

Задание 1.1 (УТП)

«Обзор программного продукта: JetBrains CLion»

Подготовили:

**Симонов Владимир
Каменский Ярослав**

Преподаватель:

Жуков Н.Н.

1. Общая характеристика

JetBrains CLion — это мощная интегрированная среда разработки (IDE), ориентированная на разработку программного обеспечения на языке C и C++. CLion разработан компанией JetBrains и предоставляет все необходимые инструменты для разработки, отладки и тестирования C/C++ проектов. CLion также поддерживает языки, такие как Python, Rust, Swift, и другие, через плагины.

Основные особенности CLion включают:

- **Интеллектуальное автодополнение (IntelliSense)** для C/C++, анализ и рефакторинг кода.
- **Графический интерфейс отладки** с поддержкой различных отладчиков (GDB, LLDB).
- **Поддержка сборки проектов** через системы CMake, Gradle и Makefile.
- **Интеграция с системами контроля версий** (Git, SVN, Mercurial).
- **Поддержка юнит-тестирования** с использованием Google Test, Boost.Test и Catch2.
- **Удобные инструменты для профилирования и мониторинга производительности.**

CLion позволяет разработчикам сосредоточиться на программировании, минимизируя рутинные задачи благодаря множеству встроенных инструментов и интеллектуальным функциям, которые автоматизируют процесс разработки.

2. Необходимое для работы программное и аппаратное обеспечение (requirements)

Операционные системы:

- **Windows:** 10, 11.
- **macOS:** 10.14 и новее.
- **Linux:** любой дистрибутив с поддержкой 64-разрядной архитектуры, включающий glibc версии 2.17 и выше.

Минимальные требования:

- **Оперативная память (RAM):** минимум 4 ГБ (рекомендуется 8 ГБ и выше для комфортной работы).
- **Свободное пространство на диске:** минимум 2,5 ГБ для установки программы и около 1 ГБ для работы проекта.
- **Процессор:** поддержка 64-битных процессоров с минимум двумя ядрами.
- **Программное обеспечение:**
 - Компиляторы: GCC, Clang, MSVC (для Windows).
 - Для отладки: GDB, LLDB.
 - Java Runtime Environment (JRE) встроен в дистрибутив CLion и не требует отдельной установки.

Интеграции:

- Поддержка Docker, Vagrant, WSL (Windows Subsystem for Linux).
- Интеграция с сервисами CI/CD, такими как GitLab CI, Jenkins.

3. Функции

3.1 Создание проекта

CLion поддерживает создание проектов как через систему CMake, так и через Makefile и Gradle. Пользователи могут быстро создать новый проект с помощью встроенных шаблонов для C/C++ или импортировать существующие проекты, используя системы сборки.

3.2 Кодирование

- **Интеллектуальное автодополнение:** CLion предоставляет интеллектуальное автодополнение кода, ускоряя процесс написания кода.
- **Статический анализ:** встроенные инструменты анализа помогают находить ошибки, предупреждения и недочёты в коде.
- **Шаблоны кода:** можно быстро вставлять часто используемые конструкции с помощью горячих клавиш или настраиваемых шаблонов.

3.3 Форматирование кода

CLion поддерживает автоматическое форматирование кода согласно стандартам (например, Google C++ Style Guide). Также есть возможность кастомизации правил форматирования через настройки IDE.

3.4 Отладка

- **Графический интерфейс отладки:** CLion предоставляет мощные инструменты для отладки на базе GDB или LLDB. Поддерживаются точки останова, просмотр переменных, стек вызовов, анализ памяти и многое другое.
- **Интеграция с Valgrind:** для профилирования и поиска утечек памяти.

3.5 Запуск и компиляция

CLion поддерживает запуск проектов как в конфигурации отладки, так и в конфигурации релиза. IDE использует CMake для управления процессом сборки и компиляции, но также поддерживает Makefile и другие системы сборки. Можно настроить разные конфигурации сборки для разных целей (например, Debug и Release).

3.6 Версионирование

CLion интегрируется с системами контроля версий, такими как Git, SVN и Mercurial. Поддерживаются все основные функции:

- Коммиты, слияния, разрешение конфликтов.
- Просмотр истории изменений файлов и проектов.
- Работа с ветками, создание Pull Request-ов прямо из IDE.

3.7 Публикация в репозитории

CLion позволяет интегрировать работу с удалёнными репозиториями, такими как GitHub, GitLab и Bitbucket. Поддерживается автоматическая отправка изменений в удалённые репозитории и создание pull request-ов через графический интерфейс.

3.8 Профилирование и мониторинг

- **Valgrind и другие профайлеры:** встроенная поддержка для поиска узких мест в производительности и анализа использования памяти.
- **Рефакторинг:** CLion предоставляет мощные инструменты для безопасного рефакторинга кода, включая переименование, извлечение функций, переменных и методов.

3.9 Интеграция с плагинами

JetBrains CLion поддерживает большое количество плагинов, которые позволяют расширить функциональность среды разработки. Плагины можно найти и установить через **JetBrains Marketplace**, доступный из интерфейса CLion. Вот несколько самых популярных плагинов:

1. Markdown Support

- **Функционал:** Этот плагин предоставляет полную поддержку редактирования и предпросмотра Markdown-файлов прямо в CLion. Он полезен для написания документации, README-файлов и других текстов в формате Markdown.
- **Ключевые особенности:** Включает подсветку синтаксиса, автодополнение и просмотр в реальном времени.

2. Docker

- **Функционал:** Плагин интегрирует Docker в CLion, позволяя разрабатывать, запускать и отлаживать контейнеризированные приложения. Это полезно, если ваши проекты зависят от контейнеров или вы хотите стандартизировать среду разработки.
- **Ключевые особенности:** Управление контейнерами Docker и образами, запуск контейнеров, отладка приложений внутри контейнеров.

3. C/C++ Coverage

- **Функционал:** Этот плагин помогает анализировать покрытие кода тестами, что важно для создания качественного программного обеспечения. Плагин работает с существующими фреймворками для тестирования C/C++.
- **Ключевые особенности:** Показывает, какие строки кода были протестированы, а какие — нет, улучшая качество и полноту тестирования.

4. CMake Plus

- **Функционал:** Расширяет возможности работы с CMake, основной системой сборки, используемой в CLion. Плагин предлагает удобные функции для автозаполнения и управления зависимостями в CMake-файлах.

- **Ключевые особенности:** Улучшенные предложения автодополнения для CMakeLists.txt, управление зависимостями проекта.

5. PlantUML Integration

- **Функционал:** Этот плагин позволяет создавать UML-диаграммы прямо в CLion с помощью языка PlantUML. Он помогает визуализировать структуры данных и архитектуру программных систем.
- **Ключевые особенности:** Автоматический рендеринг диаграмм, предпросмотр UML в реальном времени.

6. GitToolBox

- **Функционал:** Расширяет стандартные возможности Git в CLion, добавляя дополнительные функции и улучшая работу с системой контроля версий.
- **Ключевые особенности:** Показывает статус файлов в Git, автоматические операции с Git, поддержка множественных репозиториях.

Эти плагины делают CLion более гибкой средой для разработки, улучшая рабочие процессы и добавляя важные функции для конкретных задач.

Заключение

JetBrains CLion — это профессиональная IDE, предназначенная для эффективной разработки программ на C и C++. Она предлагает богатый набор инструментов для отладки, тестирования и анализа кода, что делает её незаменимым инструментом для разработчиков в организациях, где важна высокая производительность и качество кода.