

Задание №2

Clion

CLion — это интегрированная среда разработки (IDE) от компании JetBrains, предназначенная для разработки на языках C и C++.

1) Помимо C и C++ поддерживает Python, Rust, Swift (и другие языки программирования)

2) Присутствует Умное автодополнение (что позволяет быстрее и удобнее писать код) и рефакторинг (см. Приложение 1).

3) Сборка с помощью утилиты Cmake (см. приложение 2)

4) функция Debug (отладка кода)

5) Подходит для операционных систем:

1. Windows,

2. macOS,

3. Linux.

6) поддерживается удаленная разработка без дополнительных инструментов и других ПО

7. Интеграция с другими инструментами JetBrains.

8. Настраиваемый интерфейс.

для работы с CLion необходимы следующие системные требования:

Оперативная память: минимум 2 ГБ, рекомендуется 8 ГБ.

Свободное место на жёстком диске: минимум 3,5 ГБ, рекомендуется SSD.

Разрешение экрана: минимум 1024x768.

Функции

Почему же согласно статистике 14,66% все разработчиков используют CLion как среду разработки (см. приложение 3). В нем присутствуют следующие функции:

1) Создание проектов на C/C++

- 2) Редактирование кода
- 3) Сборка/запуск/отладка проектов
- 4) Статический анализ кода
- 5) Возможность удаленной отладки
- 6) Разработка под встраиваемые системы (IoT)
- 7) Поддержка пакетных менеджеров (Conan, vcpkg)
- 8) Написание и отладка вспомогательных скриптов на Python, Javascript, Typescript
- 9) Работа с HTML и CSS
- 10) Полная поддержка Docker-контейнеров
- 11) Поддержка фреймворков для тестирования
- 12) Интеграция систем контроля версий (Git)
- 13) Совместная разработка (Code with me)
- 14) Интегрированы системы контроля работы с динамической памятью (Valgrind)
- 15) Расширение функционала за счет установки плагинов от JetBrains и сторонних разработчиков
- 16) Интегрирована работа с базами данных
- 17) Средства непрерывного развертывания проектов

Приложения

```

int main() {
    int **matrix;
    int rows, cols;
    printf(_Format: "input rows:");
    scanf(_Format: "%d", &rows);
    printf(_Format: "input cols:");
    scanf(_Format: "%d", &cols);

    int res = memoryh(&matrix, rows, cols);
    if (res != 0) {
        printf(_Format: "error memory\n");
        return res;
    }

    ge(&matrix, rows, cols);

```

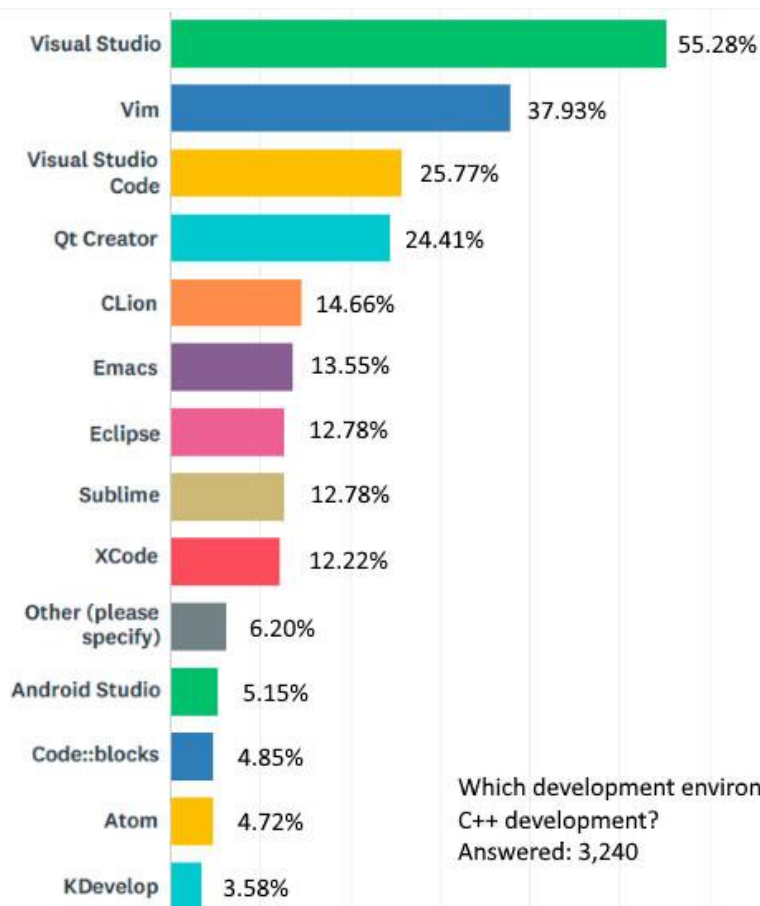
f	genmatrix(int ***matrix, int rows, int cols)	void
f	getc(FILE *_Stream)	int
f	getchar()	int
f	getenv_s(size_t *_RequiredCount, char *_Buffer, ...)	errno_t
f	gets_s(char *_Buffer, rsize_t _Size)	char *
f	getwc(FILE *_Stream)	wint_t
f	getwchar()	wint_t
f	getenv(const char *_VarName)	char *
f	getw(FILE *_Stream)	int
f	_get_doserrno(unsigned long *_Value)	errno_t
f	_get_errno(int *_Value)	errno_t
f	get_fmode(int *_PMode)	errno_t

Press Enter to insert, Tab to replace

```

CMakeLists.txt x  main.c x
1  # cmake_minimum_required(VERSION <specify CMake version here>)
2  project(untitled C)
3
4  set(CMAKE_C_STANDARD 99)
5
6  add_executable(untitled main.c)

```



Which development environments (IDEs) or editors do you use for C++ development?
Answered: 3,240