Affinity), согласно которой за пользовательский интерфейс отвечает один поток. И чтобы пользовательский интерфейс мог взаимодействовать с другими потоками, WPF использует концепцию диспетчера - специального объекта, управляющего обменом сообщениями, через которые взаимодействуют потоки. Наследование типов от класса DispatcherObject позволяет получить доступ к подобному объекту-диспетчеру и и другим функциям по управлению параллелизмом.

### System.Windows.DependencyObject

Наследование от этого класса позволяет взаимодействовать с элементами в приложении через их специальную модель свойств, которые называются свойствами зависимостей (dependency properties). Эта модель упрощает применение ряда особенностей WPF, например, привязки данных. Так, система свойств зависимостей отслеживает зависимости между значениями свойств, автоматически проверяет их и изменяет при изменении зависимости.

### System.Windows.Media.Visual

Класс Visual содержит инструкции, которые отвечают за отрисовку, визуализацию объекта.

### System.Windows.UIElement

Класс UIElement добавляет возможности по компоновке элемента, обработку событий и получение ввода.

### System.Windows.FrameworkElement

Класс FrameworkElement добавляет поддержку привязки данных, анимации, стилий. Также добавляет ряд свойств, связанных с компоновкой (выравнивание, отступы) и ряд других.

Задание № 4

На выполнение практического занятия.

По учебной практике ПМ 01. Разработка модулей программного обеспечения для компьютерных систем

Виды работ:Связывание обработчиков событий с элементами интерфейса.

Тема: Маршрутизация событий

### Цель: Отработать подключение обработчиков событий

Норма времени: 10 ч.

Место проведения: учебная лаборатория.

Оснащенность рабочего места: ПК, проектор, экран, локальная сеть, программное обеспечение общего и профессионального назначения.

Литература для самоподготовки:

* <https://www.wpf-tutorial.com/ru/55/элементы-управления-текстом/введение-в-элементы-управления-текстом-wpf>
* <https://metanit.com/sharp/wpf/5.5.php>

Задание:

1. Реализовать в модуле:

* События клавиатуры
* События мыши
* События стилуса
* События сенсорного экрана/мультитач
* События жизненного цикла

1. Windows Forms, предлагает новую концепцию событий - маршрутизированные события (routed events).

Для элементов управления в WPF определено большое количество событий, которые условно можно разделить на несколько групп:

Обработчик событий можно назначить элементу в Windows Presentation Foundation (WPF) с помощью разметки или кода программной части. Хотя обычно назначается обработчик событий на языке XAML, иногда может потребоваться назначить обработчик событий в коде программной части. Например, код используется, когда:

обработчик событий назначается элементу после страницы разметки с загрузками элемента;

добавляется элемент и ему назначается обработчик событий после страницы разметки с загрузками элемента;

дерево элементов для приложения определяется полностью в коде.

Синтаксис назначения обработчика событий

Язык C# поддерживает назначение обработчика событий с использованием следующих средств:

оператор +=, который также используется в модели обработки событий CLR;

метод [UIElement.AddHandler](https://docs.microsoft.com/ru-RU/dotnet/api/system.windows.uielement.addhandler) ;

Язык VB поддерживает назначение обработчика событий с использованием следующих средств:

Инструкция [AddHandler](https://docs.microsoft.com/ru-RU/dotnet/visual-basic/language-reference/statements/addhandler-statement) с оператором [AddressOf](https://docs.microsoft.com/ru-RU/dotnet/visual-basic/language-reference/operators/addressof-operator), который также используется в модели обработки событий CLR.

Ключевое слово [Handles](https://docs.microsoft.com/ru-RU/dotnet/visual-basic/language-reference/statements/handles-clause) в определении обработчика событий. Дополнительные сведения см. в разделе [Обработка событий в Visual Basic и WPF](https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/events/visual-basic-and-wpf-event-handling?view=netdesktop-6.0).

Метод [UIElement.AddHandler](https://docs.microsoft.com/ru-RU/dotnet/api/system.windows.uielement.addhandler) вместе с оператором AddressOf, чтобы ссылаться на обработчик событий.

Задание № 5

На выполнение практического занятия.

По учебной практике ПМ 01. Разработка модулей программного обеспечения для компьютерных систем

Виды работ:Отладка программы на уровне модуля.

Тема: Отладка и оптимизация модулей инструментальными средствами

### Цель: Отработать запуск и выполнение приложения с подключенным отладчиком

Норма времени: 10 ч.

Место проведения: учебная лаборатория.

Оснащенность рабочего места: ПК, проектор, экран, локальная сеть, программное обеспечение общего и профессионального назначения.

Литература для самоподготовки:

* <https://docs.microsoft.com/ru-ru/visualstudio/get-started/csharp/tutorial-debugger?view=vs-2019>
* <https://metanit.com/sharp/tutorial/1.2.php>

Задание:

1. Реализовать в модуле:

* Запуск отладчика и попадание в точки останова.
* Использование команд для пошагового выполнения кода в отладчике.
* Проверка переменных в подсказках к данным и окнах отладчика.
* Просмотр стека вызовов
  1. Запуск отладчика

Нажмите клавишу F5 (Отладка > Начать отладку) или кнопку Начать отладкуImage of the 'Start Debugging' button. на панели инструментов отладки.

При нажатии клавиши F5 происходит запуск приложения с присоединенным отладчиком. Но пока мы не сделали ничего особенного, чтобы проанализировать код. Поэтому приложение будет просто загружено и появятся указанные ниже выходные данные консоли.

cmdКопировать

Hello, f! Count to 1

Hello, fr! Count to 2

Hello, fre! Count to 3

Hello, fred! Count to 4

Hello, fred ! Count to 5

Hello, fred s! Count to 6

Hello, fred sm! Count to 7

Hello, fred smi! Count to 8

Hello, fred smit! Count to 9

Hello, fred smith! Count to 10

В этом руководстве мы более подробно рассмотрим приложение с отладчиком и познакомимся с возможностями отладчика.

Остановите отладчик, нажав красную кнопку остановки Image of the 'Stop Debugging' button. (SHIFT + F5).

В окне консоли нажмите клавишу, чтобы закрыть его.

Задание № 6

На выполнение практического занятия.

По учебной практике ПМ 01. Разработка модулей программного обеспечения для компьютерных систем

Виды работ:Тестирование с помощью инструментов среды разработки.

Тема: Создание модульных тестов

### Цель: Создать проект модульного теста

Норма времени: 16 ч.

Место проведения: учебная лаборатория.

Оснащенность рабочего места: ПК, проектор, экран, локальная сеть, программное обеспечение общего и профессионального назначения.

Литература для самоподготовки:

* <https://docs.microsoft.com/ru-ru/visualstudio/test/getting-started-with-unit-testing?toc=%2Fvisualstudio%2Fget-started%2Fcsharp%2Ftoc.json&bc=%2Fvisualstudio%2Fget-started%2Fcsharp%2Fbreadcrumb%2Ftoc.json&view=vs-2019&tabs=dotnet%2Cmstest>
* <https://metanit.com/sharp/tutorial/1.2.php>

Задание:

1. Реализовать в модуле:

Создать и запустить модульные тесты

* 1. В целях демонстрации примера модульного теста в этой статье тестируется простой консольный проект C# или C++ "Hello World" с именем HelloWorld. Пример кода для такого проекта выглядит следующим образом:

C#Копировать

namespace HelloWorld

{

public class Program

{

public static void Main()

{

Console.WriteLine("Hello World!");

}

}

}

Выберите узел решения в обозревателе решений. Затем в верхней строке меню выберите Файл>Добавить>Новый проект.

В диалоговом окне нового проекта найдите проект модульного теста, который хотите использовать.

Введите test в поле поиска, чтобы найти шаблон проекта модульного теста для тестовой среды, которую вы хотите использовать, например MSTest (C#) или проект Native Unit Test (C++), и выберите его.

Начиная с Visual Studio 2017 версии 14.8, языки .NET включают встроенные шаблоны для NUnit и xUnit. Для C++ в этом примере выберите проект Собственный модульный тест, в котором используется платформа Microsoft Native Unit Test Framework. (Чтобы использовать другую среду тестирования C++, см. Написание модульных тестов для C/C++). Сведения о настройке тестового проекта на Python см. в статье Настройка модульного тестирования для кода Python.

Задание № 7

На выполнение практического занятия.

По учебной практике ПМ 01. Разработка модулей программного обеспечения для компьютерных систем

Виды работ:Создание модуля доступа к БД.

Тема: Организация доступа к данным

### Цель: Определить строки подключения, вставку данных и выполнение хранимых процедур

Норма времени: 20 ч.

Место проведения: учебная лаборатория.

Оснащенность рабочего места: ПК, проектор, экран, локальная сеть, программное обеспечение общего и профессионального назначения.

Литература для самоподготовки:

* <https://docs.microsoft.com/ru-ru/visualstudio/data-tools/create-a-simple-data-application-with-wpf-and-entity-framework-6?view=vs-2019>
* <https://metanit.com/sharp/wpf/19.2.php>

Задание:

1. Реализовать в модуле:

определение строк подключения, вставка данных и выполнение хранимых процедур

* 1. Настройка образца базы данных

Создайте образец базы данных, выполнив следующие действия:

В Visual Studio откройте окно обозревателя серверов.

Щелкните правой кнопкой мыши подключения к данным и выберите "Создать новую базу данных SQL Server".

В текстовом поле "Имя сервера " введите (localdb)\mssqllocaldb.

В текстовом поле "Создать имя базы данных " введите "Продажи" и нажмите кнопку "ОК".

Пустая база данных Sales создается и добавляется в узел подключений к данным в обозревателе серверов.

Щелкните правой кнопкой мыши подключение к данным "Продажи" и выберите "Создать запрос".

Откроется окно редактора запросов.

Скопируйте скрипт Transact-SQL Sales в буфер обмена.

Вставьте скрипт T-SQL в редактор запросов и нажмите кнопку "Выполнить".

Через некоторое время запрос завершится и будут созданы объекты базы данных. База данных содержит две таблицы: Customer и Orders. Эти таблицы изначально не содержат данных, но вы можете добавлять данные при запуске создаваемого приложения. База данных также содержит четыре простые хранимые процедуры.

Создание форм и добавление элементов управления

Создайте проект для приложения Windows Forms и назовите его SimpleDataApp.

Visual Studio создает проект и несколько файлов, включая пустую форму Windows Forms с именем Form1.

Добавьте две формы Windows Forms в проект, чтобы он включал три формы, и назначьте им следующие имена:

Навигация

NewCustomer

FillOrCancel

Для каждой формы добавьте текстовые поля, кнопки и другие элементы управления, которые отображаются на рисунках ниже. Для каждого элемента управления задайте свойства, указанные в таблицах.

Задание № 8

На выполнение практического занятия.

По учебной практике ПМ 01. Разработка модулей программного обеспечения для компьютерных систем

Виды работ:Создание модуля вывода информации БД на печать

Тема: Печать из базы данных

Цель: Организовать подключение к базе данных и вывод информации на печать

### Норма времени: 24 ч.

Место проведения: учебная лаборатория.

Оснащенность рабочего места: ПК, проектор, экран, локальная сеть, программное обеспечение общего и профессионального назначения.

Литература для самоподготовки:

* <https://metanit.com/sharp/wpf/19.2.php>
* https://professorweb.ru/my/WPF/documents\_WPF/level28/28\_9.php

Задание:

1. Реализовать в модуле:

подключение к базе данных и вывод информации на печать

1. В любом проекте WPF, как и в ряде других типов проектов для .NET, по умолчанию есть файл конфигурации, который называется *app.config* и который имеет следующее содержимое:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <?xml version="1.0" encoding="utf-8" ?>  <configuration>      <startup>          <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6" />      </startup>  </configuration> |

PrintDialog заключает в себе знакомое диалоговое окно Print, которое позволяет пользователю выбрать принтер и установить несколько стандартных настроек печати, такие как количество копий. Однако класс PrintDialog — это нечто больше, чем просто симпатичное окно; он также обладает встроенной способностью инициировать вывод на печать

Чтобы отправить задание на печать посредством класса PrintDialog, необходимо воспользоваться одним из двух описанных ниже методов:

**PrintVisual()**

Работает с любым классом, унаследованным от System.Windows.Media.Visual. Сюда относится любая графика, нарисованная вручную, и любой элемент, помещенный в окно.

**PrintDocument()**

Работает с любым объектом *DocumentPaginator*. Сюда относятся те, что используются для разбиения на страницы FlowDocument (или XpsDocument), и все специальные объекты DocumentPaginator, которые создаются для работы с данными.

В следующих разделах рассматриваются разнообразные стратегии, которые можно применять для создания вывода на печать.

Задание № 9

На выполнение практического занятия.

По учебной практике ПМ 01. Разработка модулей программного обеспечения для компьютерных систем

Виды работ:Разработка модуля многооконного интерфейса

## Тема: Взаимодействие между окнами

Цель: Организовать взаимодействие с несколькими окнами в WPF

### Норма времени: 18 ч.

Место проведения: учебная лаборатория.

Оснащенность рабочего места: ПК, проектор, экран, локальная сеть, программное обеспечение общего и профессионального назначения.

Литература для самоподготовки:

* https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/getting-started/walkthrough-my-first-wpf-desktop-application?view=netframeworkdesktop-4.8
* https://metanit.com/sharp/wpf/20.2.php

Задание:

1. Реализовать в модуле:

взаимодействие с несколькими окнами в WPF

1. Рассмотрим, как мы можем взаимодействовать с несколькими окнами в WPF. Для этого создадим новый проект. По умолчанию он уже содержит одно главное окно MainWindow. Теперь добавим еще одно окно.

Назовем его **TaskWindow**.

Теперь определим на главном окне MainWindow кнопку для открытия нового окна:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <Window x:Class="WindowApp.MainWindow"          xmlns="<http://schemas.microsoft.com/winfx/2006/xaml/presentation>"          xmlns:x="<http://schemas.microsoft.com/winfx/2006/xaml>"          xmlns:d="<http://schemas.microsoft.com/expression/blend/2008>"          xmlns:mc="<http://schemas.openxmlformats.org/markup-compatibility/2006>"          xmlns:local="clr-namespace:WindowApp"          mc:Ignorable="d"          Title="MainWindow" Height="250" Width="300">      <Grid>          <Button Width="100" Height="30" Content="Новое окно" Click="Button\_Click" />      </Grid>  </Window> |

Обработчик нажатия кнопки Button\_Click будет выглядеть так:

|  |  |
| --- | --- |
| 1  2  3  4  5 | private void Button\_Click(object sender, RoutedEventArgs e)  {      TaskWindow taskWindow = new TaskWindow();      taskWindow.Show();  } |

Для открытия нового окна создаем его объект и затем вызываем метод **Show()**.

При нажатии на кнопку открывается окно TaskWindow.

Используя ссылку на окно, мы можем взаимодействовать с ним, например, передавать ему данные из главной формы или вызывать его методы.

Задание № 10

На выполнение практического занятия.

По учебной практике ПМ 01. Разработка модулей программного обеспечения для компьютерных систем

Виды работ:Оформление документации на программные средства с использованием инструментальных средств

## Тема: Автоматизированные средства оформления документации

Цель: Автоматизация разработки технической документации

### Норма времени: 10 ч.

Место проведения: учебная лаборатория.

Оснащенность рабочего места: ПК, проектор, экран, локальная сеть, программное обеспечение общего и профессионального назначения.

Литература для самоподготовки:

* https://skillbox.ru/media/code/kak\_pisat\_tekhnicheskuyu\_dokumentatsiyu/
* <https://metanit.com/sharp/wpf/15.php>
* <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/xmldoc/>
* https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/xmldoc/recommended-tags

Задание:

1. Реализовать в модуле:

документирование программного обеспечения

1. В C# есть два вида комментариев. Однострочные:

int a = 11 + 12; //Это однострочный комментарий

И многострочные:

/\* Начало комментария

Конец комментария \*/

Компилятор во время сборки игнорирует комментарии и просто вырезает их, поэтому на работу программы они не влияют.

Более продвинутый вариант — использовать XML. Чтобы вставить XML-комментарий, нужно перед названием класса, поля, свойства или метода поставить тройной слеш.

После этого автоматически будет создано два элемента:

1. **Summary** — общий комментарий. В нем пишут, что делает метод или для чего нужен класс.
2. **Param** — комментарий об аргументе. В нем указывается, какое значение надо передать.

Практически все инструменты, в том числе и Visual Studio, поддерживают вывод подсказок, которые подгружаются из документации. И теперь, если навести на метод Main () или его аргумент, то можно увидеть, что было написано в комментарии.

Такой способ намного лучше, потому что человеку вообще не нужно ничего открывать, чтобы определить, что делает какой-нибудь фрагмент кода. Конечно, наличие XML создает визуальный шум, но его можно просто скрыть.

Еще можно использовать такие XML-элементы, как:

* **Returns** — возвращаемое значение;
* **Value** — значение свойства;
* **Exception** — исключение;
* **Remarks** — ремарка к основному комментарию.

Таких элементов очень много, подробнее почитать о них можно в [документации Microsoft](https://docs.microsoft.com/ru-ru/dotnet/csharp/codedoc). Цель же этой статьи — показать, как документировать код, чтобы разбираться в проекте стало легче, а не сложнее.

Задание № 11

На выполнение практического занятия.

По учебной практике ПМ 01. Разработка модулей программного обеспечения для компьютерных систем

Виды работ:Оформление отчета по разработке программы

# Тема: Документирование в разработке ПО

Цель: Сформировать отчет по разработанному проекту

### Норма времени: 6 ч.

Место проведения: учебная лаборатория.

Оснащенность рабочего места: ПК, проектор, экран, локальная сеть, программное обеспечение общего и профессионального назначения.

Литература для самоподготовки:

* <https://systems.education/reports>
* https://habr.com/ru/post/215837/

Задание:

1. Реализовать отчет по разработанному проекту
2. При создании отчёта этапы проектирования и разработки последовательно сменяют друг друга.  
     
   На этапе проектирования отчёта системный аналитик:
3. Определяет по техническому заданию (ТЗ) показатели, которые должны выводиться в отчёт
4. Создаёт прототип для формы вывода отчёта и согласовывает его с заинтересованными лицами
5. Описывает последовательность и формулы для расчёта показателей в отчёте
6. Связывает показатели, выводимые в отчёт, с атрибутами модели данных
7. Оформляет постановку на реализацию отчёта

На этапе реализации отчёта разработчик:

1. Берёт постановку на реализацию отчёта в работу
2. Верстает форму для вывода отчёта на экран (или печатную форму отчёта)
3. Создаёт SQL-запросы для выборки нужных данных из базы данных (БД)
4. Реализует расчётные процедуры с полученными данными
5. Выводит полученные результаты в форму вывода