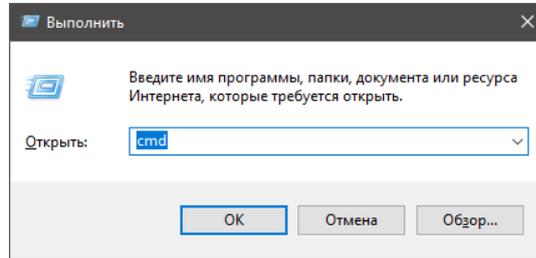


Мануал по настройке MS Visual Studio для работы с библиотекой GTK.

Установка библиотеки GTK.

1. Для установки библиотеки GTK удобнее всего воспользоваться утилитой vcpkg. Открываете консоль (сочетание клавиш Win+R, затем вводите команду «cmd»):



В консоли вводите следующую команду:

```
git clone https://github.com/Microsoft/vcpkg
```

С github скачается проект и распакуется в каталог по умолчанию. По умолчанию распаковка произойдет в каталоге C:\Users\<имя пользователя>. Например, у меня она расположена по следующему пути: C:\Users\TopGun\vcpkg

Если в консоли будет написано:

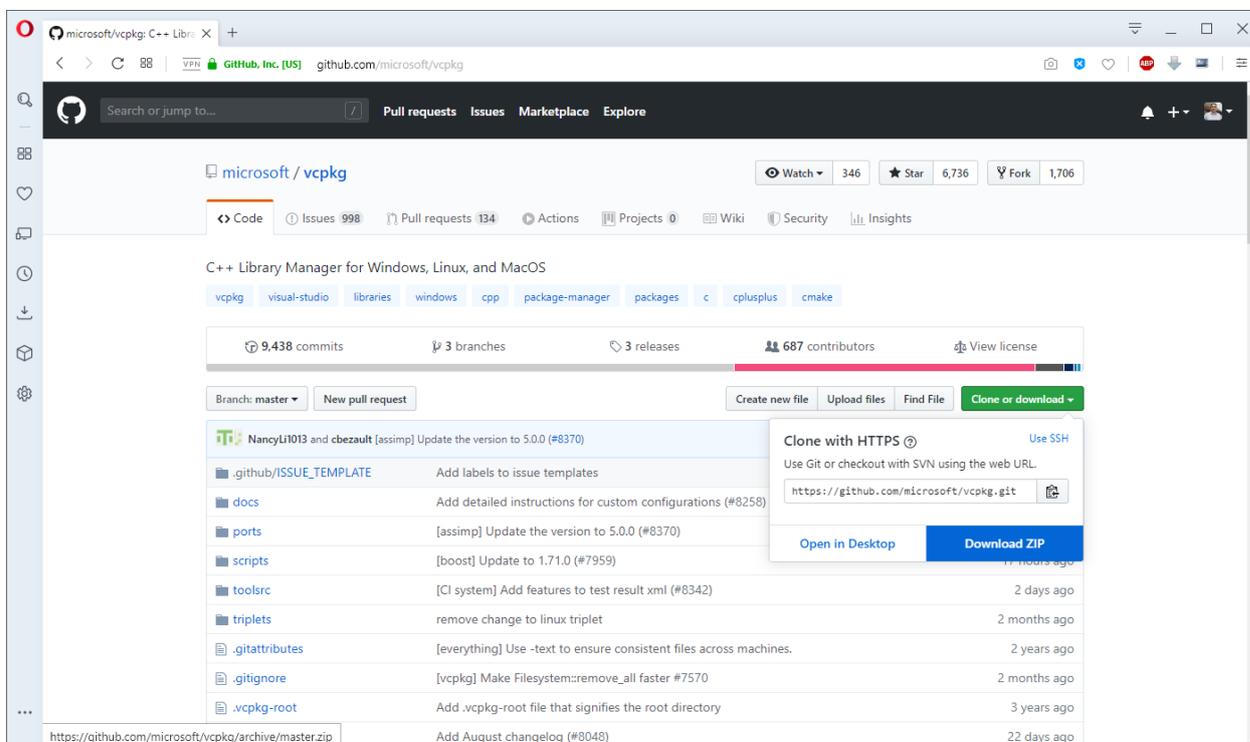
```
C:\Users\TopGun>git  
"git" не является внутренней или внешней  
командой, исполняемой программой или пакетным файлом.
```

то вы читаете, что написано ниже курсивом. У кого стоит git и всё скачалось – переходите к пункту 2.

Что делать, если у вас не установлена система контроля версий git?

Идёте на сайт: <https://github.com/microsoft/vcpkg>

Ищите зелёную кнопку «Clone or download», в открывшемся диалоге жмёте «Download ZIP». Скачается архив 3.5 Мб.



Чтобы дальше всё было валидно и для вас, делаете следующее:

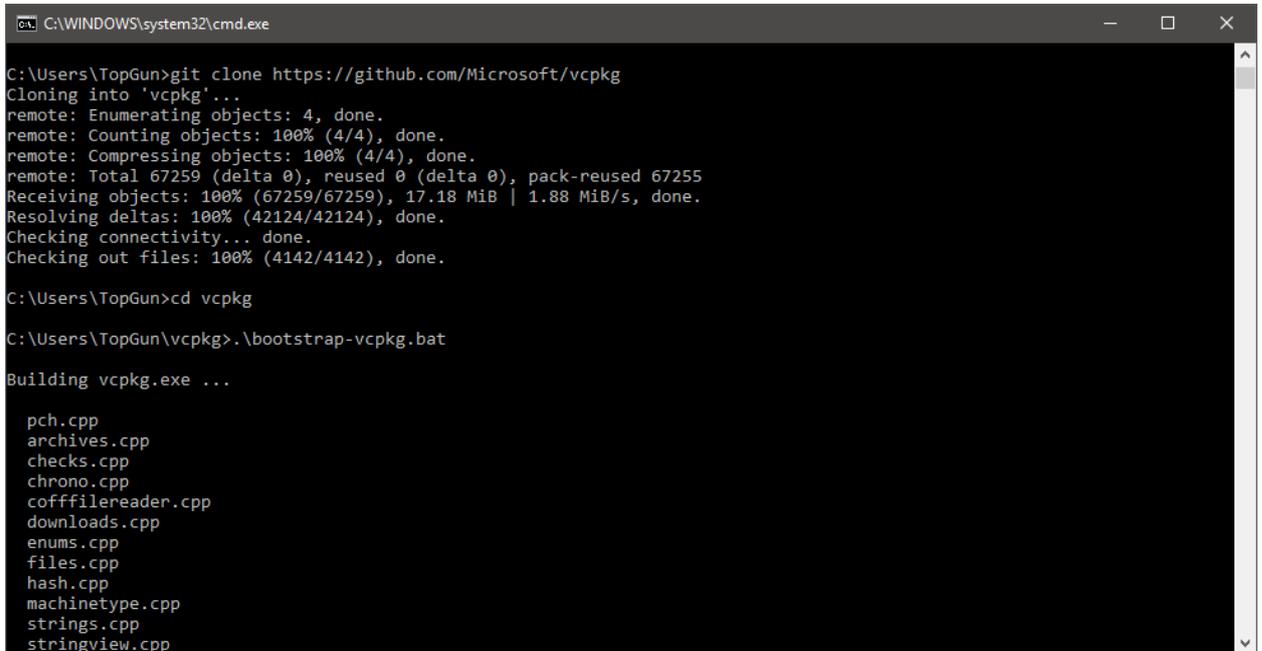
1. открываете в проводнике папку «C:\Users\<имя вашего пользователя>»
2. переписываете в неё архив, распаковываете его, должна появиться папка `vsrkg-master`
3. переименовываете папку `vsrkg-master` в `vsrkg`

ВСЁ! Читаем дальше и делаем всё, что написано!

2. Когда скачивание будет завершено, необходимо зайти в этот каталог и запустить в нём скрипт сборки. Для этого выполните (в консоли) две команды:

```
cd vcpkg
.\bootstrap-vcpkg.bat
```

Первая – зайти в каталог, вторая – выполнить скрипт.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\TopGun>git clone https://github.com/Microsoft/vcpkg
Cloning into 'vcpkg'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 67259 (delta 0), reused 0 (delta 0), pack-reused 67255
Receiving objects: 100% (67259/67259), 17.18 MiB | 1.88 MiB/s, done.
Resolving deltas: 100% (42124/42124), done.
Checking connectivity... done.
Checking out files: 100% (4142/4142), done.

C:\Users\TopGun>cd vcpkg

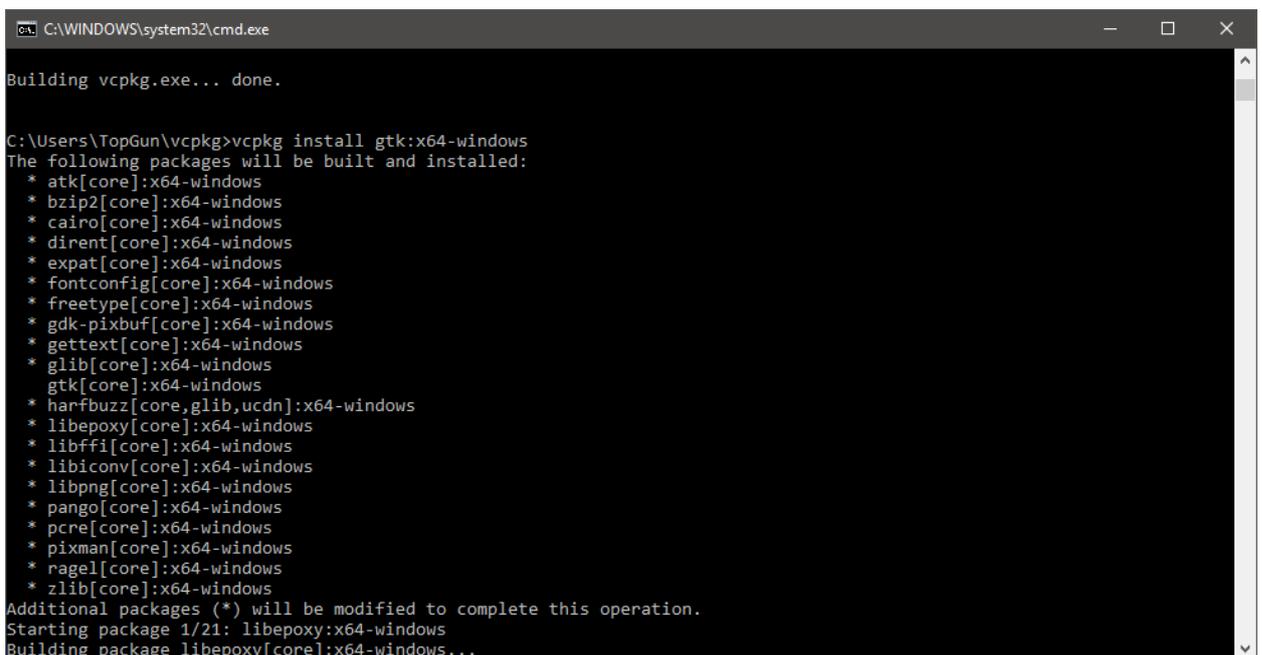
C:\Users\TopGun\vcpkg>.\bootstrap-vcpkg.bat

Building vcpkg.exe ...

pch.cpp
archives.cpp
checks.cpp
chrono.cpp
cofffilereader.cpp
downloads.cpp
enums.cpp
files.cpp
hash.cpp
machinetype.cpp
strings.cpp
stringview.cpp
```

3. После сборки проекта vcpkg можно попросить его установить библиотеку GTK, выполнив следующую команду (в примере собирается 64-х битная версия):

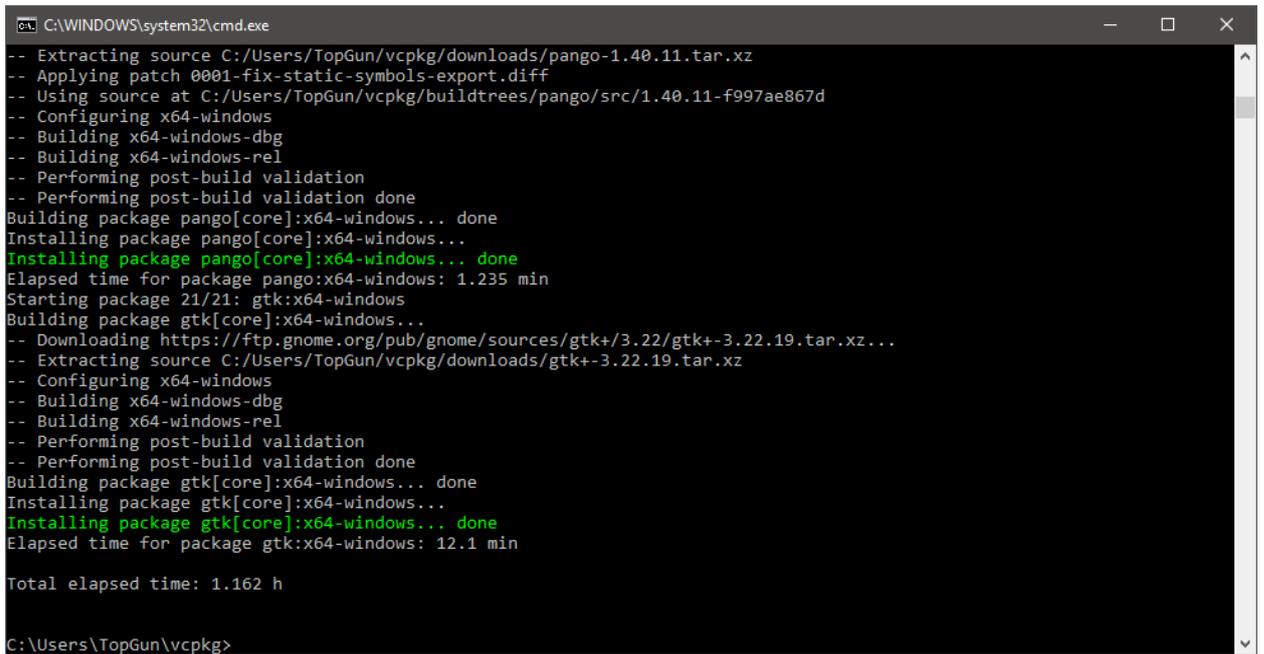
```
vcpkg install gtk:x64-windows
```



```
C:\WINDOWS\system32\cmd.exe
Building vcpkg.exe... done.

C:\Users\TopGun\vcpkg>vcpkg install gtk:x64-windows
The following packages will be built and installed:
 * atk[core]:x64-windows
 * bzip2[core]:x64-windows
 * cairo[core]:x64-windows
 * dirent[core]:x64-windows
 * expat[core]:x64-windows
 * fontconfig[core]:x64-windows
 * freetype[core]:x64-windows
 * gdk-pixbuf[core]:x64-windows
 * gettext[core]:x64-windows
 * glib[core]:x64-windows
 * gtk[core]:x64-windows
 * harfbuzz[core,glib,ucdn]:x64-windows
 * libepoxy[core]:x64-windows
 * libffi[core]:x64-windows
 * libiconv[core]:x64-windows
 * libpng[core]:x64-windows
 * pango[core]:x64-windows
 * pcre[core]:x64-windows
 * pixman[core]:x64-windows
 * ragel[core]:x64-windows
 * zlib[core]:x64-windows
Additional packages (*) will be modified to complete this operation.
Starting package 1/21: libepoxy:x64-windows
Building package libepoxy[core]:x64-windows...
```

У меня на компьютере сборка заняла чуть больше часа:



```
C:\WINDOWS\system32\cmd.exe
-- Extracting source C:/Users/TopGun/vcpkg/downloads/pango-1.40.11.tar.xz
-- Applying patch 0001-fix-static-symbols-export.diff
-- Using source at C:/Users/TopGun/vcpkg/buildtrees/pango/src/1.40.11-f997ae867d
-- Configuring x64-windows
-- Building x64-windows-dbg
-- Building x64-windows-rel
-- Performing post-build validation
-- Performing post-build validation done
Building package pango[core]:x64-windows... done
Installing package pango[core]:x64-windows...
Installing package pango[core]:x64-windows... done
Elapsed time for package pango:x64-windows: 1.235 min
Starting package 21/21: gtk:x64-windows
Building package gtk[core]:x64-windows...
-- Downloading https://ftp.gnome.org/pub/gnome/sources/gtk+/3.22/gtk+-3.22.19.tar.xz...
-- Extracting source C:/Users/TopGun/vcpkg/downloads/gtk+-3.22.19.tar.xz
-- Configuring x64-windows
-- Building x64-windows-dbg
-- Building x64-windows-rel
-- Performing post-build validation
-- Performing post-build validation done
Building package gtk[core]:x64-windows... done
Installing package gtk[core]:x64-windows...
Installing package gtk[core]:x64-windows... done
Elapsed time for package gtk:x64-windows: 12.1 min

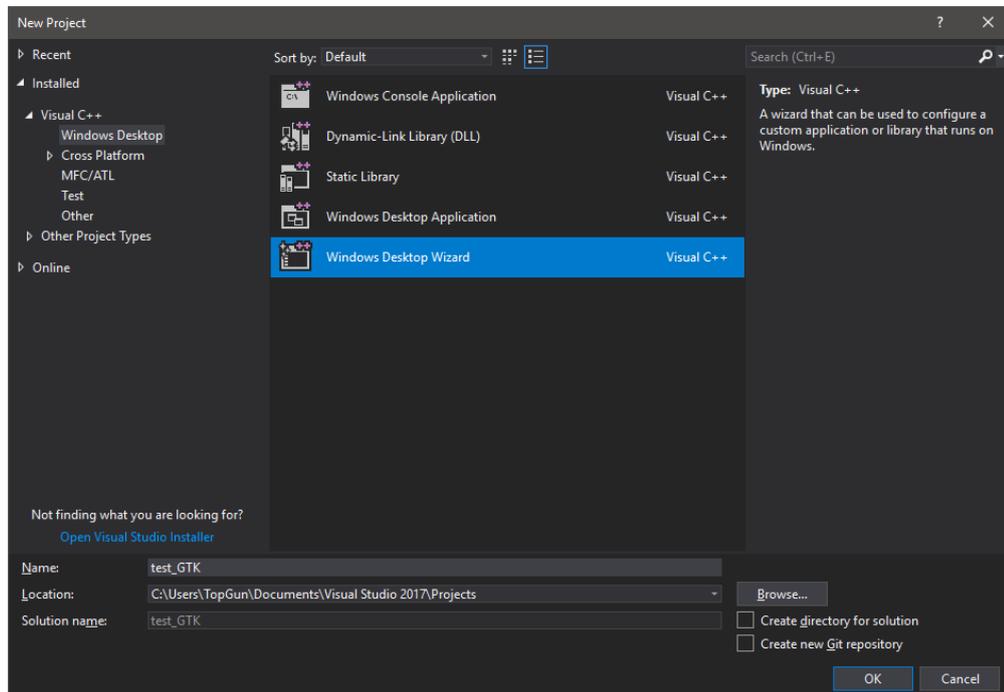
Total elapsed time: 1.162 h

C:\Users\TopGun\vcpkg>
```

Библиотека GTK скачана и собрана, можно создавать проект в Visual Studio.

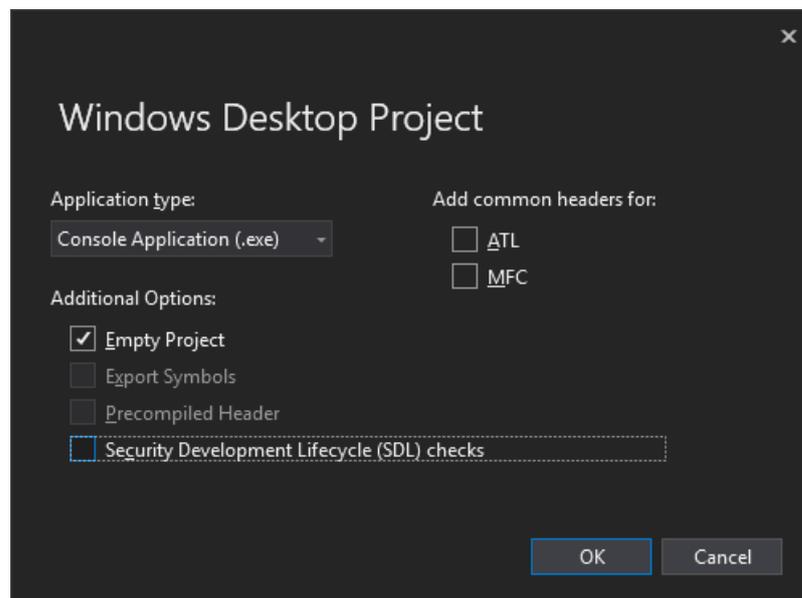
Создание и настройка проекта для работы с GTK.

1. Необходимо запустить Visual Studio и создать простое консольное приложение. Выбираете пункт меню «File» → «New» → «Project».

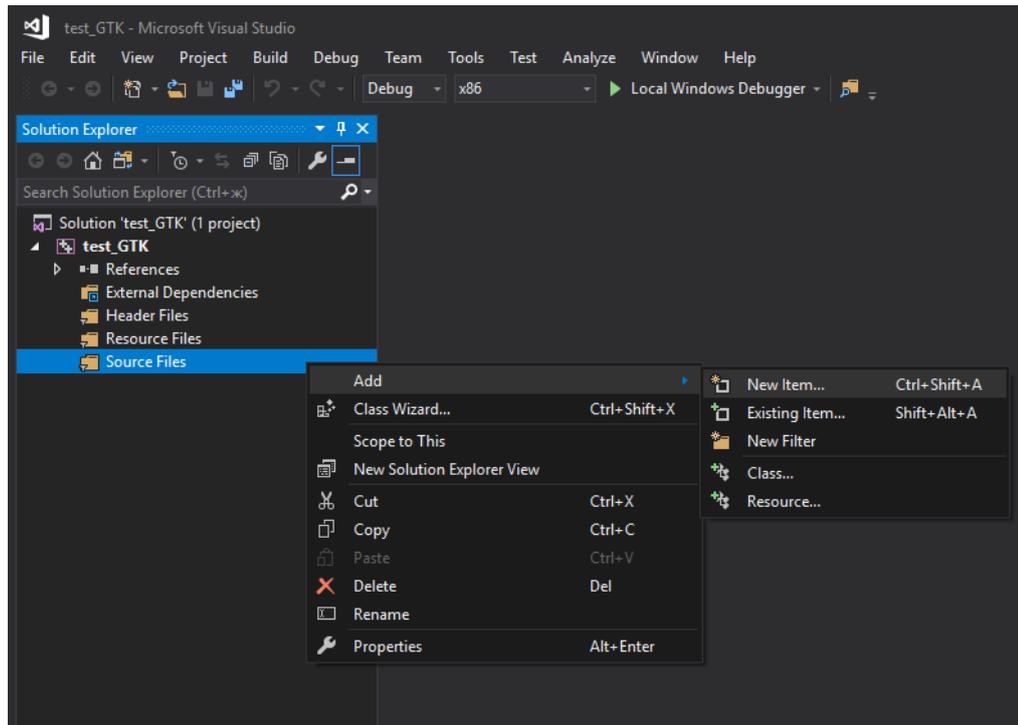


В открывшемся диалоге выбираете слева раздел «Installed» → «Visual C++» → «Windows Desktop», по центру – «Windows Desktop Wizard». Водите имя проекта внизу, в поле «Name», жмёте «Enter».

Откроется второй диалог. Во втором диалоге проверяете, что в выбрасываемом списке «Application type» выбран элемент «Console Application (.exe)», ставите галочку «Empty Project» - мы создаём пустой проект. Жмёте «OK»

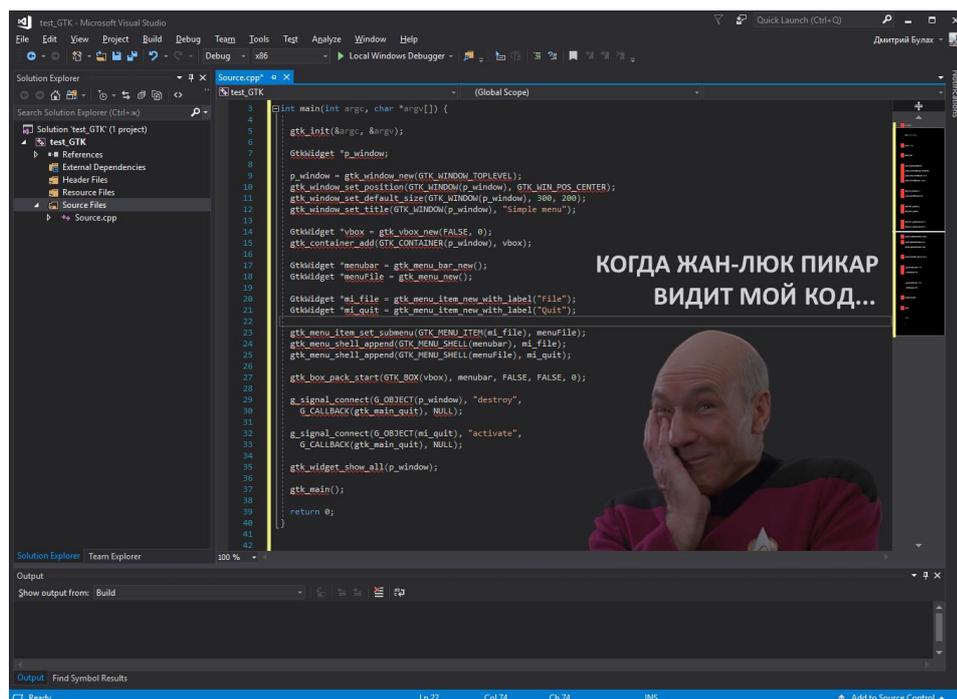


2. Проект создан, добавляем файл с кодом. В дереве проекта на папке «Source Files» вызываете контекстное меню, выбираете пункт «Add» → «New Item...»



В открывшемся диалоге выбираете «C++ File» и жмёте «ОК» (имя файла можно оставить тем, которое предлагается по умолчанию).

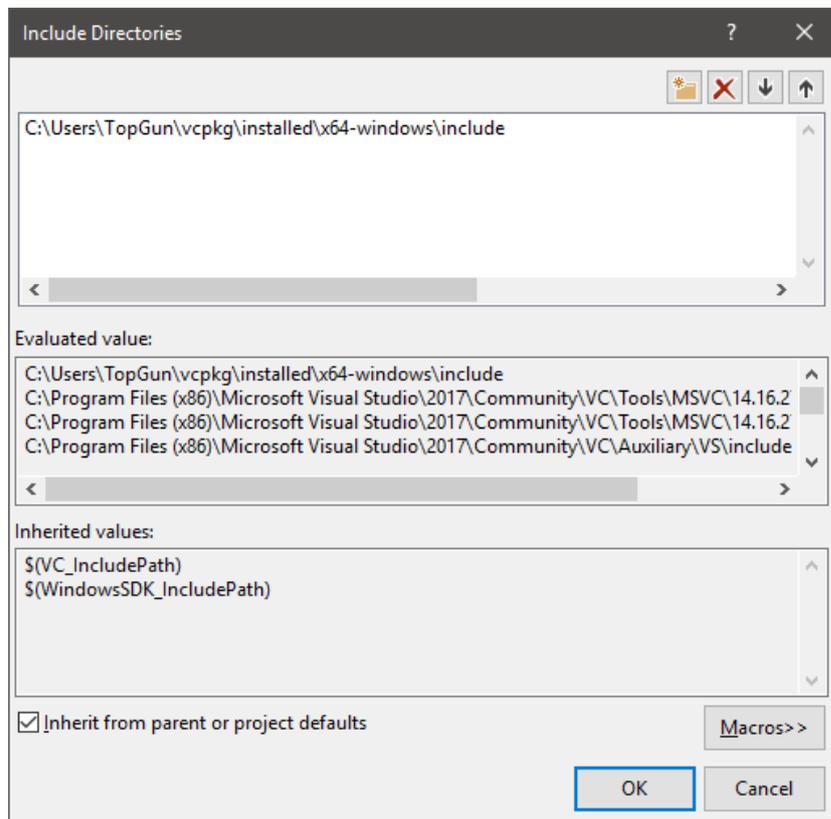
Вставляете код любого из файлов, которые у нас были на ЛР, архив к лабе лежит на сайте. В моём примере я взял код из файла gtk_03.cpp, который с меню.



В открывшемся диалоге вам нужно добавить два пути, по которым содержатся необходимые заголовочные файлы GTK. Эти пути, с точностью до имени пользователя, следующие:

```
C:\Users\<имя пользователя>\vcpkg\installed\x64-windows\include
```

Диалог должен выглядеть следующим образом:



Нажимаете кнопку «OK».

Аналогичные действия нужно проделать для добавления папки с библиотеками (раздел «Library Directories» предыдущего диалога), но путь должен быть следующий:

```
C:\Users\<имя пользователя>\vcpkg\installed\x64-windows\lib
```

Если вы всё сделаете правильно, редактор кода Visual Studio больше не должен будет ругаться и подчёркивать все функции библиотеки GTK.

Для успешной компиляции проекта осталось только указать, какие именно библиотеки из указанного пути нужно подключить к проекту. Делается это в диалоге с настройками свойств проекта. Слева нужно выбрать раздел «Configuration Properties» → «Linker» → «Input».

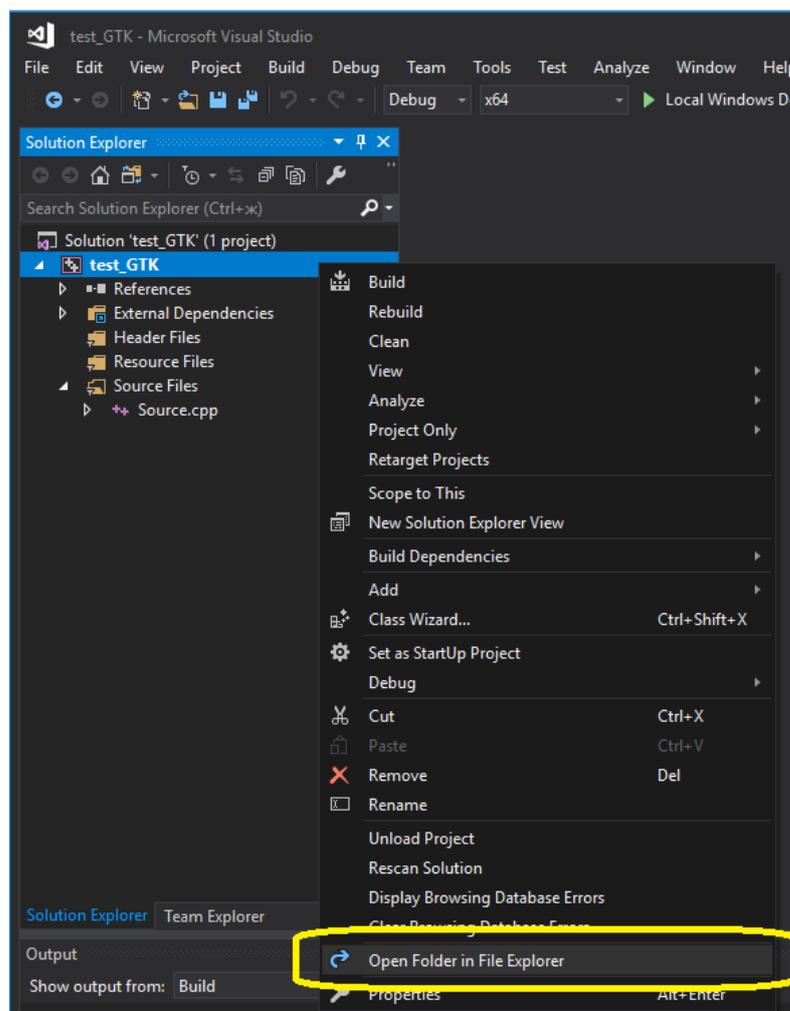
Теперь проект успешно компилируется. Но при попытке запустить будет ругаться.

Последнее, что нужно сделать, так это добавить необходимые динамические библиотеки (файлы с расширением .dll) к вашему проекту. Они лежат в каталоге, расположенном по пути:

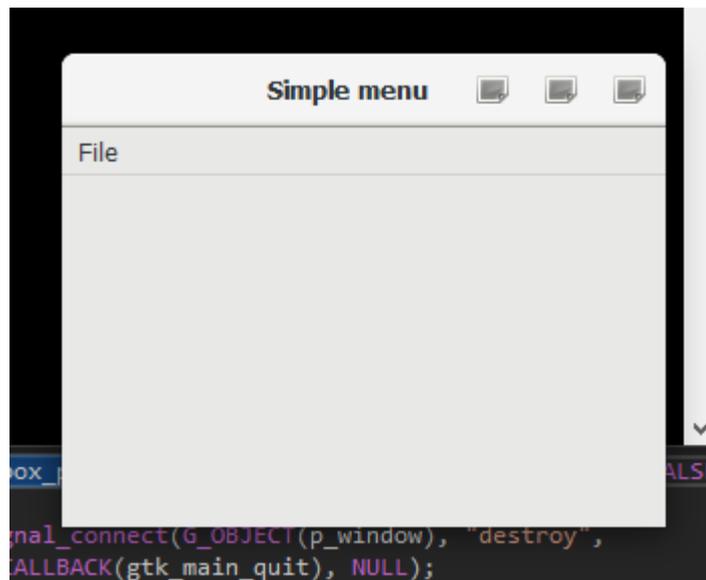
```
C:\Users\<имя пользователя>\vsrpkgs\installed\x64-windows\bin
```

Для запуска проекта потребуются VCE dll из этого каталога!

Их нужно скопировать в папку с вашим проектом, добраться до которой можно вызвав контекстное меню в дереве проекта на имени самого проекта (в дереве проекта отмечается жирным, второй сверху пункт) и выбрав предпоследний пункт меню – «Open Folder in File Explorer».



Теперь проект компилируется, запускается и работает.



Как видно на скриншоте, наблюдается проблема с отрисовкой кнопок окна, но на это можно не обращать внимание.

Как только разберусь, как это поправить, обновлю мануал.