**Программируем самостоятельно**

Вы прошли немалый путь и научились программировать в нашем тренажёре. Теперь расскажем, как делать это самостоятельно на своём компьютере. Если вы пока не чувствуете себя готовым к экспериментам с C++, то можете смело пропустить этот урок и продолжать заниматься в тренажёре. Этот урок — обзорный. Подробные инструкции по установке и использованию каждого инструмента легко найти самостоятельно.

Как вы, возможно, помните, чтобы запустить код C++, его нужно предварительно скомпилировать. Для этого нужен компилятор. В отличие от многих других языков программирования, тут не существует канонического, главного или основного компилятора. C++ — это стандарт, описывающий синтаксис языка. C++ — это не программа, которую можно скачать и установить на компьютер.

Поэтому, чтобы начать самостоятельный путь программиста, вам нужно в первую очередь выбрать компилятор. Есть три кита — три главных компилятора, которые позволяют использовать все современные возможности C++:

* GCC (читается «джи-си-си») — компилятор по умолчанию под многими Linux-системами. Он имеет богатую историю, но не устарел, а, напротив, успевает за всеми нововведениями языка, оставаясь востребованным компилятором. Запускается командой g++ в терминале системы. Доступен под Windows в составе пакета MinGW.
* clang (читается «клэнг») — часто используется под macOS и Linux. clang — реализует только первый этап компиляции, так называемый фронтенд. Второй этап, получение исполняемого файла, делает программа LLVM. Запускается командой clang или c++ в терминале. Можно скачать LLVM с clang и на Windows.
* Visual C++ (читается «вижуал си-плюс-плюс») — компилятор под Windows. В качестве бонуса к нему прилагается удобная среда разработки Visual Studio, которую можно скачать бесплатно. Последнее время Visual C++ улучшился и по качеству не уступает двум своим конкурентам. Команда запуска в терминале — cl.

Все эти компиляторы бесплатные, к тому же вы можете ознакомиться с исходным кодом GCC и clang. Написаны они, к слову, на самом C++.

Самый простой способ скомпилировать программу — это написать её код в текстовом редакторе (например, [gedit](https://wiki.gnome.org/Apps/Gedit) или Блокнот), сохранить в файл с расширением .cpp и запустить компилятор так: <команда компиляции> <название файла>.cpp. Если ошибок нет, то создастся исполняемый файл, который можно запустить.

Но мало кто из программистов работает в текстовом редакторе. В основном используют IDE — мощные пакеты инструментов для запуска и компиляции программ, работы с исходным кодом. Современные IDE подсвечивают синтаксис, могут моментально находить ошибки, дополнять вводимое, быстро переходить к объявлению функции или переменной. Они показывают дополнительную информацию о программе, анализируют код, создают деревья для навигации по нему. Прямо в IDE можно скомпилировать и запустить программу, а затем пройти её по шагам, наблюдая, как меняются переменные и какие функции вызываются. Этот процесс называется «**отладка»**.

Мир IDE гораздо более многообразен, чем мир компиляторов. Рассмотрим две из них:

* Microsoft Visual Studio. Её плюс — компилятор уже встроен, ничего не нужно устанавливать дополнительно и настраивать. Её минус — работает только на Windows. Версия Community доступна бесплатно.
* В этом курсе будет рассматриваться среда Eclipse CDT. Она тоже бесплатная, но установить её несколько сложнее, ведь нужно ещё связать IDE с компилятором. Установка Eclipse CDT и компилятора GCC на разные платформы подробно описана в основном курсе.

Есть ещё много сред разработки: QT Creator, Clion, Visual Studio Code, Code Blocks. Выберите себе по вкусу. Ведь главное, чтобы вам было комфортно и ничто не отвлекало от программирования!