Прежде чем мы продолжим изучение языка C++, давайте изучим, как настроить рабочее окружение, чтобы вы могли сразу как-то экспериментировать с теми примерами, которые мы вам показываем, и уже получать какие-то практические навыки работы с языком программирования C++. В этом видео мы посмотрим, как настроить рабочее окружение в операционной системе Windows. Мы говорили, что C++ — это компилируемый язык, поэтому наша настройка будет состоять из двух этапов. Мы, во-первых, поставим компилятор, установим компилятор на нашу систему, а во-вторых, установим и настроим интегрированную среду разработки «Эклипс», чтобы вам было удобнее создавать программы, выполнять их и отлаживать. Итак, давайте начнем с того, что установим компилятор. Для этого запустим браузер и в поисковой строке введем запрос: mingw64 download. MinGW64 — это компилятор, который мы будем использовать под Windows. И перейдем по первой ссылке на проект Sourceforge, с которого, собственно, надо скачать установщик компилятора MinGW. Ну и да, я не сказал, что мы сейчас будем ориентироваться на 64-битную версию Windows и устанавливать все для нее. Для 32-битной, я думаю, вы разберетесь, как сделать аналогично. Итак, мы скачали наш установщик, теперь мы его запускаем. Браузер нам уже не нужен. И здесь мы выбираем версию компилятора, которая нам нужна, самая свежая — 6.3. Архитектура x86\_64, все остальное оставляем, как есть. Так, теперь нас спрашивают, куда поставить компилятор. Лучше всего указать путь, не содержащий пробелов. Поэтому я поставлю его в папку dev, и в ней создаем папку mingw64. Не нужны нам ярлыки на рабочем столе. Мы запустили инсталлятор, а сейчас он скачает все необходимые файлы и установит их. Итак, у нас установка компилятора успешно завершилась. Давайте проверим, что у нас действительно все стало хорошо. Для этого перейдем в ту папку, куда мы уставили наш компилятор. Перейдем в папку bin — там, где находятся исполняемые файлы. И на самом деле, да, это лучше сделать из командной строки. Мы перейдем mingw64\bin и проверим, что компилятор, правда, запускается. Исполняемый файл компилятора называется g++, и узнаем его версию. Вот, все хорошо, у нас все выполнилось, версия компилятора — 6.30. Собственно, готово. Мы с вами установили компилятор. Единственное, что нам надо еще сделать — это прописать путь до него в переменной среды path. Это существенно упростит нам установку и настройку «Эклипса». Поэтому давайте перейдем в изменение переменных среды, и здесь у меня уже была заготовка, я ее поправлю. Все, я в переменную среды path своего пользователя прописал полный путь до того места, где лежит компилятор языка C++. OK.

Итак, мы с вами успешно установили компилятор. Теперь давайте перейдем к установке интегрированной среды разработки «Эклипс». Особенностью «Эклипса» является то, что он работает, он написан на «Джаве», и, для того чтобы он работал, нужна виртуальная машина «Джавы». Поэтому начнем мы с того, что установим ее. Снова пойдем в «Поиск» и введем запрос: java runtime environment 64 bit. Выберем с вами сайт Oracle и попадем на страницу загрузки исполняемой среды «Джавы». Здесь примем соглашение и выберем оффлайн-установщик для Windows 64. Давайте его загрузим. Готово, установщик «Джавы» загружен, давайте его выполним и установим «Джаву» на наш компьютер. [БЕЗ\_ЗВУКА] Нажимаем Install и ждем, когда «Джава» установится. Готово. Мы установили виртуальную машину «Джава» на наш компьютер. Теперь давайте перейдем к установке «Эклипса». Откроем поисковую систему, введем «Эклипс». Открылось. Здесь мы выбираем Download, и выберем Download Packages, потому что нам «Эклипс» изначально создавался как среда для разработки на «Джава», но у него есть версия для разработки на C++. Вот она нам и нужна, поэтому мы выберем Download Packages и здесь найдем Eclipse IDE for C/C++ Developers. Вот именно он нам и нужен. Выбираем Windows 64 бита и скачиваем файл. Готово, «Эклипс» скачался. Давайте откроем папку, в которую попал этот файл. «Эклипс» скачивается просто архивом, и он не требует установки. Этот архив достаточно просто распаковать. Давайте это сделаем. Готово, архив распаковался. Зайдем внутрь, здесь есть папка Eclipse. И давайте для удобства мы ее, эту папку, скопируем туда, куда мы установили компилятор — в папочку dev. [БЕЗ\_ЗВУКА] В принципе, не важно, откуда мы будем запускать «Эклипс», но я предпочитаю, чтобы был порядочек и все разработческие инструменты лежали в одном месте. У меня это папочка под названием dev, от слова development. [БЕЗ\_ЗВУКА] Готово, мы скопировали папку, теперь можем запускать «Эклипс». Открываем ее, находим здесь исполняемый файл eclipse.exe и запускаем. У нас «Эклипс» запускается, первый его запуск может занимать какое-то продолжительное время, потому что он делает там какие-то свои настройки. Готово. Первое, что у нас спрашивает — это где будет находится workspace. Workspace в «Эклипсе» — это такая папка, где лежат все ваши проекты и настройки, ваши настройки «Эклипса». Давайте согласимся на предложенный путь по умолчанию в корневой папке нашего пользователя. Окей. Итак, готово, «Эклипс» у нас запустился, мы с вами его успешно установили.

[БЕЗ\_ЗВУКА] Теперь давайте перейдем к следующему этапу. Давайте создадим проект в интегрированной среде «Эклипс» и добьемся того, чтобы он у нас скомпилировался и созданная программа запустилась. Давайте посмотрим на приветственный экран и увидим, что здесь есть пункт меню Create a new C++ project. Это, собственно, то, что нам нужно. Выбираем его и здесь есть, значит, специальный диалог, который нам помогает. Давайте назовем наш проект hello и выберем Hello World C++ Project. Это для нашего варианта наиболее подходящий шаблон проекта, потому что он нам сразу создаст в проекте файл и с функцией main. И вот здесь справа есть такая колонка Toolchains, и здесь вот есть MinGW, вот давайте выберем MinGW GCC — это как раз тот самый компилятор, который мы с вами установили. И нажмем Finish. У нас создается проект, Да, проект у нас создался, но чтобы его увидеть, надо закрыть вот этот стартовый экран. Итак, готово. «Эклипс» нам создал проект, в котором есть программа Hello world!. Мы с вами уже рассматривали, как ее написать. Здесь он нам написал целую стопку комментариев, вот давайте мы их уберем, они нам не нужны. Сохраним наш проект и убедимся, что он компилируется. Для этого мы пойдем в панель инструментов и увидим здесь символ молоточка. Вот он. Собственно, давайте запустим сборку. Запускаем сборку, видим окно и видим, что у нас внизу в панели ошибок ничего нет, значит, наша программа скомпилировалась. Можем еще заглянуть в консоль, тут тоже видно, что ошибок никаких нет. Ну и прекрасно, теперь осталось нам нашу программу запустить. Для этого у нас есть специальная иконка для запуска. Нажмем на нее, зайдем в консоль и увидим, что да, действительно, на экран вывелось сообщение Hello World!, то есть мы с вами все правильно настроили. У нас установлен компилятор, «Эклипс» его увидел, корректно с ним связался, и смог скомпилировать нашу простую программу, и вывел на консоль Hello World!. Конечно же, в процессе разработки компилировать с помощью нажимания на иконочку неудобно, и запускать программу тоже, нажимая на пункт меню, неудобно. Для этого, естественно, в «Эклипсе» есть горячие клавиши. Для сборки это Ctrl + B, а для запуска приложения это Ctrl + F11. Теперь давайте немножечко настроим нашу среду «Эклипс», чтобы в ней было поудобнее работать, потому что некоторые настройки по умолчанию иногда бывают неудобные. Настраивается он у нас в пункте меню Window Preferences. Давайте мы его немножечко распахнем. Значит, здесь какие есть важные вещи, которые надо сразу настроить? Во-первых, пойдем в пункт меню Workspace и там выберем пункт Save automatically before build, чтобы каждый раз, когда вы запускаете сборку, «Эклипс» сохранял изменения, которые вы внесли в файл, потому что по умолчанию он этого не делает и кажется, что вы что-то поправили, у вас, допустим, программа на компилировалась, вы что-то поправили и запускаете компиляцию, а она продолжает не компилироваться. Все из-за того, что «Эклипс» просто не сохранил ваши изменения перед сборкой. И еще такая вещь, которая лично мне очень мешает — это проверка правописания. Вот давайте мы ее отключим, я думаю, вам только легче будет от того, что «Эклипс» не будет проверять правописание текста в ваших строковых литералах. На самом деле, это две самые главные, на мой взгляд, настройки, которые нужно поменять в чисто установленном «Эклипсе». Давайте сделаем еще одну вещь: давайте проверим, что если в нашей программе есть какие-то ошибки, то она действительно не компилируется. Давайте, например, заменим cout на out, запустим компиляцию и увидим, что да, у нас действительно не компилируется: ошибка, говорит о том, что компилятор не знает, что такое out. Значит, все хорошо — мы корректно настроили «Эклипс».

Ну и наконец давайте посмотрим и изучим, как работать с отладчиком в Eclipse. Вот у нас есть программа, Hello World. Давайте я, наверное, чтобы было лучше видно, увеличу шрифт, потому что, возможно, шрифт по умолчанию мелковат. Мы пойдем сюда и увеличим шрифт, сделаем шрифт побольше. Вот. Хорошо. Служебные окна закроем. Они нам сейчас не нужны. И давайте, например, напишем какую-то программу, которая что-нибудь будет считать. Ну например, сумму чисел от 1 до n. Мы этот пример периодически используем в наших лекциях. Напишем for (int i = 1; i <= n [ЗВУК] Вот мы написали нашу программу, скомпилировали, она компилируется. И давайте ее отладим. Для этого мы запустим отладчик. Он у нас представлен в виде иконки жучка. Ну либо его можно запустить горячей клавишей F11. Давайте все-таки запустим по иконке, Eclipse немножечко подумает. И вот у нас он спрашивает, а переключить ли расположение окон с того, что у нас есть на специально подобранный набор окон для отладки. Ну, я считаю, что да, лучше это сделать, потому что там есть удобные окна, позволяющие просматривать текущее значение переменной. Ну и собственно вот, запустилась отладка и остановилась на первой команде нашей программы. Замечательно. Давайте мы пройдем по шагам эту программу. Значит, у нас вот здесь вот есть специальные иконки, позволяющие управлять выполнением программы в отладчике. Вот есть, например, Step Over, который выполняет программу шаг за шагом, не заходя внутрь функций. Вот давайте ей и воспользуемся. Вот мы нажали, у нас выполнение перешло с пятой строчки на шестую, и вот здесь вверху вы видите, что желтым загорелось значение переменной n, потому что она поменялась. Ну и вы можете видеть, что есть горячая клавиша F6 для Step Over. Вот ей я дальше буду пользоваться. Вот я нажал, мы перешли к циклу, и здесь можно видеть, как в процессе работы цикла вверху справа у нас меняются значения локальных переменных, и мы можем отслеживать изменение переменной суммы. Ну и все, мы посчитали нашу сумму, и чтобы завершить выполнение программы, мы можем нажать на вот такой зеленый треугольник. Все, программа выполнилась, мы ее... мы выполнили процесс отладки. Чтобы вернуться в привычный режим, вот здесь вверху справа можно нажать на иконку с надписью C/C++, и тогда мы вернемся в наш редактор. Ну и соответственно обратно в отладчик — здесь вот тоже есть иконка с жучком, выбираем ее и возвращаемся в отладчик. Таким образом мы можем переключаться между расположением окон для набора кода и для отладки. Итак, давайте подведем итог. Мы с вами настроили рабочее окружение под операционной системой Windows. Мы установили компилятор, MinGW, мы установили Eclipse, создали в Eclipse новый проект и посмотрели, как его компилировать и отлаживать.

Что надо отправлять на проверку?

Во всех заданиях по программированию надо отправить на проверку файл с исходным [**кодом**](https://www.coursera.org/learn/c-plus-plus-white/supplement/vFav3/chto-nado-znat-pieried-piervym-zadaniiem-po-proghrammirovaniiu) вашего решения.

Некоторые задачи могут содержать особые условия отправки решений, в таком случае они указаны в описании задачи. Например, в задачах на тему «Функции» надо отправлять файл, не содержащий функции main.

Как тестируется моё решение?

В первую очередь, файл с вашим решением компилируется. Если компиляция завершается с ошибкой, то тестирующая система выдаёт вам сообщение „Compile error: *<список ошибок компиляции>*“.

Если компиляция прошла успешно, то ваша программа запускается на наборе тестов. В большинстве задач ваша программа должна что-то считать из стандартного ввода и вывести что-то в стандартный вывод. Один тест — это один набор входных данных для вашей программы. После того, как она отработала, её стандартный вывод анализируется специальной проверяющей программой, которая проверяет его правильность и соответствие условию задачи.

Если на всех тестах ваша программа отработала правильно, то тестирующая система возвращает вам сообщение «Good job!».

Если же на каком-то из тестов ваша программа отработала неправильно, то тестирующая система возвращает сообщение, объясняющее, что работает неправильно. Примеры таких сообщений:

* «Failed case #5/16: time limit exceeded» — это сообщение означает, что ваша программа успешно прошла четыре теста, а на пятом работала слишком долго; всего в задаче 16 тестов. Если в условии задачи не оговорено иное, вашей программе даётся одна секунда на исполнение. Если вы видите такую ошибку, проверьте, не возникает ли в вашей программе бесконечный цикл. Более распространённой причиной такой ошибки является неэффективный алгоритм решения. Например, в теме про контейнер set подразумевается его использование в соответствующих задачах: если вы используете контейнер vector, то ваше решение рискует оказаться неэффективным и получить ошибку «time limit exceeded».
* «Failed case #5/16: unknown signal 1» — это сообщение говорит о том, что ваша программа завершилась с ненулевым [**кодом**](https://www.coursera.org/learn/c-plus-plus-white/supplement/vFav3/chto-nado-znat-pieried-piervym-zadaniiem-po-proghrammirovaniiu) возврата. Причины могут быть разными, например, деление на ноль или выход за границы вектора.
* «Failed case #5/16: (Wrong answer)» — ваша программа успешно завершилась, но вывела в стандартный вывод неправильный ответ.
* В задачах, в которых нужно реализовать заданную функцию или класс, тестирующая система может возвращать специфические для этой задачи сообщения.

Если ваша программа получила «Wrong answer», то для первых нескольких тестов тестирующая система сообщает вам тест, на котором это произошло. Однако в общем случае мы не показываем вам тест, на котором программа работает неправильно. В таком случае вам надо внимательно перечитать условие задачи, изучать свой [**код**](https://www.coursera.org/learn/c-plus-plus-white/supplement/vFav3/chto-nado-znat-pieried-piervym-zadaniiem-po-proghrammirovaniiu) и постараться самостоятельно найти ошибку, подбирая различные входные данные.

Что делать, когда задача решена?

Поздравляем! Продвигайтесь дальше. Но обратите внимание, что для большинства обязательных задач после их успешной сдачи становятся доступными решения. Прочитайте их: с их помощью вы узнаете более простые подходы к решению задач.

## адание по программированию: A + B

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

i

Похоже, это ваше первое задание по программированию.  Подробнее

|  |  |
| --- | --- |
| **Срок сдачи** | Сдайте это задание до июль 30, 11:59 вечера PDT |

1. [**Инструкции**](https://www.coursera.org/learn/c-plus-plus-white/programming/9LFl2/a-b)
2. [Моя работа](https://www.coursera.org/learn/c-plus-plus-white/programming/9LFl2/a-b/submission)
3. [Обсуждения](https://www.coursera.org/learn/c-plus-plus-white/programming/9LFl2/a-b/discussions)

На вход программе через стандартный ввод передаются два целых числа, по модулю не превышающие 100000. Выведите в стандартный вывод их сумму.

|  |  |
| --- | --- |
| **stdin** | **stdout** |
| 2 3 | 5 |
| -4 -9 | -13 |
| -1 2 | 1 |

### Пояснение

Если вы не знаете, как компилировать и запускать программы на C++, вы можете просмотреть видео из блока "Компиляция, запуск, отладка" и, настроив среду разработки, вернуться к этой задаче.