Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Проект**

По дисциплине «Проектирование программ в интеллектуальных системах»

Тема: «Редактор кодов интерфейсов оконных приложений»

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Карагодин Д.Л.

**Проверил:**

Муравьев Г.Л.

Брест 2023

# **Назначение проекта и его функции**

Данный проект был разработан с целью упрощения разработки некоторый графических элементов (Окна-сообщения «MessageBox», Диалоговые окна «DialogBox», Меню главного окна) и внедрения их в проект пользователя. Данный проект использует смешанную интеграцию: часть шаблонов хранится в виде файлов определенных форматов, другая часть хранится в памяти программы. Также следует учитывать, что программа НЕ ИНТЕГРИРУЕТ в проект графические элементы, а ЗАМЕНЯЕТ исходный код проекта в соответствии с желаниями пользователя.

Был разработан данный проект в среде разработки Visual Studio 2022. В качестве языка программирования был использован Visual C++.

Данный проект имеет следующие функции:

* Внедрение окон-сообщений одного типа в главный .cpp файл проекта пользователя и их появление в программе пользователя по следующим возможным событиям: WM\_LBUTTONDOWN (Нажатие ЛКМ), WM\_RBUTTONDOWN (Нажатие ПКМ), WM\_KEYDOWN (Нажатие клавиши на клавиатуре), WM\_PAINT (перерисовка окна).
* Внедрение диалоговых окон одного типа аналогичное внедрению окон-сообщений, но с созданием обработчиков для данных ДО. Также создается ресурсная часть в ресурсный файл необходимая для ДО.
* Внедрение меню с выбором количества пунктов в состав главного окна. Также добавляются обработчики для каждого пункта меню.

Данный проект имеет следующие ограничения:

* Проект пользователя должен быть создан в среде разработки Microsoft Visual C++ 6.0, и пользователь должен создать проект каркаса Simple или Empty с созданным главным .cpp файлом, либо использовать другой пустой проект с наличием внутри главного .cpp файла.
* Если пользователь сам создал проект, то в главный .cpp файл пользователь может вносить сам ТКП, либо оставить файл пустым.
* Ввиду особенностей среды Visual C++ 6.0, в проекте пользователь до начало работы должен создать через среду .rc файл (Resource Script) с именем проекта, иначе результат работы программы будет неполноценным.
* Пользовать должен сам указывать главный файл проекта. Создано данное ограничение, чтобы программа не влезала в другие файлы проекта.

# **Файловая структура проекта**

Структуру файлов и папок в корневом каталоге проекта можно увидеть на рисунке 1.

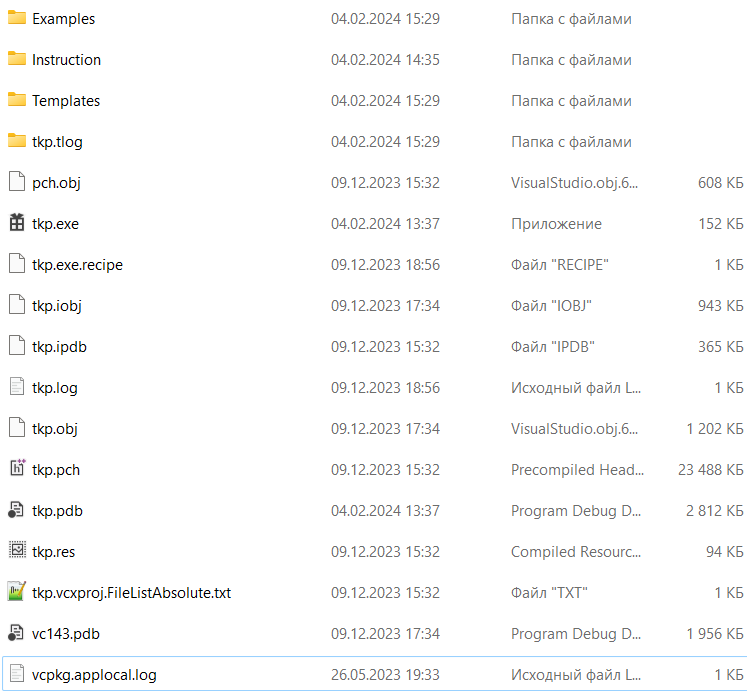


Рис. 1. Структура файлов в корневом каталоге проекта.

Папка *Examples* содержит примеры работы программы:   
1. *SimpleTemplate* – созданный проект на Visual Studio 6.0 без добавления ресурсного файла. Можно использовать для корректной установки программы.  
2. *ModifedProject* – Проект, уже модифицированный программой. Здесь можно просмотреть функциональность программы.

Папка *Instruction* содержит данную инструкцию.

Папка *Template* содержит шаблоны, который пользователь может изменять (на свой страх и риск):   
1. RcTemplate.rc – шаблон, который отвечает за ресурсный файл (данные о графических элеметнах).  
2. ResTemplate.h – шаблон, который отвечает за заголовочный файл ресурсного элемента (идентификаторы графических элементов).  
3. TkpTemplate.cpp – шаблон, который отвечает за изменение главного .cpp файла проекта, выбранный пользователем.

Исполнительный файл *tkp.exe* является самой программой.

Остальные файлы и папки являются системными, и никакого интереса в себе не несут.

# **Установка проекта**

Данный проект размещен по ссылке: <https://github.com/DanikKaragodin/TkpCompilator> . Чтобы скачать данный проект можно использовать следующие способы:

1. (Если не имеется установленный Git на ПК). По ссылке, написанной сверху, пользователь можно скачать проект в виде .zip архива c именем соответствующему названию проекта. Достаточно нажать кнопку Code – Download ZIP. Где находятся и как они выглядят показано на рисунке 2. Данный .zip файл будет находится в системной папке « Загрузки ». Распакуйте любым удобным вам архиватором и ваша программа готова к использованию.

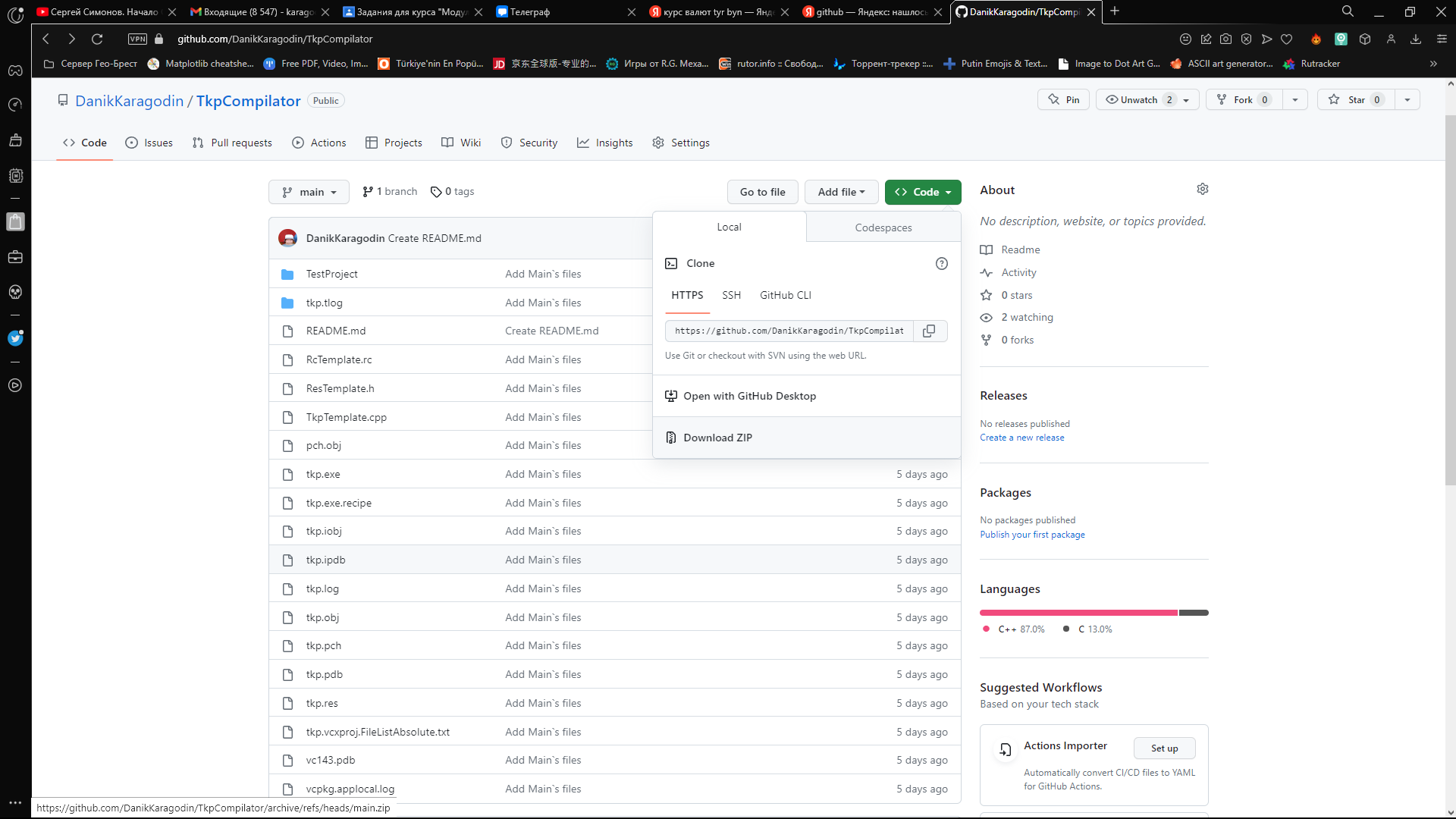


Рис. 2. Местоположение и вид кнопок Code и Download ZIP.

1. (Если имеется установленный Git на ПК). По комбинации клавиш Win+R запуститься окно «Выполнить» (вид окна представлен на рисунке 3). В нем необходимо прописать следующее: git clone https://github.com/DanikKaragodin/TkpCompilator . Далее прожать OK и дождаться окончания загрузки. Данный проект можно будет увидеть в системном каталоге с именем пользователя. Он будет сразу готов к работе.

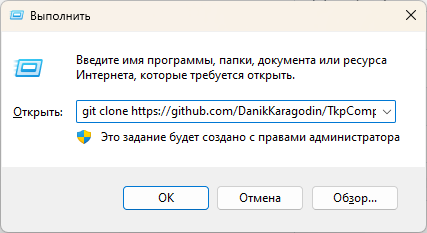


Рис. 3. Вид окна «Выполнить».

1. (С помощью инсталлятора). По ссылке, используемой в третьем пункте, необходимо найти вкладку «Releases» и перейти по ней (Рис. 4). После надо найти и скачать файл с форматом файла \*.exe (Рис. 5). Скачать инсталлятор и следовать его инструкциям (Р ис. 6).

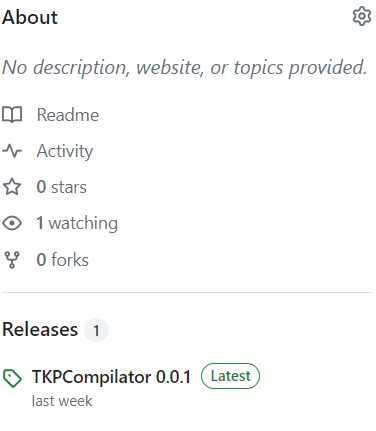


Рис. 4. Блок сайта с вкладкой «Releases».

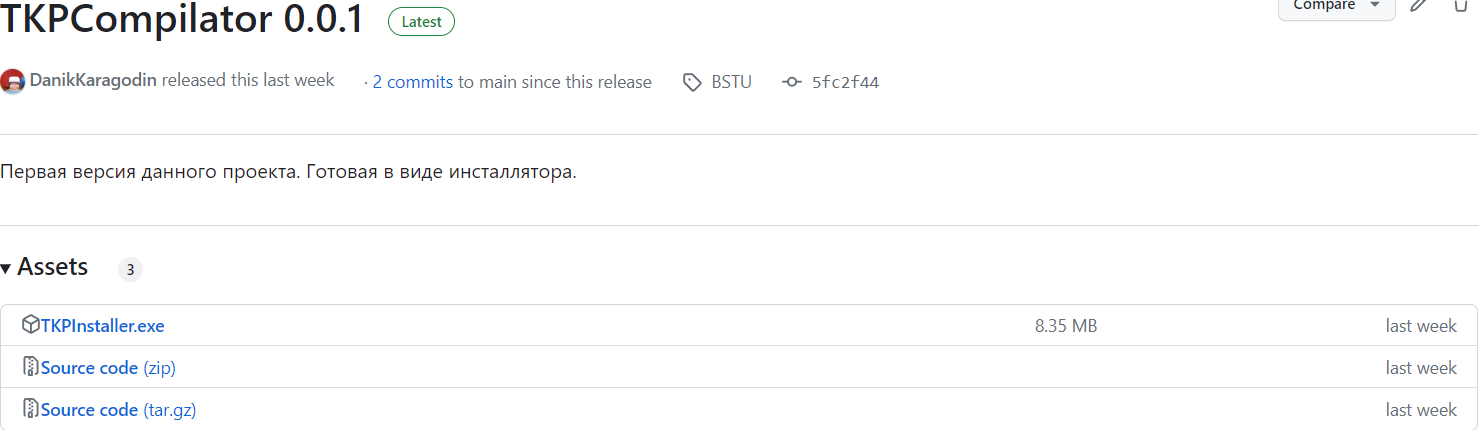


Рис. 5. Вкладка «Releases».

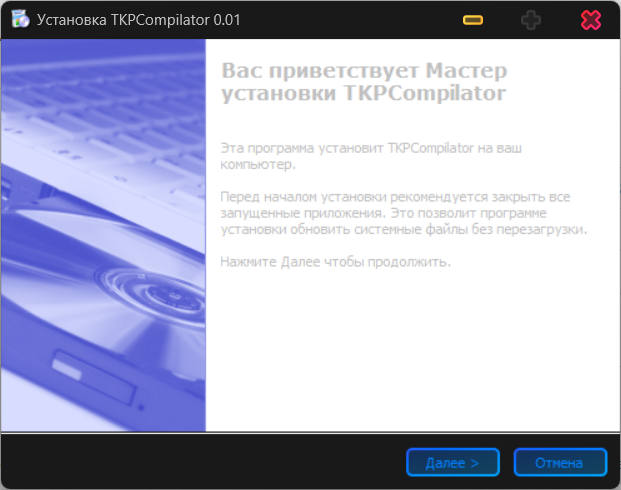


Рис. 6. Инсталлятор программы.

# **Инструкция к использованию**

Необходимо проверить наличие контрольного примера по пути: %путь до корневой папки проекта%/Examples/SimpleTemplate. Если там присутствуют файлы, то всё должно быть хорошо. Пункт 1-2 описываются для тех, у кого контрольный пример отсутствует, или имеет непредвиденные ошибки. Если всё хорошо, то можно переходить сразу к пункту 3.

1. Первым делом необходимо создать пустой проект. К примеру, создать проект с каркасом Simple:

1. Открываем окно создания проектов комбинацией Ctrl + N.
2. Выбираем тип проекта Win32 Application и пишем название для своего проекта (рисунок 7).

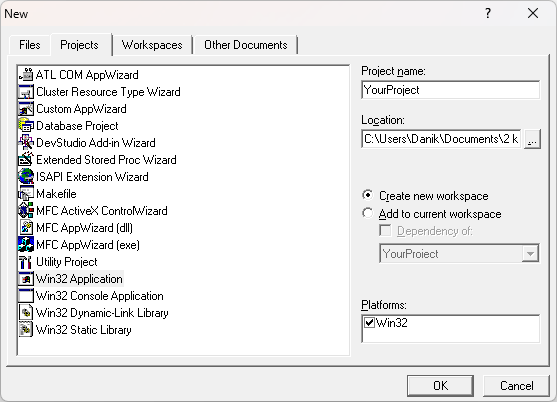


Рис. 7. Вид окна создания проектов.

1. Нажимаем OK. Выбираем A Simple Win32 Application. В дальнейшем везде нажимаем OK.
2. Проверяем создался ли проект. Слева-снизу в окне среды разработки прожимаем FileView. Прожимаем Source Files и <Имя вашего проекта>.cpp. Если все этапы успешно пройдены, значит проект был успешно создан.

2. Необходимо добавить .rc файл к нашему проекту:

1. Открываем окно создания проектов и на вкладку Files. Там выбираем тип Resource Script и делаем название идентичное проекту (рисунок 8). Если будут некоторые диалоговые окна, то везде проставляем OK.

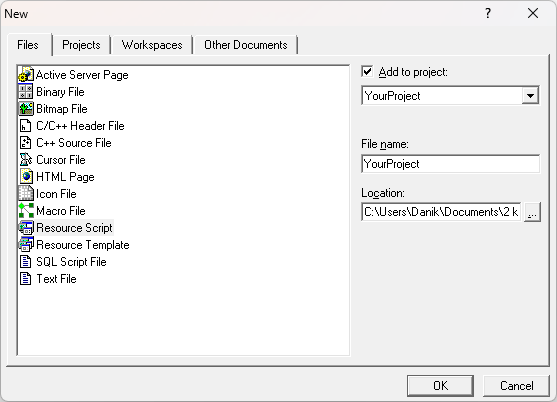


Рис. 8. Местоположение тип файла Resource Script.

1. Для проверки наличия файла можно в окно обозревателя решений и увидеть сам файл в Source Files и дополнительную вкладку Reso…(рисунок 9).

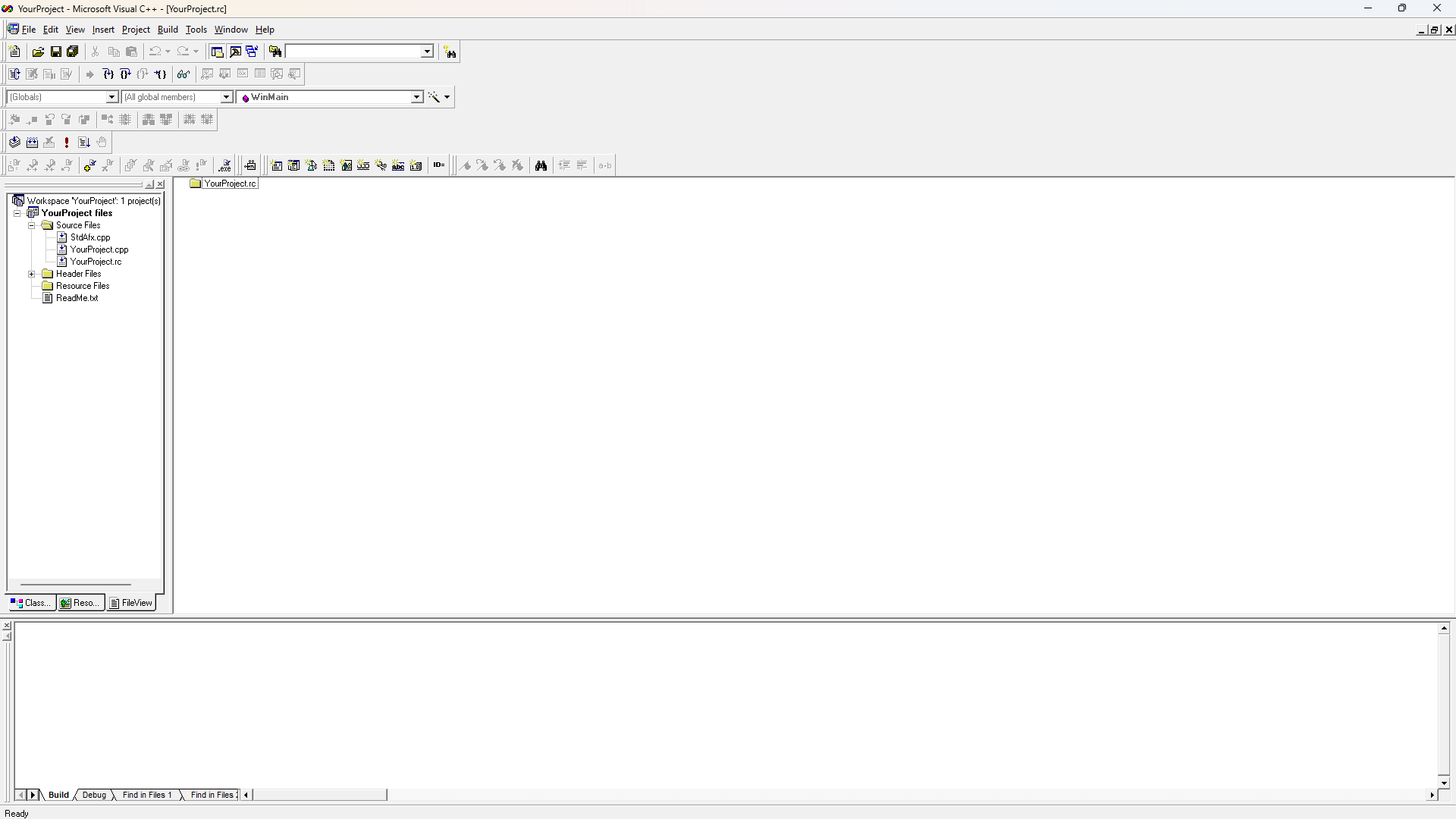


Рис. 9. Вид окна обозревателя решений после добавления .rc файла.

3. Запускаем tkp.exe файл из загруженного проекта. Графический вид окна представлен на рисунке 10.

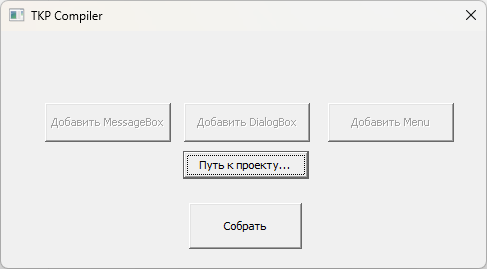


Рис. 10. Вид окна проекта ТКП.

4. Нажимаем на клавишу «Путь к проекту…» и выбираем главный .cpp файл созданного нами проекта (рисунок 11). Появится окно с путем к .cpp файлу. Нажимаем OK.

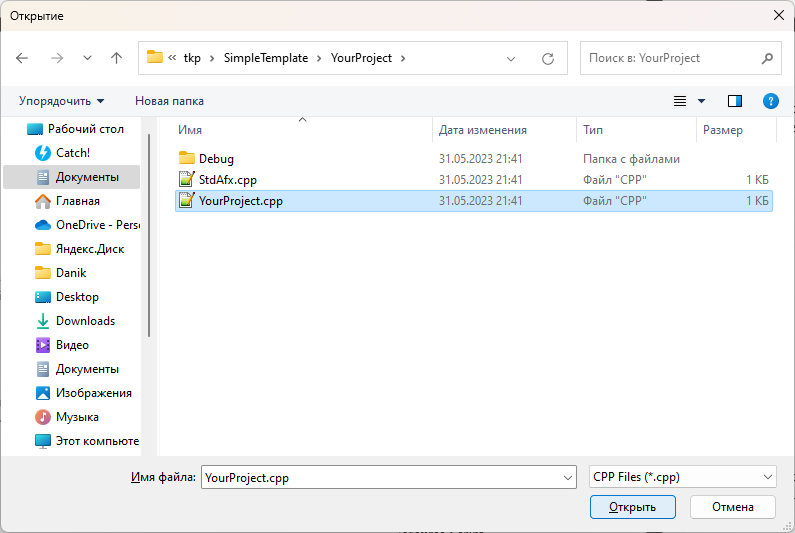


Рис. 11. Пример окна открытия главного .cpp файла.

5. Делаем настройку добавления элементов к проекту и запускаем процесс добавления:

1. Нажимаем на кнопку « Добавить MessageBox ». Появится новое ДО. Прожимаем на какие события добавить окна-сообщения и что писать в этих сообщениях. Пример заполненного окна на рисунке 12. Нажимаем на OK и возвращаемся в главное окно.

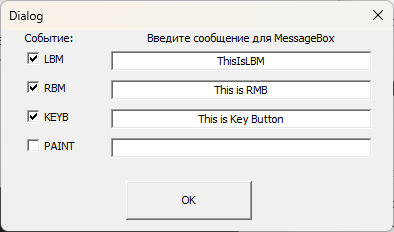


Рис. 12. Вид окна добавления MessageBox`ов.

1. Нажимаем на кнопку « Добавить DialogBox ».. Появится новое ДО. Прожимаем на какие события добавить диалоговые окна. Пример окна на рисунке 13. Нажимаем на OK и возвращаемся в главное окно.

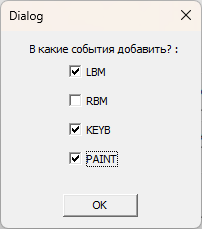


Рис. 13. Вид окна добавления DialogBox`ов.

1. Нажимаем на кнопку « Добавить Menu ».. Появится новое ДО. Пишем сколько пунктов в меню создать. Пример окна на рисунке 14. Нажимаем на OK и возвращаемся в главное окно.

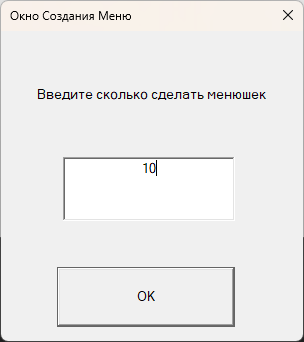


Рис. 14. Вид окна добавления Menu.

1. Прожимаем клавишу « Собрать ». Появятся некоторые отладочные окна-сообщения. Везде прожимаем OK. Если среди них было окно с сообщением «Compiled!», то работа программы выполнена успешно (рисунок 15).

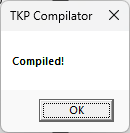


Рис. 15. Окно с сообщением «Compiled!».

6. Проверяем работу созданного проекта:

1. При возращении к Visual C++ появятся много окон-сообщений. Везде прожимаем OK.
2. Для первичной проверки можно посмотреть на главный .cpp файл проекта, он должен был изменится (рисунок 16).

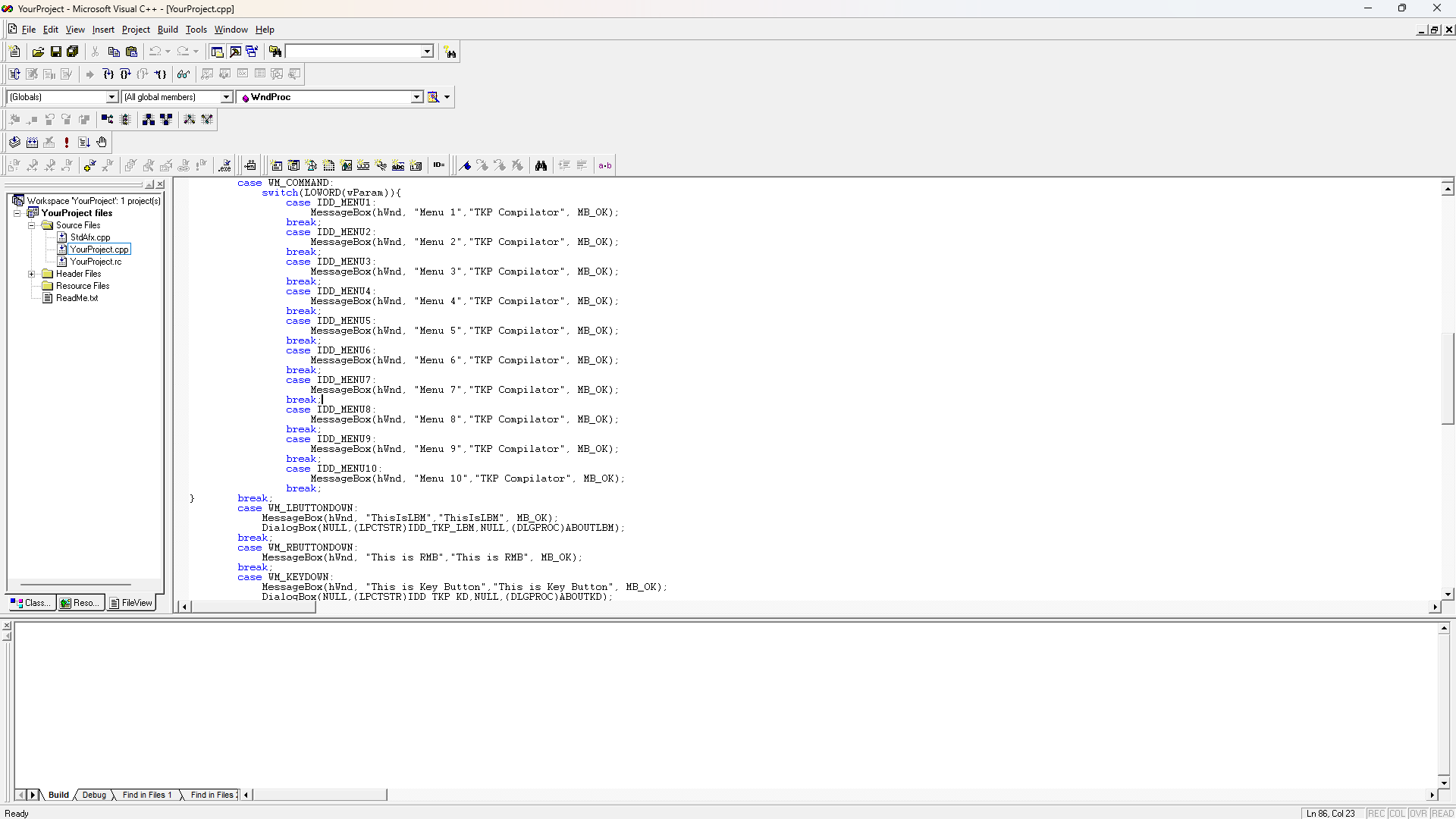


Рис. 16. Окно с измененный .cpp файлом созданного проекта.

1. Запускаем проект с помощью клавиши F5. Появится приложение измененного проекта(рисунок 17).

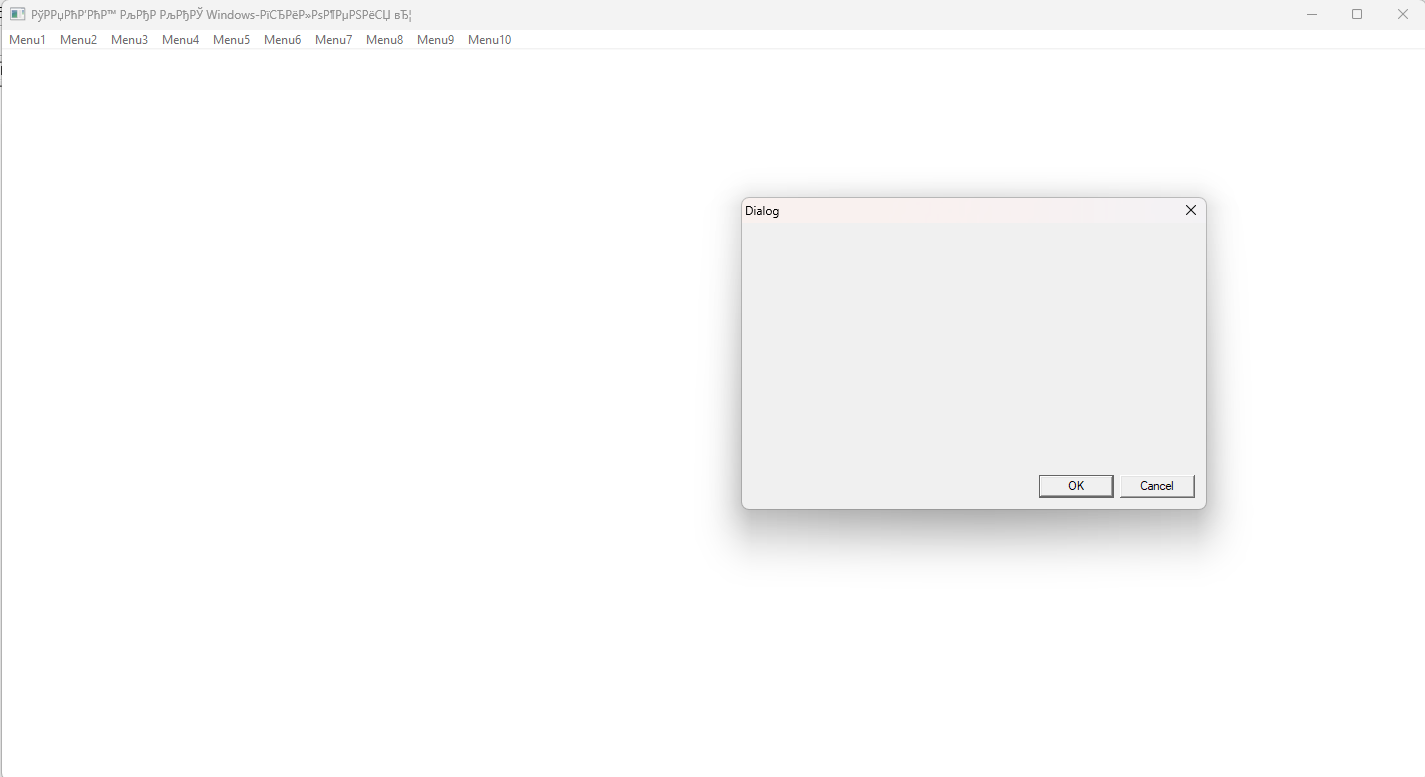


Рис. 17. Окно приложения на основе измененного проекта.

1. Проверяем на наличие ресурсных элементов и их поведение. Примеры на рисунках 18, 19, 20, 21.

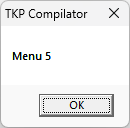
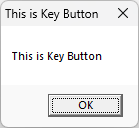
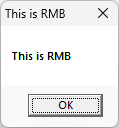


Рис. 18-21. Примеры работы приложения.

Рисунок 18 произошел после нажатия ЛКМ.   
Рисунок 19 произошел после нажатия ПКМ.  
Рисунок 20 произошел после нажатия клавиши на клавиатуре.  
Рисунок 21 произошел после нажатия пункта меню Menu5.

Инструкцию в виде видео-формата можно увидеть по ссылке:  
<https://github.com/DanikKaragodin/TkpCompilator/assets/51264803/0e0ab579-dd54-4b54-aef9-b340b0ac97a5>.