**Методические рекомендации**

**по разработке с использованием IDE**

**Microsoft Visual Studio 2019 Community**

Оглавление

1. [Введение 3](#_Toc21692254)
2. [Установка Visual Studio 5](#_Toc21692255)
3. [Начальный экран Visual Studio. Организация проектов и решений 6](#_Toc21692256)
4. [Настраиваемые окна и панели инструментов 8](#_Toc21692257)
5. [Организация процесса разработки на C++ 11](#_Toc21692258)

# Введение

Интегрированная среда разработки (**Integrated Development Environment, далее «среда разработки» или «IDE»**) – это программа, используемая для разработки программного обеспечения.

Среда разработки взаимодействует с пользователем и настраиваемым набором внешних программ.

Как правило, пользователь не имеет доступа к конкретным внешним программам, IDE вызывает их, передает в них пользовательский ввод и выводит результат их работы в свой пользовательский интерфейс самостоятельно (IDE является “front end”, внешние программы являются “back end”).

При этом текстовый редактор обычно является интегрированным в пользовательский интерфейс и неотъемлемым компонентом среды разработки.

В данных рекомендациях рассматривается разработка на C++ и C# с использованием Microsoft Visual Studio 2019 Community (**далее – Visual Studio**) в ОС Windows.

Для C++ приводятся необходимые составляющие **toolchain** (набора внешних программ, необходимых для разработки) и рассматривается разработка с использованием toolchainпо умолчанию в Visual Studio. Рассматривается разработка консольного приложения, статической библиотеки, динамической библиотеки, оконного приложения для Windows.

Сборкой (**build**) называется получение программы или программного компонента из входных файлов, в том числе файлов исходного кода (**source code**, мн. ч. **sources**), мультимедиа ресурсов (звуков, изображений, видео), сценария сборки (**build script**).

Рассматриваются встроенные средства сборки Visual Studio (интегрированная система сборки **MSBuild**), а также сторонняя система сборки **CMake**.

В приложении к рекомендациям приведены типовые сценарии системы сборки CMake для перечисленных случаев, а также типовая стратегия контроля версий с использованием системы контроля версий (**Version Control System, VCS**) **Git**.

# Установка Visual Studio

Загрузите Visual Studio Installer с [официального сайта Microsoft](https://www.microsoft.com), выбрав в числе продуктов Visual Studio 2019 Community.   
Visual Studio Installer позволяет объединять составляющие среды разработки в т. н. «рабочие нагрузки» (**workloads**), например, рабочая нагрузка «Разработка классических приложений на C++» содержит средства, необходимые для разработки консольных программ без использования сторонних средств, например, C++ / CLI. Убедитесь, что будут установлены workloads, соответствующие разработке на C++ и C# в консоли.

Также возможен ручной выбор набора для установки. Вне зависимости от выбора рабочей нагрузки или отдельных компонентов, убедитесь, что установлены:

* Средства CMake C++ для Windows,
* Расширение GitHub для Visual Studio,
* Git для Windows.

# Начальный экран Visual Studio. Организация проектов и решений

После установки запустите Visual Studio и ознакомьтесь с Начальным экраном (рисунок 1).

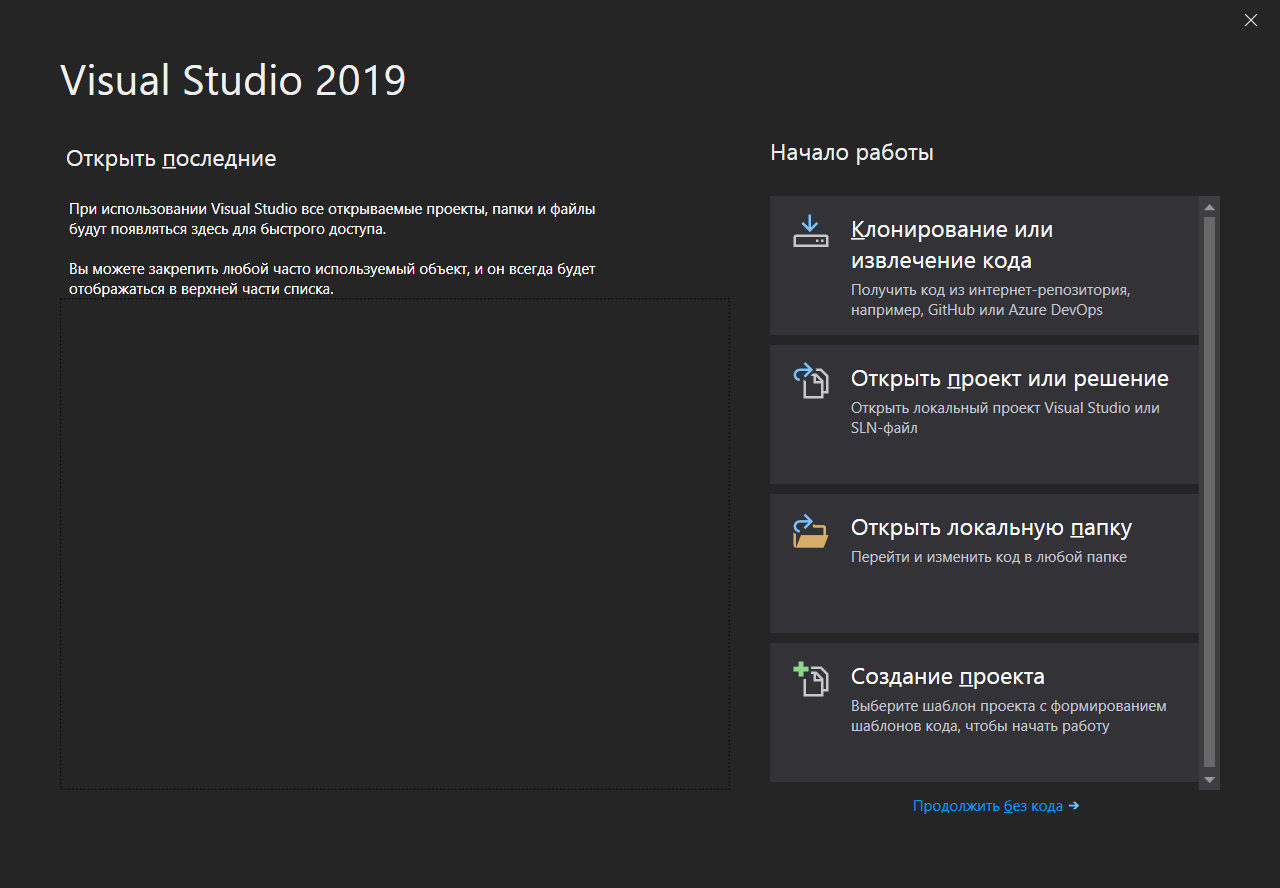


Рисунок - Начальный экран

Проектом (**project**) в Visual Studio называется набор файлов, участвующих в сборке. Один проект представляет одно приложение, одну библиотеку или один набор ресурсов.

Решением (**solution**) в Visual Studio называется один или несколько проектов, их общие настройки интерфейса среды разработки, их общие сценарии сборки или прочие файлы, не связанные с конкретным проектом.

Одно решение может использоваться для представления одного программного продукта целиком.

В рамках лабораторных работ возможно сопоставление каждой дисциплине своего решения: так, Объектно-ориентированное программирование на C++ потребует стандартных средств разработки на C++ для всех проектов, а работы по Технологиям пространственного моделирования могут иметь общий toolchain для разработки на C# + Windows.Forms с использованием библиотеки OpenTK.

Репозиторием (**software** **repository,** также просто **repository**) называется централизованное хранилище файлов, как правило, представляющих проект.

Работа в Visual Studio возможна после создания проекта, клонирования существующего проекта из локального (на диске) или удаленного (в сети) репозитория, открытия существующего проекта или открытия произвольной папки на диске. Для репозиториев под управлением системы контроля версий Git возможна работа с версиями из интерфейса Visual Studio и соответствующего расширения от GitHub (GitHub для Visual Studio).

В дальнейшем будет рассмотрен процесс разработки по перечисленным сценариям.

# Настраиваемые окна и панели инструментов

На начальном экране Visual Studio выберите «Продолжить без кода».

Откроется главный экран среды разработки без активного проекта, где можно настроить предпочитаемый внешний вид, а затем перейти к управлению проектами (например, созданию, открытию или клонированию), как если бы соответствующий сценарий был выбран непосредственно на начальном экране.

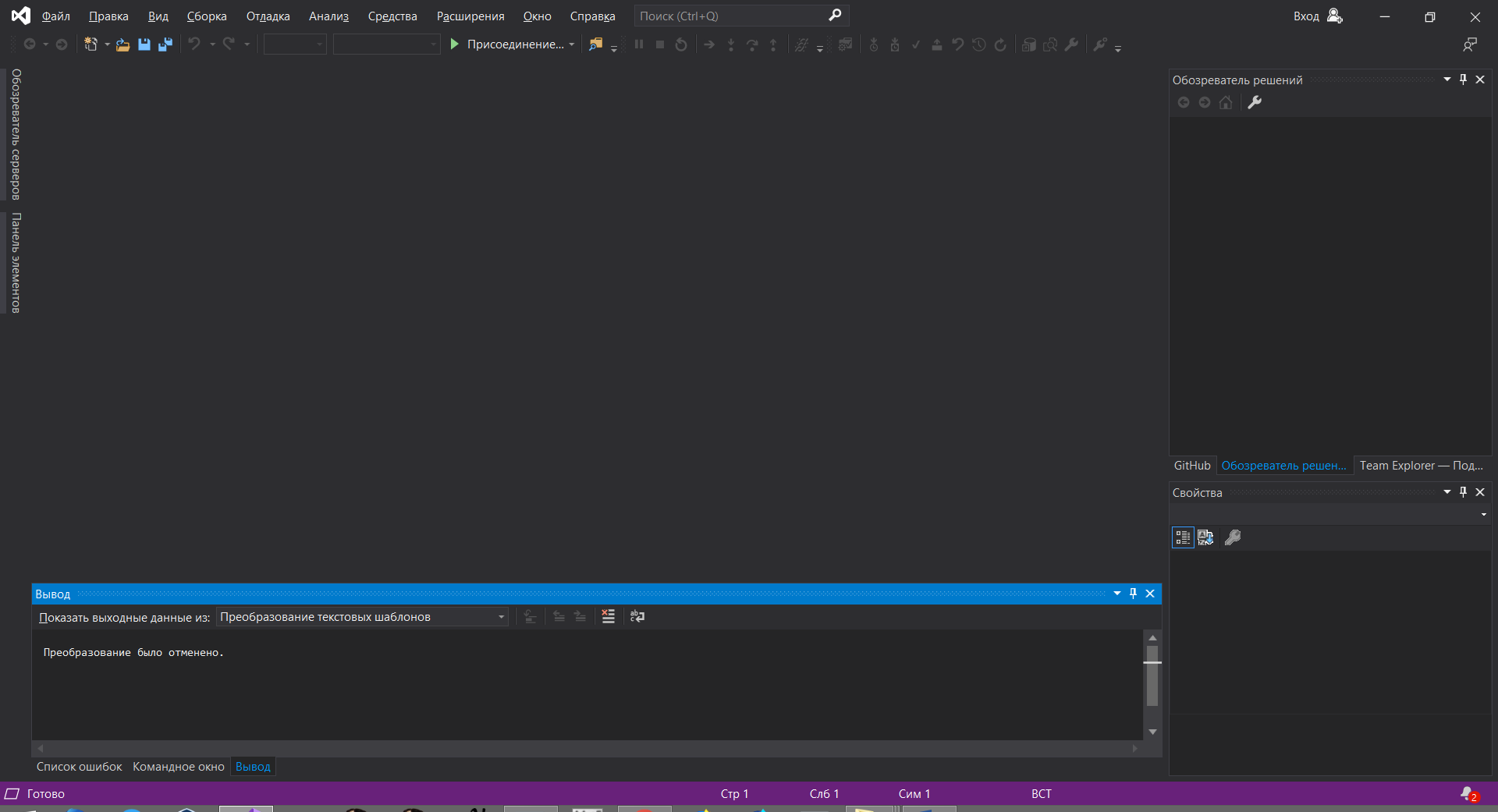


Рисунок - Главный экран

**Настраиваемые окна**

Измените по своему усмотрению взаимное расположение настраиваемых окон (в примере на Рисунке 2 это Вывод, Обозреватель решений и Свойства).

Для этого перетащите окно, кликнув по его заголовку, в произвольную область для его открепления от главного экрана или на одну из позиций всплывающей подсказки, позволяющей разместить окно по границе экрана.

После закрепления окна в пределах главного экрана, возможно размещение других окон относительно него (например, окно Свойства выровнено по нижней границе окна Обозревателя решений).

Окна Список ошибок, Командное окно и Вывод на Рисунке 2 размещены в одной области главного экрана. Для операций с одним из них перетаскивайте его не за заголовок, содержащий название активного окна (на Рисунке 2 это Вывод), а за вкладку в нижней части с названием перетаскиваемого окна.

Возможно добавление в эту группу еще нескольких вкладок, для этого перетащите добавляемое окно на её область экрана так, чтобы подсветить эту область целиком.

Наконец, клик по пиктограммам канцелярской кнопки и “X” в правом верхнем углу окна позволяет настроить автоматическое скрытие окна (окно открывается и закрывается по клику на заголовок, который может изменить своё расположение) или закрыть его.

Закрытые окна можно восстановить из элемента меню Вид главного экрана и разместить заново.

Для управления файлами проекта (в том числе добавления файлов в проект и создания новых файлов) необходимо окно Обозреватель решений.

Окна Вывод и Список ошибок используются компилятором для вывода хода компиляции, возможных ошибок и предупреждений.

В обзоре средств контроля версий и отладки будут рассмотрены и другие.

Добавьте перечисленные окна на главный экран и перейдите к подзаголовку Панели инструментов.

**Панели инструментов**

**Возможно выборочное включение и отключение панелей инструментов – групп иконок, расположенных под строкой меню главного экрана.**

**Список панелей инструментов с переключаемым отображением (вкл – выкл) находится в Главный экран –> Вид -> Панели инструментов.**

**После включения панели она появляется под строкой меню. Для перетаскивания доступна левая сторона панели. При наведении курсора он приобретает форму четырехгранной звезды с лучами-стрелками.**

**Раскрывающийся список на правой стороне каждой панели содержит пункт Добавить или удалить кнопки для тонкой настройки.**

**Убедитесь, что на главном экране присутствуют панели инструментов Сборка, Отладка, Управление версиями и Стандартная.**

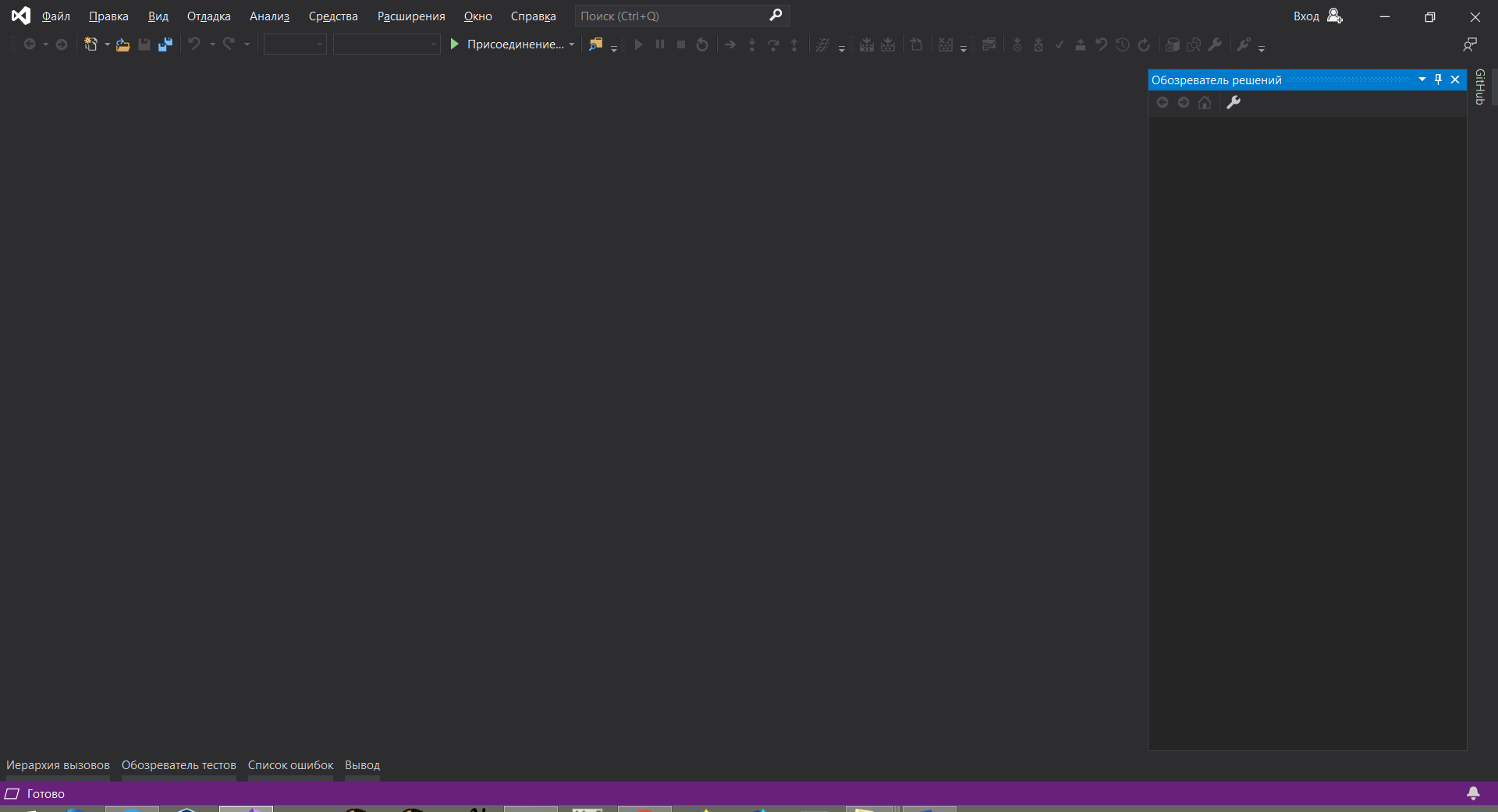


Рисунок - Пример настройки окон и панелей инструментов на главном экране

В примере на Рисунке 3 окна Иерархия вызовов, Обозреватель тестов, Список ошибок и Вывод имеют объединенную рабочую область, закреплены и выровнены по нижней границе главного экрана, автоматическое скрытие включено.

Окно Обозреватель решений выровнено по правой границе главного экрана, не скрыто и сейчас активно.

Окно Github также выровнено по правой границе, но скрыто и будет доступно по клику по заголовку.

Отображены панели инструментов Стандартная, Сборка, Система управления версиями и Отладка. Для панели Отладка добавлен элемент Начать отладку.

Настройка пользовательского интерфейса завершена. В следующих главах будет рассмотрен процесс разработки.

# Организация процесса разработки на C++