**Среда разработки (IDE)**

## 1. IDE (Интегрированная среда разработки)

IDE — это не просто текстовый редактор, а комплексный программный пакет, который объединяет в себе все необходимые инструменты для разработки программного обеспечения в едином интерфейсе. Это рабочее место программиста, где все инструменты настроены и находятся под рукой.

Ключевые компоненты современной IDE:

* Интеллектуальный текстовый редактор (включает в себя):
  + **Подсветку синтаксиса:** различные элементы кода (ключевые слова, типы, переменные) окрашиваются в разные цвета для лучшей читаемости.
  + **Автодополнение кода (IntelliSense):** система предлагает варианты для завершения кода (имена функций, переменных, классов) по мере ввода, значительно ускоряя разработку и снижая количество опечаток.
  + **Навигацию по коду:** быстрый переход к определению функции или класса, поиск всех ссылок на функцию.
  + **Рефакторинг:** безопасное переименование переменных, функций и классов во всем проекте, извлечение методов и т.д.
* **Встроенный компилятор (или интерпретатор):** возможность компилировать и/или запускать код без переключения на внешние инструменты.
* **Отладчик (Debugger):** мощный инструмент для поиска и исправления ошибок, позволяет как минимум:
  + **Выполнять программу пошагово** (с заходом в функции или обходом их).
  + **Устанавливать точки останова (Breakpoints)** — места, где выполнение программы приостанавливается для инспекции состояния.
  + **Условные точки останова:** Остановка происходит только при выполнении заданного условия.
* **Система сборки (Build System):** автоматизирует процессы компиляции и линковки, управляет зависимостями проекта. Позволяет настроить различные конфигурации сборки (Debug, Release).
* **Дополнительные инструменты:** интеграция с системами контроля версий (например, Git), менеджеры проектов, инструменты для работы с базами данных, профилировщики производительности и т.д.

## 2. Visual Studio Code: Редактор кода, претендующий на звание IDE

Изначально VS Code позиционируется как **редактор кода**, а не как полноценная IDE. Однако после соответствующей настройки он трансформируется в мощную и гибкую среду, чья функциональность сопоставима с традиционными IDE. Это “конструктор“, который вы собираете под свои нужды. Чтобы лучше изучить элементы “конструктора”, из которого состоит IDE, и был выбран VS Code.

Ключевые особенности VS Code:

* **Кроссплатформенность:** одинаково работает на Windows, Linux и macOS. Это позволяет унифицировать среду разработки на разных машинах.
* **Легковесность:** запускается и работает значительно быстрее, чем "тяжелые" IDE вроде Visual Studio или CLion, особенно на слабом железе.
* **Бесплатность и открытость:** полностью бесплатен с открытым исходным кодом.
* **Гибкость и модульность:** основная мощь заключается в системе расширений. Можно установить плагины под конкретный язык (C++, Python, Rust) или задачу (Git, Docker, SSH).
* **Интегрированный терминал:** позволяет работать с системами сборки и командной строкой, не покидая редактор. Можно открыть несколько терминалов одновременно.

## 3. Настройка окружения для C++

**3.1. Для Windows:**

***Установка Visual Studio Code:***

1. Перейдите на официальный сайт: <https://code.visualstudio.com/download>.
2. Скачайте и запустите установщик для Windows.
3. Во время установки рекомендуется отметить галочки:
   * "Добавить действие "Открыть с помощью Code" в контекстное меню проводника Windows" (позволяет открывать папки и файлы в VS Code прямо из Проводника).
   * "Зарегистрировать Code в качестве редактора для поддерживаемых типов файлов".

***Установка компилятора и отладчика C++ (рекомендуемый способ — MSYS2 + MinGW)***:

1. Скачайте установщик MSYS2 с <https://www.msys2.org/>.
2. Установите его, следуя инструкциям. В конце установки запустите MSYS2.
3. В открывшейся консоли выполните команду для установки компилятора MinGW:

pacman -S --needed base-devel mingw-w64-ucrt-x86\_64-toolchain

1. В процессе установки при запросе подтверждения введите Y и нажмите Enter.
2. **Важный дополнительный шаг:** Добавьте путь к компилятору в системную переменную PATH (например, C:\msys64\ucrt64\bin). Это позволит запускать g++ и gdb из любого места, включая терминал VS Code.
3. **Проверка установки:** Откройте новое окно Command Prompt или PowerShell и выполните:

g++ --version

Если вы видите информацию о версии — установка прошла успешно.

**3.2. Для Linux (Ubuntu):**

Процесс значительно проще, что является одним из плюсов разработки под Linux.

1. ***Установка VS Code*** (проще всего через пакетный менеджер snap):

sudo snap install --classic code

1. ***Установка инструментов для C++*** (компилятор, отладчик GDB, системная сборка) [чаще всего компилятор C++ уже установлен в системе по умолчанию]:

sudo apt update && sudo apt install build-essential gdb

1. Проверка установки:

g++ --version

**3.3. Установка расширения для C++ в VS Code**

Этот шаг одинаков для всех операционных систем.

1. На левой панели VS Code нажмите на иконку **Расширений** (или нажмите Ctrl+Shift+X).
2. В поиске введите C++.
3. Найдите и установите официальное расширение **"C/C++"** от **Microsoft**. Именно оно предоставляет ядро функциональности: IntelliSense, отладку, навигацию по коду.

## Создание первого проекта: "Hello World" и настройка сборки

Для создания первого проекта в VS Code необходимо последовательно выполнить следующие шаги:

1. **Создайте проект:** File -> Open Folder и откройте или создайте новую папку для вашего проекта.
2. **Создайте файл:** Нажмите на иконку создания файла в интерфейсе VS Code, назовите его hello\_world.cpp.
3. Напишите код, например:

#include <iostream>

using namespace std;

int main() {

cout << "Hello World!" << endl;

return 0;

}

1. **Решение проблемы с IntelliSense:** если <iostream> подсвечен красным, это означает, что расширение C++ не нашло компилятор. Наведитесь на него, нажмите на (Quick Fix) и выберите Edit "compilerPath" settings. В открывшемся окне в поле Compiler path выберите установленный компилятор.
2. **Сборка через терминал:**
   * Откройте встроенный терминал (Terminal -> New Terminal или Ctrl+`).
   * Выполните команду компиляции:

g++ hello\_world.cpp -o hello\_world.exe

* + Запустите программу:

./hello\_world.exe # на Linux: ./hello\_world

***Настройка кнопок для запуска и отладки:***

Чтобы не вводить команды вручную каждый раз, можно настроить кнопки в правом верхнем углу.

1. Нажмите на кнопку в виде треугольника справа сверху и выберите опцию **"Run C/C++ File"**.
2. В появившемся списке выберите компилятор g++.
3. **Что произошло?** VS Code автоматически создал в папке проекта папку .vscode, которая теперь содержит файл tasks.json. Этот файл содержит инструкции (задачи) для сборки вашего проекта. Его можно редактировать для добавления флагов компиляции (например, -Wall для включения всех предупреждений, -g для добавления отладочной информации).

## 5. Использование отладчика (Debugger)

Отладчик — незаменимый инструмент для поиска ошибок в логике программы. Базовое его использование заключается в следующих пунктах:

1. **Установите точку останова:** Щелкните левой кнопкой мыши слева от номера строки, где хотите приостановить выполнение (появится красная точка).
2. **Запустите отладку:** Нажмите кнопку **"Run and Debug"** (или F5). В первый раз VS Code создаст файл launch.json для конфигурации отладчика. Этот файл "общается" с tasks.json, чтобы сначала собрать проект, а потом запустить его в режиме отладки.
3. Панель управления отладчиком:
   * **Continue (F5):** продолжить выполнение до следующей точки останова.
   * **Step Over (F10):** выполнить следующую строку кода, не заходя в функции.
   * **Step Into (F11):** выполнить следующую строку, заходя в тело функции.
   * **Step Out (Shift+F11):** выйти из текущей функции и вернуться на уровень выше.
   * **Restart (Ctrl+Shift+F5):** перезапустить сеанс отладки.
   * **Stop (Shift+F5):** завершить отладку.
4. Наблюдение за состоянием программы:
   * **VARIABLES:**показывает все локальные и глобальные переменные в текущей области видимости.
   * **WATCH:** позволяет вручную добавить выражения для отслеживания (например, i \* 2 или array[5]).
   * **CALL STACK:** показывает цепочку вызовов функций, которая привела к текущей точке выполнения.

## 6. Дополнительные возможности и расширения VS Code:

VS Code обладает огромным количеством расширений и опций, часть из которых могут показаться полезными Вам для разработки:

***Для C++ разработки:***

* **CMake Tools (расширение):** если ваш проект использует систему сборки CMake, это расширение практически необходимо. Оно позволяет:
  + Автоматически конфигурировать и собирать CMake-проекты.
  + Выбирать нужную конфигурацию сборки (Debug/Release).
  + Запускать и отлаживать целевые исполняемые файлы прямо из интерфейса VS Code.
* **Clangd (альтернатива стандартному C/C++ расширению):** мощный Language Server от LLVM, который часто предоставляет более точное и быстрое автодополнение, подсветку ошибок. Работает на основе файла compile\_commands.json, который можно сгенерировать с помощью CMake или другими системами сборки.
* **C++ TestMate (расширение):** позволяет удобно запускать Unit-тесты (Google Test, Catch2 и др.) прямо из дерева файлов в VS Code, с отображением зеленых/красных галочек рядом с тестами.
* **Doxygen Documentation Generator (расширение):** упрощает создание документации в формате Doxygen прямо в коде, генерируя шаблоны комментариев.
* И многое другое.

***Общие дополнительные расширения и опции:***

* **Интегрированный контроль версий (git):**
  + На панели управления версиями видны все измененные файлы.
  + Встроенный diff-инструмент позволяет наглядно видеть изменения в коде.
  + Возможность делать коммиты, создавать ветки, пушить изменения без использования командной строки.
* **Multi-Cursor (Множественный курсор):**
  + Ctrl+D выделяет следующее вхождение слова под курсором.
  + Alt+Click ставит курсор в любое место.
  + Позволяет одновременно редактировать несколько строк или изменять однотипные переменные в разных местах.
* **Emmet и Сниппеты (Шаблоны кода):**
  + Набираете аббревиатуру (например, for), нажимаете Tab, и разворачивается полный шаблон цикла for. Можно создавать и свои собственные сниппеты.
* **Breadcrumbs (Навигационная цепочка):**
  + Под строкой заголовка файла отображается "путь" до текущего символа в коде (Файл -> Класс -> Функция), позволяя быстро перемещаться по структуре файла.
* **Peek View (Быстрый просмотр):**
  + Нажмите Alt+F12 на имени функции, чтобы посмотреть ее определение во всплывающем окне, не покидая текущее место в коде.
* **Command Palette (Палитра команд):**
  + Главный инструмент управления, для открытия которого необходимо нажать Ctrl+Shift+P. Позволяет выполнять любую команду VS Code (сменить тему, перезагрузить окно, запустить задачу, настроить настройки) без использования мыши и меню.
* **Settings Sync (Синхронизация настроек):**
  + Позволяет залогиниться с помощью GitHub аккаунта и синхронизировать все ваши настройки, сниппеты, установленные расширения на разных машинах.
* **Remote Development (Расширение):**
  + Позволяет работать с кодом, который находится в контейнере Docker, на удаленной машине по SSH или в подсистеме Windows для Linux (WSL) так, как будто он локальный. Все расширения и функциональность VS Code работают на удаленной стороне.
* **Live Share (Расширение):**
  + Позволяет совместно редактировать код в реальном времени с другими разработчиками. Каждый участник сессии имеет свой курсор, может видеть изменения других и даже совместно использовать отладчик и терминал.
* И многое другое.

## 7. Полезные ссылки:

* Официальное руководство по настройке C++ в VS Code: <https://code.visualstudio.com/docs/cpp/config-mingw>
* Документация по VS Code: <https://code.visualstudio.com/docs>
* Магазин расширений: <https://marketplace.visualstudio.com/VSCode>