Колледж Автономной некоммерческой образовательной организации высшего образования

«Научно-технологический университет «Сириус»

**Учебная дисциплина «Введение в специальность»**

**Реферат «Разработка приложений на C++»**

Преподаватель:

Яковлева Софья Вячеславовна

Работу подготовила:

Паньшина Виктория

Студент группы К0709-24/2

# Содержание

[Содержание 2](#_Toc191827864)

[Введение 3](#_Toc191827865)

[Цели реферата 4](#_Toc191827866)

[История развития С++ 5](#_Toc191827867)

[Плюсы С++ 6](#_Toc191827868)

[Минусы С++ 8](#_Toc191827869)

[Удобство проектирования и разработки приложений. 9](#_Toc191827870)

[Какие приложения написаны на С++ 10](#_Toc191827871)

[Заключение 11](#_Toc191827872)

[Список использованных источников 12](#_Toc191827873)

# Введение

C++ – мощный и универсальный язык программирования, который, несмотря на свою зрелость, остается одним из ключевых инструментов в создании высокопроизводительных и надежных приложений в самых разных областях. C++ остается актуальным и широко используется во многих областях информационных технологий благодаря высокой производительности, эффективности и возможности работать на различных платформах.  Этот язык программирования доминирует там, где критически важны скорость, контроль над ресурсами и масштабируемость. C++ подходит для разных задач, включая робототехнику, ядра ОС, драйверы, микроконтроллеры, приложения, нейросети и обработку статистики.

Сейчас С++ — один из самых популярных языков программирования. Например, в рейтинге TIOBE\* за февраль 2024 года он занял второе место (на первом — Python).

В рамках этого доклада мы рассмотрим преимущества и недостатки C++ в сравнении с другими языками программирования и практические примеры успешных проектов, реализованных на C++.

\* Индекс TIOBE — это рейтинг популярности языков программирования, обновляемый ежемесячно. Он основан на анализе поисковых запросов в интернете и учитывает данные из различных источников, таких как Google и Stack Overflow.

# Цели реферата

1. Обосновать актуальность C++;
2. Ознакомиться с историей развития C++;
3. Представить преимущества C++;
4. Обозначить недостатки C++;
5. Показать примеры успешных проектов на C++.

# История развития С++

С — процедурный язык программирования, разработанный в 1969—1973 годах сотрудниками Bell Labs (корпорация, крупный исследовательский центр в области телекоммуникаций, электронных и компьютерных систем) Кеном Томпсоном и Деннисом Макалистэйр Ритчи. С был создан для использования в операционной системе UNIX, ключевым разработчиком которой являлся Деннис Ритчи. С тех пор он был перенесён на многие другие операционные системы и стал одним из самых используемых языков программирования.

C++ был создан в начале 1980-х годов Бьерном Страуструпом, также сотрудником Bell Labs.  Изначально он разрабатывал язык, известный как "C с классами", чтобы улучшить возможности языка C, добавив в него объектно-ориентированные элементы, такие как классы и наследование. Это было вызвано необходимостью создания более эффективных инструментов для моделирования сложных систем, например, телефонных звонков.

В 1983 году язык получил название C++, где "++" символизирует увеличение функциональности по сравнению с C. Его первый коммерческий выпуск состоялся в октябре 1985 года.

С момента создания C++ стал широко использоваться для разработки программного обеспечения, став одним из самых популярных языков программирования. В 1985 году вышло первое издание книги «Язык программирования C++»

Б. Страуструпа, которое стало классическим.

# Плюсы С++

* Высокая производительность, потому что C++ не накладывает никакой избыточной нагрузки на программу, не использующую какие-либо возможности; Производительность является важной характеристикой для, например, компьютерной игры. Counter-Strike, StarCraft: Brood War, World of Warcraft — все они появились давно и были написаны на C++, как и [операционные системы](https://blog.skillfactory.ru/glossary/operaczionnaya-sistema/) консолей Xbox и PlayStation, ядра популярных игровых движков Unreal Engine или Unity;
* Поддержка множества стилей программирования (процедурное программирование, абстракцию данных, объектно-ориентированное программирование и обобщенное программирование). Поэтому разработчик может сам выбрать, в каком стиле ему писать программу;
* Кроссплатформенность. C++ поддерживает разработку кроссплатформенных приложений, что позволяет создавать приложения для различных операционных систем и устройств. Это особенно важно в современной игровой индустрии, где игры должны работать на ПК, консолях и мобильных устройствах. Кроссплатформенность C++ позволяет сократить время и усилия на портирование приложений на разные платформы. Например, разработчики могут использовать один и тот же код для создания игр, которые будут работать на Windows, macOS, Linux, PlayStation, Xbox и других платформах. Это значительно упрощает процесс разработки и позволяет быстрее выпускать продукт на рынок.
* Поддержка многопоточности. Современные приложения часто требуют выполнения множества задач одновременно, таких как обработка данных, выполнение вычислений и управление пользовательским интерфейсом. C++ предоставляет мощные инструменты для работы с многопоточностью, что позволяет эффективно распределять задачи между процессорными ядрами и улучшать общую производительность приложения.
* Большое сообщество. Большое сообщество разработчиков C++ всегда готово помочь с решением возникающих проблем и поделиться опытом. Форумы, такие как Stack Overflow, и специализированные сообщества, такие как Reddit, являются отличными ресурсами для получения помощи и обмена знаниями;
* Регулярное обновление (каждые 3 года).
* Существует множество библиотек, которые расширяют функциональность C++. Вот некоторые из них:
* Qt: Используется для разработки кроссплатформенных приложений с графическим интерфейсом. Qt предлагает широкий набор инструментов для работы с GUI (графический пользовательский интерфейс), сетевыми приложениями и базами данных.
* OpenCV: Библиотека для компьютерного зрения и обработки изображений, широко используемая в научных и промышленных приложениях.

C++ поддерживается множеством компиляторов, что позволяет разработчикам выбирать наиболее подходящий инструмент для своих задач:

* GCC (GNU Compiler Collection): Один из самых популярных компиляторов, поддерживающий множество языков, включая C++.
* Microsoft Visual C++ (MSVC): Часть среды разработки Visual Studio, предоставляет мощные инструменты для разработки приложений под Windows.
* Intel C++ Compiler: Оптимизированный компилятор от Intel, который предлагает улучшенную производительность для вычислительно интенсивных приложений.

Эти библиотеки и компиляторы делают C++ универсальным языком для решения самых разнообразных задач в программировании.

# Минусы С++

* Сложность разработки и время компиляции. Разработка на C++ может потребовать значительно больше времени из-за его сложности и необходимости тщательного управления памятью. Это приводит к более медленной скорости написания программного обеспечения по сравнению с языками, такими как Python. Кроме того, процесс компиляции в C++ может занять много времени, особенно в больших проектах. Чем больше кодовая база и чем сложнее проект, тем дольше компилятор будет преобразовывать код в исполняемый файл. Это может быть неудобно, если необходимо часто вносить изменения и проверять результаты.
* Трудности с памятью. В C++ нужно самостоятельно управлять памятью, что может быть сложно для новичков. Например, нужно вручную выделять и освобождать память, чтобы избежать ошибок, таких как утечка памяти. Иногда возможность управлять ресурсами полезна, но в некоторых случаях это создаёт лишние сложности. В любом случае на это придётся тратить дополнительное время.
* Сложный синтаксис. C++ имеет много сложных конструкций, таких как указатели, ссылки и перегрузка операторов, которые могут запутать по началу. Понимание всех этих аспектов требует времени и опыта.
* Программируя на C++, нужно держать в голове сразу много нюансов: отслеживать типы данных, контролировать выделение и освобождение памяти. Сейчас появляется все больше средств, которые перекладывают эти обязанности на сам язык. Уже на этапе написания кода современные IDE (среды разработки) помогают следить за соблюдением стандартов, отлавливают большое количество ошибок, не всегда очевидных человеку без глубокого знания языка, автоматизируют процесс написания большого объема кода с помощью типовых конструкций.

# Удобство проектирования и разработки приложений.

Современные IDE (среды разработки) значительно упрощают разработку на C++. Такие IDE как, например, Visual Studio предоставляют:

* Автодополнение кода: Ускоряет написание кода и снижает вероятность ошибок.
* Интегрированный отладчик: Облегчает поиск и исправление ошибок.
* Интеграция с системами контроля версий (Git): Облегчает совместную работу над проектами.
* Анализ кода: Обнаруживает потенциальные проблемы в коде, такие как утечки памяти или небезопасное использование указателей.

Стандарты C++ привносят новые возможности, которые облегчают разработку.

Например, лямбда-выражения упрощают создание функционального кода.

Широкий выбор библиотек и фреймворков значительно ускоряет разработку приложений. Такие библиотеки как Boost (для работы с многопоточностью), Dear ImGui (Для создания UI), SFML (Для создания игр).

и фреймворки позволяют повторно использовать готовые решения, сокращая время разработки и повышая качество кода.

C++ остаётся незаменимым для разработки операционных систем, драйверов и другого системного программного обеспечения, где важен прямой доступ к аппаратному обеспечению и оптимизация производительности.

C++ поддерживает объектно-ориентированный, процедурный и обобщенный стили программирования, что позволяет выбрать наиболее подходящий подход для конкретной задачи.

# Какие приложения написаны на С++

C++ позволяет создавать приложения и программы любой сложности: от простых консольных утилит до сложных игровых движков. Также на этом языке можно программировать микроконтроллеры и системы в реальном времени.

Многие известные приложения были написаны на C++, включая операционные системы Windows и OS X, многие игры, такие как World of Warcraft и Counter-Strike. На C++ можно написать практически любое программное обеспечение, от простых консольных утилит до сложных приложений. Благодаря C++ работают Unreal Engine 4, Microsoft Office и Adobe Photoshop.

С помощью C++ [Mozilla](https://github.com/mozilla) Corporation разработала opensource-браузер Firefox, была создана [Amadeus](https://corporate.amadeus.com/documents/en/resources/corporate-information/corporate-documents/global-reports/2019/amadeus-global-report-2019.pdf" \t "_blank) — система электронного бронирования авиабилетов, гостиниц, круизных туров и проката автомобилей (через нее проходит 5000 транзакций в секунду, более миллиарда транзакций ежедневно), благодаря C++ работают Яндекс.Поиск и поисковый [движок Google](http://infolab.stanford.edu/~backrub/google.html). У Spotify код бэкенда написан на C++, и он же используется при обработке и воспроизведении видео на YouTube. Большая часть кода профессионального графического редактора Photoshop, который разрабатывает Adobe, написана на C++, как и другие продукты компании.

# Заключение

Вопреки распространенным мифам, C++ не теряет своей актуальности. Он по-прежнему широко используется в областях, таких как игровая индустрия, разработка операционных систем, высокопроизводительные вычисления и встраиваемые системы.

Стандарты C++ развиваются, добавляя новые возможности, которые упрощают разработку и повышают безопасность кода. Современный C++ - это не тот C++, который учили 20 лет назад.

Экосистема C++ продолжает развиваться: Появляются новые библиотеки, фреймворки и инструменты, которые упрощают разработку на C++ и делают его более привлекательным для новых поколений программистов.

C++ - это язык для тех, кто готов учиться и углубляться в детали. Однако, усилия, затраченные на изучение C++, окупаются возможностью создавать мощные и эффективные приложения.

# Список использованных источников

# 

1. C++ // Знание.Вики URL: <https://znanierussia.ru/articles/C%2B%2B>
2. Кому и для чего нужен C++ // SKILLFAACTORYMEDIA URL: <https://blog.skillfactory.ru/c-plus-komu-i-dlya-chego-nuzhen-yazik/>
3. История возникновения языка программирования С++ // CppStudio URL: http://cppstudio.com/post/1984/
4. Стоит ли учить C++ // Tproger URL: <https://tproger.ru/articles/stoit-li-uchit-c-->
5. Преимущества и недостатки C++ // Skypro URL: <https://sky.pro/wiki/gamedev/preimushestva-i-nedostatki-c/>
6. Зачем изучать C++ // Практикум Блог URL: <https://practicum.yandex.ru/blog/yazyk-c-plus-dlya-chego-nuzhen/>
7. 22 важнейшие программы на C++ // Дзен URL: <https://dzen.ru/a/YASTuvgBHEoBqVUQ>