

# 1

## Введение и тест-драйв популярных бесплатных компиляторов C++

### **В этой главе вы:**

- узнаете о структуре книги и получите советы о том, с чего начать изучение большого, сложного и мощного языка C++20;
- научитесь компилировать и запускать написанное на C++ тестовое приложение с помощью трех компиляторов, которые мы рекомендуем: Visual C++ из Microsoft Visual Studio (в Windows), Clang C++ из Xcode (в macOS) и g++ из GNU Compiler Collection (в Linux);
- узнаете, как запускать компиляторы g++ и clang++ в контейнерах Docker, с которыми можно работать и в Windows, и в macOS, и в Linux;
- найдете ссылки на интернет-ресурсы, рассказывающие о более чем 40-летней истории C++;
- узнаете, почему многопоточное (конкурентное) программирование на C++ ускоряет работу приложений на компьютерах с многоядерными процессорами;
- познакомитесь с концепцией объектов, которую мы в начале книги продемонстрируем на простых примерах с готовыми объектами, а затем подробно рассмотрим в главах, посвященных объектно-ориентированному программированию (начиная с главы 9).

## 1.1. Введение

Добро пожаловать в C++! Это один из самых популярных в мире языков программирования<sup>1</sup>. Мы поможем вам изучить его в современной версии, которая называется C++20 («Си плюс плюс двадцать»). Это последняя на сегодняшний день версия, которая стандартизирована Международной организацией по стандартизации (International Organization for Standardization, ISO). Темы главы 1 рассчитаны на изучение языка практически с нуля:

- ✦ краткий обзор книги для быстрого ознакомления с ее структурой;
- ✦ тест-драйв трех популярных бесплатных компиляторов C++;
- ✦ закон Мура, многоядерные процессоры и решающая роль многопоточного программирования в создании быстродействующих приложений на C++;
- ✦ основные принципы и термины объектно-ориентированного программирования, которые будут встречаться в книге далее.

### Краткий обзор этой книги

Прежде чем вы углубитесь в чтение, мы рекомендуем вам быстро взглянуть на всю структуру книги «с высоты 40 000 футов», чтобы получить общее представление о том, как вы будете изучать большой, сложный и мощный язык C++20. Для этого в книге есть:

- ✦ Оглавление в виде наглядной схемы на внутренней стороне обложки.
- ✦ Предисловие, поясняющее суть книги и наш подход к современному программированию на C++. Здесь мы представляем метод готовых объектов, который позволит вам использовать минимум простых инструкций C++, чтобы заставить мощные классы выполнять важные задачи. Вы сможете это делать задолго до того, как самостоятельно напишете ваши первые классы. Обязательно прочтите в предисловии раздел «Краткое содержание книги», в котором указаны ключевые темы каждой главы. При чтении этого раздела вам будет удобно сверяться со схемой-оглавлением.

### Ресурсы по истории C++

В 1979 году Бьерн Страуструп (Bjarne Stroustrup; в другой транскрипции — Бьярне Страуструп) приступил к разработке языка программирования C++, который он сначала назвал «C with Classes»<sup>2</sup>. Сегодня не менее 5 миллионов разработчиков (по некоторым оценкам — до 7,5 миллиона<sup>3, 4</sup>) используют C++ для создания

<sup>1</sup> «ТЮБЕ Index». Был доступен: 15 сентября 2023 г. <https://www.tiobe.com/tiobe-index>.

<sup>2</sup> «Страуструп, Бьерн». [https://ru.wikipedia.org/wiki/Страуструп,\\_Бьерн](https://ru.wikipedia.org/wiki/Страуструп,_Бьерн).

<sup>3</sup> State of the Developer Nation, 21st Edition», Q3 2021. Был доступен: 15 сентября 2023 г. <https://www.slashdata.co/free-resources/state-of-the-developer-nation-21st-edition>.

<sup>4</sup> Тим Андерсон (Tim Anderson), «Report: World's Population of Developers Expands, JavaScript Reigns, C# Overtakes PHP», 26 апреля 2021 г. Был доступен: 15 сентября 2023 г. [https://www.theregister.com/2021/04/26/report\\_developers\\_slashdata](https://www.theregister.com/2021/04/26/report_developers_slashdata).

критически важных цифровых систем и прикладных программ (приложений)<sup>1,2</sup>. На этом языке написаны многие компоненты операционных систем Windows<sup>3</sup> и macOS<sup>4</sup>. C++ широко применяется при разработке браузеров (например, Google Chrome<sup>5</sup> и Mozilla Firefox<sup>6</sup>), систем управления базами данных (например, MySQL<sup>7</sup> и MongoDB<sup>8</sup>) и многих других видов приложений.

История языка C++ хорошо документирована:

- ✦ Эта статья в «Википедии» содержит подробную историю C++ и много полезных ссылок:

<https://ru.wikipedia.org/wiki/C++>

- ✦ Бьери Страуструп, создатель C++, рассказывает о том, как устроен этот язык и как он совершенствовался с момента создания до появления современной версии C++20:

<https://www.stroustrup.com/C++.html#design>

- ✦ Здесь вы найдете список основных этапов развития C++ с момента создания и другие исторические сведения:

<https://ru.cppreference.com/w/cpp/language/history>

## 1.2. Тест-драйв: компиляция приложения, написанного на C++20

В этом разделе вы будете компилировать, запускать и проверять в работе ваше первое приложение на C++<sup>9</sup>. Это игра, в которой надо угадать случайное число от 1 до 1000. Если вы введете правильное число, игра закончится. Если вы ошибетесь, приложение сообщит, что вы ввели слишком большое или слишком маленькое число. Нет никаких ограничений на количество попыток, которые вы можете сделать.

---

<sup>1</sup> «Top 10 Reasons to Learn C++». Был доступен: 15 сентября 2023 г. <https://www.geeksforgEEKS.org/top-10-reasons-to-learn-c-plus-plus>.

<sup>2</sup> «What Is C++ Used For? Top 12 Real-World Applications and Uses of C++». Был доступен: 15 сентября 2023 г. <https://www.softwaretestinghelp.com/cpp-applications>.

<sup>3</sup> «What Programming Language Is Windows Written In?» Был доступен: 15 сентября 2023 г. <https://social.microsoft.com/Forums/en-US/65a1fe05-9c1d-48bf-bd40-148e6b3da9f1/what-programming-language-is-windows-written-in>.

<sup>4</sup> «macOS». <https://ru.wikipedia.org/wiki/MacOS>.

<sup>5</sup> «Google Chrome». [https://ru.wikipedia.org/wiki/Google\\_Chrome](https://ru.wikipedia.org/wiki/Google_Chrome).

<sup>6</sup> «Firefox». <https://ru.wikipedia.org/wiki/Firefox>.

<sup>7</sup> «MySQL». <https://ru.wikipedia.org/wiki/MySQL>.

<sup>8</sup> «MongoDB». <https://ru.wikipedia.org/wiki/MongoDB>.

<sup>9</sup> Мы намеренно не приводим исходный код этого приложения. Пока мы хотим только показать вам, как компилировать и запускать программы с помощью трех компиляторов, которые мы рекомендуем. А с генерацией случайных чисел вы познакомитесь в главе 5.

## Рассматриваемые компиляторы и среды разработки

Мы покажем, как компилировать исходный код на языке C++ и запускать скомпилированную программу в трех операционных системах:

- ✦ в Windows — с помощью интегрированной среды разработки Microsoft Visual Studio 2022 Community Edition (раздел 1.2.1);
- ✦ в macOS — с помощью интегрированной среды разработки Apple Xcode (раздел 1.2.2);
- ✦ в Linux — с помощью компилятора g++, который входит в набор GNU Compiler Collection (раздел 1.2.3);
- ✦ в любой из трех операционных систем — с помощью этого же компилятора g++, работающего в контейнере Docker (раздел 1.2.4);
- ✦ в любой из трех операционных систем — с помощью компилятора clang++, работающего в контейнере Docker (раздел 1.2.5).

Вы можете прочитать только тот раздел, который соответствует вашему компьютеру. Чтобы использовать компиляторы g++ и clang++ в контейнерах Docker, предварительно установите и запустите Docker, как описано в разделе «Перед началом работы».

### 1.2.1. Компиляция и запуск приложения в Windows с помощью Visual Studio 2022

В этом разделе вы скомпилируете и запустите написанную на C++ программу в операционной системе Windows, используя интегрированную среду разработки Microsoft Visual Studio 2022 Community Edition<sup>1</sup>. Она существует в нескольких версиях. Параметры, меню и необходимые действия, описанные в этом разделе, могут незначительно отличаться в зависимости от версии. Далее мы будем использовать термины Visual Studio и IDE (Integrated Development Environment — интегрированная среда разработки).

#### Шаг 1. Подготовка программного обеспечения

Если вы еще не сделали этого, прочитайте раздел «Перед началом работы», установите Visual Studio и скачайте с сайта Deitel файл с включенными в книгу примерами исходного кода.

#### Шаг 2. Запуск Visual Studio

Откройте Visual Studio из меню операционной системы Start (Пуск). Закройте начальное окно Visual Studio, нажав клавишу Esc. Не щелкайте мышью на кнопке X в правом верхнем углу: она завершает работу Visual Studio. Вы сможете снова открыть начальное окно в любое время, выбрав File > Start window. Мы будем

---

<sup>1</sup> Мы рассматриваем Microsoft Visual Studio 2022 Community Edition версии 17.0.5.

обозначать символом > выбор пункта меню. Например, `File > Open` означает «выберите пункт `Open` в меню `File`».

### Шаг 3. Создание проекта

**Проект (project)** — это группа взаимосвязанных файлов (содержащих исходный код на языке C++), из которых состоит приложение. Visual Studio позволяет объединять разрабатываемые приложения в проекты и **решения (solutions)**. Каждое решение может содержать один или несколько проектов. Решения, состоящие из нескольких проектов, необходимы только для очень больших приложений. Для каждого приложения из этой книги нужно решение только с одним проектом. Изучая наши примеры кода, вы будете начинать с **пустого проекта (empty project)** и добавлять в него файлы. Для создания проекта:

1. Выберите `File > New > Project...`, чтобы открыть диалоговое окно `Create a New Project`.
2. Выберите шаблон с именем `Empty Project` и тегами `C++`, `Windows` и `Console`. Он предназначен для программ, выполняемых в командной строке. (В зависимости от версии и настроек Visual Studio в списке может быть много других шаблонов. Для поиска нужного шаблона введите его имя в строку `Search for templates`). Щелкните на кнопке `Next`, чтобы открыть диалоговое окно `Configure your new project`.
3. Укажите `Project name` (имя проекта) и `Location` (путь к файлам проекта). В качестве имени мы выбрали `cpp20_test`. В качестве пути мы выбрали папку `examples`, содержащую примеры кода из этой книги. Щелкните на кнопке `Create`, чтобы создать новый проект.

Visual Studio создаст ваш проект, назначив для его файлов папку:

```
C:\Users\ИмяПользователя\Documents\examples
```

(или другую указанную вами папку), и откроет главное окно.

Когда вы пишете и редактируете код, Visual Studio отображает каждый файл с кодом вашего приложения в отдельной вкладке. Для просмотра этих файлов и для управления ими предназначено окно `Solution Explorer` (оно может находиться справа или слева в главном окне Visual Studio). В нем есть папка `Source Files`, в которую вы будете помещать файлы с нашими примерами кода. Если `Solution Explorer` не отображается, вы можете вывести его на экран, выбрав `View > Solution Explorer`.

### Шаг 4. Добавление файла `GuessNumber.cpp` в проект

Теперь добавьте файл `GuessNumber.cpp` к созданному на шаге 3 проекту. Для этого в `Solution Explorer`:

1. Щелкните правой кнопкой мыши на папке `Source Files` и выберите `Add > Existing Item...`

2. В появившемся диалоговом окне перейдите в папку `ch01`, вложенную в папку `examples`, выберите `GuessNumber.cpp` и щелкните на кнопке `Add`<sup>1</sup>.

## Шаг 5. Настройка проекта для работы с C++20

Входящий в состав Visual Studio компилятор C++ поддерживает несколько версий стандарта C++. Для работы с этой книгой надо выбрать в параметрах проекта версию C++20:

1. В `Solution Explorer` щелкните правой кнопкой мыши на узле проекта `cpp20_test` и выберите `Properties`, чтобы открыть диалоговое окно `Property Pages` проекта `cpp20_test`.
2. В раскрывающемся списке `Configuration` выберите `All Configurations`. В раскрывающемся списке `Platform` выберите `All Platforms`.
3. В левом столбце разверните узел `C/C++` и выберите `Language`.
4. В правом столбце щелкните в поле справа от `C++ Language Standard`, затем щелкните на стрелке вниз, выберите `ISO C++20 Standard (/std:c++20)` и щелкните на кнопке `OK`.

## Шаг 6. Компиляция и запуск проекта

Чтобы скомпилировать и запустить проект, то есть проверить приложение в работе, выберите `Debug > Start without debugging` или нажмите `Ctrl + F5`. Если исходный код компилируется без ошибок, Visual Studio откроет окно командной строки и выполнит приложение. Вы увидите такое сообщение:

```
I have a number between 1 and 1000.  
Can you guess my number?  
Please type your first guess.  
?
```

## Шаг 7. Ввод первого числа

После знака `?` в командной строке введите `500` и нажмите `Enter`. Ответ приложения будет меняться при каждом его запуске. У нас приложение вывело сообщение `"Too low. Try again."`, указывающее, что введенное число меньше случайного числа, которое надо угадать:

```
I have a number between 1 and 1000.  
Can you guess my number?  
Please type your first guess.  
? 500  
Too low. Try again.  
?
```

<sup>1</sup> Если исходный код состоит из нескольких файлов (вы встретите такие программы в последующих главах), выбирайте в этом диалоговом окне все эти файлы сразу. Когда вы начнете писать программы самостоятельно, вы можете щелкнуть правой кнопкой мыши на папке `Source Files` и выбрать `Add > New Item...`, чтобы открыть диалоговое окно добавления нового файла.

### Шаг 8. Ввод второго числа

После следующего знака `?`, если первое число оказалось слишком маленьким, введите `750` и нажмите `Enter`; в противном случае введите `250` и нажмите `Enter`. Например, мы ввели `750`, и приложение вывело сообщение `"Too high. Try again."`, потому что введенное значение оказалось слишком большим:

```
I have a number between 1 and 1000.  
Can you guess my number?  
Please type your first guess.  
? 500  
Too low. Try again.  
? 750  
Too high. Try again.  
?
```

### Шаг 9. Ввод следующих чисел

Продолжайте игру, вводя разные числа, пока не угадаете то число, которое выбрано приложением. После этого вы увидите сообщение `"Excellent! You guessed the number!"`:

```
I have a number between 1 and 1000.  
Can you guess my number?  
Please type your first guess.  
? 500  
Too low. Try again.  
? 750  
Too high. Try again.  
? 625  
Too high. Try again.  
? 562  
Too low. Try again.  
? 593  
Too low. Try again.  
? 607  
Too low. Try again.  
? 616  
Too high. Try again.  
? 612  
Too low. Try again.  
? 614  
Too high. Try again.  
? 613  
  
Excellent! You guessed the number!  
Would you like to play again (y or n)?
```

### Шаг 10. Повторная игра или завершение работы приложения

Когда число будет угадано, приложение спросит, не хотите ли вы сыграть еще раз: `"would you like to play again (y or n)?"`. При нажатии клавиши `y` приложение вы-

берет другое случайное число и игра начнется сначала. При нажатии клавиши `n` приложение завершит работу.

### Повторное использование проекта для последующих примеров

Вы можете создавать отдельный проект для каждого приложения из книги, выполняя описанные в этом разделе действия. Или, возможно, вам будет удобнее работать с одним проектом, каждый раз удаляя из него файл с исходным кодом и добавляя вместо него следующий файл. Чтобы удалить файл из вашего проекта (но не из вашей системы), выберите этот файл в *Solution Explorer*, затем нажмите клавишу `Del` (*Delete*). Теперь вы можете повторить шаг 4, чтобы добавить другой файл в этот же проект.

### Работа с Ubuntu в подсистеме Windows для Linux

Некоторые пользователи Windows могут захотеть работать на своих Windows-компьютерах с компилятором `g++`, входящим в пакет *GNU Compiler Collection* (*GCC*). Это можно сделать двумя способами: использовать контейнер *Docker*, содержащий *GCC* (раздел 1.2.4), или запускать этот компилятор в операционной системе *Ubuntu*, установленной в подсистеме Windows для Linux. Чтобы установить подсистему Windows для Linux, следуйте инструкциям по ссылке:

<https://docs.microsoft.com/ru-ru/windows/wsl/install>

Установите в подсистеме Windows для Linux приложение *Ubuntu* (см. раздел «Перед началом работы»). Чтобы перейти в папку с файлом исходного кода приложения, введите в командной строке *Ubuntu* следующую команду:

```
cd /mnt/c/Users/ИмяПользователя/Documents/examples/ch01
```

и продолжите с шага 2 из раздела 1.2.3 («Компиляция и запуск приложения в Linux с помощью `g++`»).

## 1.2.2. Компиляция и запуск приложения в macOS с помощью Xcode

В этом разделе вы скомпилируете и запустите написанную на C++ программу в операционной системе macOS, используя интегрированную среду разработки *Apple Xcode*<sup>1</sup>.

### Шаг 1. Подготовка программного обеспечения

Если вы еще не сделали этого, прочтите раздел «Перед началом работы», установите *Xcode* и скачайте с сайта *Deitel* все наши примеры исходного кода.

### Шаг 2. Запуск Xcode

Откройте окно *Finder*, выберите *Applications* (Программы) и дважды щелкните на значке *Xcode*.



<sup>1</sup> Мы рассматриваем *Xcode* версии 13.2.1.

Если вы впервые запускаете Xcode, появится окно `Welcome to Xcode`. Закройте это окно. Вы сможете снова открыть его в любое время, выбрав `Window > Welcome to Xcode`. Мы будем обозначать символом `>` выбор пункта меню. Например, `File > Open` означает «выберите пункт `Open` в меню `File`».


### Шаг 3. Создание проекта

**Проект (project)** — это группа взаимосвязанных файлов (содержащих исходный код на языке C++), из которых состоит приложение. Все проекты Xcode, которые мы создали для этой книги, представляют собой программы командной строки, которые вы будете выполнять непосредственно в IDE. Для создания проекта:

1. Выберите `File > New > Project...`
2. В верхней части диалогового окна `Choose a template for your new project` щелкните на `macOS`.
3. В разделе `Application` выберите `Command Line Tool` и щелкните на кнопке `Next`.
4. В поле `Product Name` введите имя вашего проекта. В качестве имени мы указали `cpp20_test`.
5. В раскрывающемся списке `Language` выберите `C++` и щелкните на кнопке `Next`.
6. Укажите, где вы хотите сохранить проект. Мы выбрали папку `examples`, содержащую примеры из этой книги.
7. Щелкните на `Create`.

После создания проекта откроется главное окно Xcode, содержащее три области: слева — `Navigator`, в центре — `Editor` и справа — `Utilities`.

В верхней части `Navigator` находятся значки переключения панелей. Для работы с этой книгой вам будут нужны в основном две панели:

`Project` () — показывает все файлы и папки вашего проекта.

`Issue` () — показывает выдаваемые компилятором предупреждения и сообщения об ошибках.

При нажатии на значок отображается соответствующая панель.

Центральная область `Editor` предназначена для редактирования исходного кода и настройки параметров проекта. Она всегда присутствует в главном окне Xcode. Когда вы выбираете файл в панели `Project`, его содержимое отображается в `Editor`. Справа, в `Utilities`, обычно отображаются **инспекторы (inspectors)**. Например, если вы будете писать приложение для iPhone, содержащее сенсорную кнопку, то сможете настроить в этой области свойства кнопки (ее заголовок, размер, расположение и т. д.). При работе с этой книгой вы не будете использовать `Utilities`. Существует также область отладки `Debug`, в которой вы будете взаимодействовать с запущенной программой. Она появляется под областью `Editor`.

На верхней панели инструментов есть кнопки запуска и остановки программы, индикатор процесса выполнения задач, кнопки сокрытия и отображения левой (`Navigator`) и правой (`Utilities`) областей.

#### Шаг 4. Настройка проекта для работы с C++20

Входящий в состав Xcode компилятор Apple Clang поддерживает несколько версий стандарта C++. Для работы с этой книгой надо выбрать в параметрах проекта версию C++20:

1. В панели **Project** выберите название вашего проекта (`cpp20_test`).
2. В левой части **Editor** выберите название вашего проекта в разделе **TARGETS**.
3. В верхней части **Editor** выберите **Build Settings**, а чуть ниже нажмите **All**.
4. Перейдите к разделу **Apple Clang > Language > C++**.
5. Щелкните на значении справа от **C++ Language Dialect** и выберите **GNU++20 [-std=gnu++20]**.
6. Щелкните на значении справа от **C++ Standard Library** и выберите **Compiler Default**.

#### Шаг 5. Удаление файла `main.cpp` из проекта

По умолчанию Xcode создает файл `main.cpp`, содержащий исходный код простой программы, которая выводит строку "Hello, world!". В этом тест-драйве вам не понадобится файл `main.cpp`, поэтому удалите его. В панели **Project** сделайте «правый щелчок» (**Ctrl** + щелчок) на файле `main.cpp` и выберите **Delete**. В появившемся диалоговом окне нажмите **Move to Trash**. Файл не будет удален из вашей системы до тех пор, пока вы не очистите корзину.

#### Шаг 6. Добавление файла `GuessNumber.cpp` в проект

В окне **Finder** откройте папку `ch01` из папки `examples`, содержащей наши примеры кода, и перетащите файл `GuessNumber.cpp` в папку `cpp20_test` на панели **Project**. В появившемся диалоговом окне убедитесь, что установлен флаг **Copy items if needed**, и нажмите кнопку **Finish**<sup>1</sup>.

#### Шаг 7. Компиляция и запуск проекта

Чтобы скомпилировать и запустить проект, то есть проверить приложение в работе, нажмите кнопку запуска (▶) на панели инструментов Xcode. Если исходный код компилируется без ошибок, Xcode откроет область **Debug** и выполнит в ее правой половине приложение, которое выведет строку "Please type your first guess." и вопросительный знак (?) в качестве запроса данных:

```
I have a number between 1 and 1000.  
Can you guess my number?  
Please type your first guess.  
?
```

<sup>1</sup> Если исходный код состоит из нескольких файлов (вы встретите такие программы в последующих главах), перетащите все эти файлы в папку проекта. Когда вы начнете писать программы самостоятельно, вы можете сделать «правый щелчок» на папке проекта и выбрать **New File...**, чтобы открыть диалоговое окно добавления нового файла.

### Шаг 8. Ввод первого числа

Щелкните мышью в области `Debug`, введите `500` и нажмите `Return`. Ответ приложения будет меняться при каждом его запуске. У нас приложение вывело сообщение "Too high. Try again.", указывающее, что введенное число больше случайного числа, которое надо угадать:

```
I have a number between 1 and 1000.  
Can you guess my number?  
Please type your first guess.  
? 500  
Too high. Try again.  
?
```

### Шаг 9. Ввод второго числа

После следующего знака `?`, если первое число оказалось слишком маленьким, введите `750` и нажмите `Return`; в противном случае введите `250` и нажмите `Return`. Например, мы ввели `250`, и приложение вывело сообщение "Too high. Try again.", потому что введенное значение оказалось слишком большим:

```
I have a number between 1 and 1000.  
Can you guess my number?  
Please type your first guess.  
? 500  
Too high. Try again.  
? 250  
Too high. Try again.  
?
```

### Шаг 10. Ввод следующих чисел

Продолжайте игру, вводя разные числа, пока не угадаете то число, которое выбрано приложением. После этого вы увидите сообщение "Excellent! You guessed the number!":

```
I have a number between 1 and 1000.  
Can you guess my number?  
Please type your first guess.  
? 500  
Too high. Try again.  
? 250  
Too high. Try again.  
? 125  
Too low. Try again.  
? 187  
Too high. Try again.  
? 156  
Too high. Try again.  
? 140  
Too low. Try again.
```

```
? 148
Too high. Try again.
? 144
Too high. Try again.
? 142
Too low. Try again.
? 143

Excellent! You guessed the number!
Would you like to play again (y or n)?
```

### Шаг 11. Повторная игра или завершение работы приложения

Когда число будет угадано, приложение спросит, не хотите ли вы сыграть еще раз: "would you like to play again (y or n) ?". При нажатии клавиши *y* приложение выберет другое случайное число и игра начнется сначала. При нажатии клавиши *n* приложение завершит работу.

### Повторное использование проекта для последующих примеров

Вы можете создавать отдельный проект для каждого приложения из книги, выполняя описанные в этом разделе действия. Или, возможно, вам будет удобнее работать с одним проектом, каждый раз удаляя из него файл с исходным кодом и добавляя вместо него следующий файл. Чтобы удалить файл из вашего проекта (но не из вашей системы), сделайте «правый щелчок» (Ctrl + щелчок) на имени файла в панели Project и выберите Delete. В появившемся диалоговом окне нажмите Remove Reference. Теперь вы можете повторить шаг 6, чтобы добавить другой файл в этот же проект.

## 1.2.3. Компиляция и запуск приложения в Linux с помощью g++

В этом разделе вы скомпилируете и запустите написанную на C++ программу в командной оболочке Linux, используя компилятор g++ из пакета GNU Compiler Collection (GCC)<sup>1</sup>. Мы предполагаем, что вы уже прочитали раздел «Перед началом работы» и поместили все наши примеры исходного кода в каталог Documents/examples, расположенный в каталоге вашей учетной записи пользователя.

### Шаг 1. Переход в каталог ch01

Выполните команду `cd` в командной оболочке Linux, чтобы перейти в каталог `ch01` (первый пример из каталога `examples`):

```
~$ cd ~/Documents/examples/ch01
~/Documents/examples/ch01$
```

<sup>1</sup> Мы рассматриваем g++ версии 11.2. Чтобы узнать номер вашей версии компилятора g++, выполните команду `g++ --version`. Если у вас более старая версия g++, найдите в интернете инструкции по обновлению GNU Compiler Collection (GCC) для вашей системы Linux или используйте контейнер Docker с GCC, описанный в разделе 1.2.4.

В этом разделе мы будем выделять в командах жирным шрифтом те данные, которые вводит пользователь. **Приглашение (prompt)** в оболочке Ubuntu Linux обозначает тильдой (~) домашний каталог. Приглашение обычно заканчивается знаком доллара (\$), но может отличаться в зависимости от вашей системы Linux.

## Шаг 2. Компиляция приложения

Перед запуском приложения надо скомпилировать его следующей командой компилятора g++<sup>1</sup>:

```
~/Documents/examples/ch01$ g++ -std=c++20 GuessNumber.cpp -o GuessNumber
~/Documents/examples/ch01$
```

В этой команде:

- ✦ Параметр `-std=c++20` указывает компилятору, что мы используем C++20.
- ✦ Параметр `-o` присваивает исполняемому файлу имя (`GuessNumber`), которое затем будет нужно для запуска программы. Если вы не включите этот параметр, g++ автоматически присвоит исполняемому файлу имя `a.out`.

## Шаг 3. Запуск приложения

Введите `./GuessNumber` в командной строке и нажмите Enter, чтобы запустить программу:

```
~/Documents/examples/ch01$ ./GuessNumber
I have a number between 1 and 1000.
Can you guess my number?
Please type your first guess.
?
```

Символы `./` перед `GuessNumber` указывают системе Linux, что надо запустить программу `GuessNumber` из текущего каталога.

## Шаг 4. Ввод первого числа

Приложение выведет сообщение "Please type your first guess." и вопросительный знак (?) в качестве запроса данных. Введите `500` (ответ приложения будет меняться при каждом его запуске):

```
~/Documents/examples/ch01$ ./GuessNumber
I have a number between 1 and 1000.
Can you guess my number?
Please type your first guess.
? 500
Too high. Try again.
?
```

<sup>1</sup> Если на вашем компьютере установлено несколько версий g++, то нужную версию можно запустить командой `g++-##`, где ## — номер версии. Например, команда `g++-11` запускает последнюю на сегодняшний день версию 11.x.

У нас приложение вывело сообщение "Too high. Try again.", указывающее, что введенное число больше случайного числа, которое надо угадать.

### Шаг 5. Ввод второго числа

После следующего знака ?, если первое число оказалось слишком маленьким, введите 750 и нажмите Enter; в противном случае введите 250 и нажмите Enter. Например, мы ввели 250, и приложение вывело строку "Too high. Try again.", потому что введенное значение оказалось слишком большим:

```
~/Documents/examples/ch01$ ./GuessNumber
I have a number between 1 and 1000.
Can you guess my number?
Please type your first guess.
? 500
Too high. Try again.
? 250
Too high. Try again.
?
```

### Шаг 6. Ввод следующих чисел

Продолжайте игру, вводя разные числа, пока не угадаете то число, которое выбрано приложением. После этого вы увидите сообщение "Excellent! You guessed the number!":

```
? 125
Too high. Try again.
? 62
Too low. Try again.
? 93
Too low. Try again.
? 109
Too high. Try again.
? 101
Too low. Try again.
? 105
Too high. Try again.
? 103
Too high. Try again.
? 102

Excellent! You guessed the number!
Would you like to play again (y or n)?
```

### Шаг 7. Повторная игра или завершение работы приложения

Когда число будет угадано, приложение спросит, не хотите ли вы сыграть еще раз: "Would you like to play again (y or n)?". При нажатии клавиши y приложение выберет другое случайное число и игра начнется сначала. При нажатии клавиши n приложение завершит работу.