**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ университет»**

**Институт геологии и нефтегазодобычи**

**КАФЕДРА КИБЕРНЕТИЧЕСКИХ СИСТЕМ**

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

тип практики:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Ознакомительная**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Направленность: Автоматизированные системы обработки информации и управления

форма обучения: \_\_\_\_\_\_\_очная (4 года)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил студент гр. АСОиУб-21-1

Евлоев Алихан Алиханович

*(ФИО)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(подпись)*

**Проверил:**

*\_* доцент Сенкевич Людмила Борисовна*\_\_*

*(должность, ФИО руководителя практики от университета)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(оценка) (подпись)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

**Оглавление**

[Введение 3](#_Toc44129272)

[1. Актуальность 4](#_Toc44129273)

[2. Назначение и область применения 4](#_Toc44129274)

[3. Технические характеристики, стандарты 4](#_Toc44129275)

[4. Разработка сайта 5](#_Toc44129276)

[5 Средства разработки 7](#_Toc44129278)

[6 Описание сайта 8](#_Toc44129279)

Заключение……………………………………………………………………….12

[Список литературы 15](#_Toc44129280)

**ВВЕДЕНИЕ**

В начале 80-х годов прошлого века сотрудник Bell Labs Бьёрн Страуструп после долгих мучений с существующими языками программирования провёл эксперимент со скрещиванием C и Simula. Он даже не рассчитывал, что его детище, получившее название C++, привлечёт столько внимания.

Однако тогда язык произвёл настоящий фурор: компилируемый, структурированный, объектно-ориентированный, невероятно упрощающий работу с большими программами и при этом имеющий огромный потенциал для развития.

Такой, что ещё почти десятилетие потребовалось Страуструпу, чтобы наделить C++ всеми характерными особенностями. Развитие же продолжается до сих пор. К примеру, стандарт C++17 был принят в конце прошлого года.

Поэтому, если вы думаете, что C++ безнадёжно устарел, у нас для вас новость – отправить на пенсию его может только какая-нибудь невероятная технологическая революция. И вот почему.

Откройте любой тест производительности языков программирования (к примеру,) и в топе вы увидите C++, лишь С однозначно быстрее. Скорость исполнения кода — пожалуй, главный аргумент в пользу того, почему C++ был, есть и будет востребован в ИТ.

Да, для реализации локальных нужд могут подойти C, Go или Rust, но если вы хотите изучить один язык, с помощью которого можно писать программы любой сложности, ваш выбор — C++.

В противовес этому часто ставится скорость написания кода, которая, например, у интерпретируемых языков на порядок выше. В этом есть доля истины — C #, Java и конечно же Python даже визуально занимают меньше места, с их помощью можно создавать сложные программы, затратив минимум времени. Однако подумайте, что лучше для конечного пользователя: время разработки приложения или его медленная работа? Ответ очевиден.

**АКТУАЛЬНОСТЬ**

**Он универсален**

Компиляторы C++ есть на каждой операционной системе, большинство программ легко переносится с платформы на платформу, со средой разработки и библиотеками у вас точно не возникнет проблем. C++ — это демонстрация идеи классического программирования, когда 90% мыслей связано с кодом и лишь 10% с используемой периферией.

Достаточно просто вспомнить, где он используется: микроконтроллеры, IoT, роботы, десктопные и мобильные приложения, веб, игры, системы моделирования, прогнозирования, обработки статистики и в нейронных сетях. Везде. Просто не существует такой области программирования, где C++ был бы бесполезен.

Да, сегодня сложно найти приложение, которое бы «от» и «до» было бы создано на C++. Но частично его использует почти любой крупный проект, поэтому его знание всегда будет являться плюсом в резюме программиста.

**Он активно поддерживается**

Речь не только об обновлениях стандартов и постоянном внедрении улучшений. C++ имеют колоссальное сообщество программистов, которые постоянно делятся библиотеками, шаблонами и кодами, приходят на помощь новичкам и опытным коллегам. При этом, в силу того, что C++ достаточно сложный язык, оно не изобилует лентяями и людьми, которые могут по 10 раз наступать на одни и те же грабли, подталкивая туда же новичков.

В первую же очередь, огромное сообщество — плодородная почва для появления качественной литературы.

По C++ есть несколько фундаментальных книг, по которым училось несколько поколений, есть новые, учитывающие все свежие изменения и актуальное ПО, есть масса интернет-ресурсов для обучения, в конце концов есть упомянутые стандарты.

**Он полезен в качестве фундамента для обучения**

Java, JavaScript, C #, как и огромное количество других популярных языков программирования, содержит в основе принципы C++. Именно поэтому учебные программы многих вузов содержат курс «плюсплюс», который идёт сразу за получением основ на C.

К примеру, принцип работы Java, одного из мощнейших языков современности, достаточно сложно понять, если не начать обучение с основ, которые впервые появились именно в C++.

Кроме того, популярно мнение, что если вы сможете изучить C++, то любой другой язык не вызовет у вас никаких затруднений. Это что-то вроде того, как легко пытаться пересесть с механической коробки передач на автоматическую и как сложно проделать путь в обратном направлении.

Почти все популярные языки программирования, что были придуманы позднее, имеют упрощенные структуру и механизмы, а процесс изучения сводится к освоению синтаксиса и ПО. Это создаёт целую армию программистов, которые просто не понимают, что создают в половине кода.

**Он востребован**

Всё вышеописанное является причиной тому, что программисты C++ востребованы по всему миру, что не в последнюю очередь сказывается на зарплатах.

Так, по России составляет 75 тысяч рублей. Более того, в отличие от многих популярных языков, офисная работа с которыми предполагает переезд в крупный город, востребованность программистами C++ имеет распределённый характер.

Перейдя по ссылке выше, вы узнаете, что максимальная зарплата предлагается не в Москве, а в Самарской области (средняя — свыше 100 тысяч рублей), в топе — Калужская и Липецкая.

Ну и самое главное, почему C++ — настоящий must-know для амбициозного программиста: работая с ним, ты ощущаешь, что куда больше создаёшь, чем используешь.

Достаточно сложный синтаксис, тысячи строк кода даже в простых программах, механизмы работы с памятью и периферией — поначалу всё это жутко бесит из-за постоянных ошибок. Но потом приходит дисциплина, опыт и ты понимаешь, что C++ — это и есть настоящее программирование.

**НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНИНИЯ**

### ****1. Операционные системы****

Язык C первоначально создавался для системного программирования, поэтому не удивительно, что его так активно применяют при создании операционных систем и программного обеспечения.

Быстрый и энергоэффективный, он достаточно близок к аппаратному слою, что позволяет работать с кодом низкого уровня.

Вот почему он идеален для разработки операционных систем.

Любая операционная система в первую очередь должна быстро запускаться и эффективно управлять системными ресурсами. Например, ****Linux, Microsoft Windows, Mac OS****полностью написаны на языке C, а Android и iOS — частично.

Разработчик на языке C может реализовывать все структуры данных самостоятельно, производя тонкую настройку мельчайших деталей операционной системы. Такая высокая гибкость — весомый аргумент в пользу языка.

В отличие от других языков программирования, язык C не имеет зависимостей времени исполнения. Это означает, что программа может выполняться силами аппаратных средств без участия менеджера памяти, входного или выходного слоя.

### ****2. Программирование встроенных систем****

Язык программирования C просто необходим при проектировании встроенных систем.

****Главным образом из-за высокой производительности и простоты использования****. Язык C очень экономичен в потреблении ресурсов системы, благодаря чему программа выполняется быстро. В результате встроенные системы работают в реальном времени без замедления.

****Ещё важнее здесь — код на C, будучи платформенно-независимым, не связан с каким-либо конкретным микропроцессором или системой.**** Так что встроенную программу можно скомпилировать в разных устройствах и платформах с незначительными изменениями или вовсе без них.

Вот почему беспилотные автомобили, умные часы, сенсоры и устройства, использующие технологию Интернета вещей (например, кофемашины), имеют встроенное ПО, целиком написанное на C или C++. В связанных системах C позволяет работать непосредственно с ресурсами памяти и править любую часть кода.

### ****3. Разработка игр и игрового движка****

Производительность крайне важна для игры, поэтому выбор языка программирования здесь ограничен.

****Являясь самым быстрым на сегодняшний день языком программирования, С++ оказывается одним из лучших для игр в 3D, многопользовательских и других.****

Например, Counter-Strike, StarCraft: Brood War, Diablo I, World of Warcraft — все эти игры написаны на C++. Не говоря уже о консолях Xbox и PlayStation, в основе которых лежит программирование C++.

В ядре игрового движка Unity — самого популярного движка для создания видеоигр под несколько операционных систем одновременно — также использовался C++.

Средства разработки C++ могут совладать даже с самой сложной игровой графикой. Они позволяют оптимизировать и регулировать то, как именно будут использоваться ресурсы памяти и структуры данных в игре.

### ****4. Разработка настольных и кроссплатформенных приложений****

C++ также можно использовать для создания настольных приложений. Всё благодаря превосходным кроссплатформенным средствам разработки (иногда называемым фреймворками), таким как Qt.

Оно позволяет нацелиться на Windows, Linux, macOS, Android и встроенные системы — все c единой кодовой базой.

Стоит упомянуть и о библиотеке SDL, нашпигованной функциями, позволяющими создавать приложения одновременно для Windows, Linux, Android, MacOS и iOS.

Кстати, Photoshop, Illustrator и Adobe Premiere целиком написаны на C++.

**ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ.СТАНДАРТЫ**

**Технические характеристики**

#### **Мультипарадигмальный**

На C++ можно писать программы в рамках нескольких парадигм программирования:

* процедурное программирование:
  + код “в стиле C”;
* объектно-ориентированное программирование:
  + классы;
  + наследование;
  + виртуальные функции;
* обобщённое программирование:
  + шаблоны функций и классов;
* функциональное программирование:
  + функторы;
  + безымянные функции;
  + замыкания;
* генеративное программирование:
  + метапрограммирование на шаблонах.

#### **Эффективный**

Одна из фундаментальных идей языков C и C++ — отсутствие неявных накладных расходов, которые присутствуют в других более высокоуровневых языках программирования.

* Программист сам выбирает уровень абстракции, на котором писать каждую отдельную часть программы.
* Можно реализовывать критические по производительности участки программы максимально эффективно.
* Эффективность делает C++ основным языком для разработки приложений с компьютерной графикой (к примеру, игры).

#### **Низкоуровневый**

Язык C++, как и C, позволяет работать напрямую с ресурсами компьютера.

* Позволяет писать низкоуровневые системные приложения (например, драйверы операционной системы).
* Неаккуратное обращение с системными ресурсами может привести к падению программы.

В C++ отсутствует автоматическое управление памятью.

* Позволяет программисту получить полный контроль над программой.
* Необходимость заботиться об освобождении памяти.

#### **Компилируемый**

C++ является компилируемым языком программирования. Для того, чтобы запустить программу на C++, её нужно сначала скомпилировать. Компиляция — преобразование текста программы на языке программирования в машинный код.

* Нет накладных расходов при исполнении программы.
* При компиляции можно отловить некоторые ошибки.
* Требуется компилировать для каждой платформы отдельно.

#### **Статическая типизация**

C++ является статически типизированным языком.

1. Каждая сущность в программе (переменная, функция и пр.) имеет свой тип,
2. и этот тип определяется на момент компиляции.

Это нужно для того, чтобы

1. вычислить размер памяти, который будет занимать каждая переменная в программе,
2. определить, какая функция будет вызываться в каждом конкретном месте.

Всё это определяется на момент компиляции и “зашивается” в скомпилированную программу. В машинном коде никаких типов уже нет — там идёт работа с последовательностями байт

**Стандарты**

## **Auto-тип переменной**

То есть описание типа может быть очень длинной строкой. Вместо этого в C++11 можно объявить имя типа, как auto, что означает, что компилятор должен сам определить тип переменной. Обратите внимание, это НЕ динамическая типизация! Переменная по-прежнему имеет строго определенный тип, который не может быть впоследствии изменен! Но этот тип программист просто не указывает явно для облегчения труда.

В C++11 появились range-based циклы: циклы, в которых переменная пробегает по всем значением контейнера. Контейнер должен поддерживать методы begin() и end() -- это может быть vector, list, set, map. Если цикл пробегает по элементам контейнера map, то переменная будет парой из двух элементов: ключ и значение.

## **Универсальная инициализация**

В С++11 такая универсальная инициализация распространена на структуры и классы. В списке инициализации в фигурных скобках необходимо указать значения полей структуры в том порядке, в котором они объявлены. Например, В этом случае P.x будет равно 1, P.y будет 2. Поскольку pair является структурой, то этот способ можно использовать для инициализации pair.

## **Шаблоны с переменным числом аргументов и std::tuple**

В C++11 появилась возможность создания шаблонов с переменным числом аргументов. Один из примеров такого шаблона - tuple, определенный в одноименном заголовочном файле. Это кортеж, то есть переменная, которая содержит несколько полей различных типов.

## **Структура tie**

tie является структурой, определенной в заголовочном файле tuple. Структура tie похожа на tuple, только членами структуры являются ссылки на какие-либо переменные (или прочие lvalue, например, элементы массива и т.д.). Это позволяет модифицировать переменные, переданные в качестве параметров при создании tie.

## **Лямбда-функции**

Лямбда-функции - это безымянные функции, которые не получают собственного имени, а используются только в месте объявления. Распространенное использование лямбда-функций - это параметр-компаратор при сортировке.

## **Угловые скобки во вложенных шаблонах**

в C++11 можно не ставить пробел между "<<", что было обязательным в ранних версиях C++.

## **Типы данных**

С С++11 стандартизирован тип long long int, который раньше был расширением GNU C++. Кроме того, появились типы данных фиксированного размера, например, "32-битное целое число", они определены в заголовочном файле.

## **Стандартная библиотека**

Много изменений произошло в стандартной библиотеке. Например, появились хеш-таблицы unordered\_set и unordered\_map. Появились усовершенствованные генераторы случайных чисел и поддержка регулярных выражений.

**РАЗРАБОТКА САЙТА**

**Алгоритм создания**

* Выбор темы

Отталкиваясь от темы мы начинаем подбор информации для заполнения сайта. Тема дана нам изначально. Тема сайта - С++.

* Подборка материала для сайта (контент)

Получая информацию из разнообразных источников, подчеркиваем дизайн и некоторые фишки оформления для следующего шага. Материал подбирается исключительно под тему сайта, то есть видео, табличные данные, логотип сайта, вставки с картинками и текст.

* Определение дизайна сайта

Определяем цвет, формы, таблицы, тематика вставок (внешний источник).

На этом этапе проводится анализ других сайтов с выявлением недочетов и недостатков тех сайтов.

* Замена прямых образов символическими

Вставка смайлов, картинок, таблиц и видео.

* Заполнить сайт информацией

Наполнение информацией, подходящей и имеющей почву для размышлений для непосредственного пользователя интернета.

Наполнение также визуальными данными, которые ослабят нагрузку на глаз пользователя.

* Протестировать сайт на работоспособность

Проверка гиперссылок, связи, текста(непосредственно пользователем).

**СРЕДСТВА РАЗРАБОТКИ**

Чтобы получить готовый сайт, необходимо проработать два направления – ввод информации и вывод ее в браузере. Ввод может осуществляться также через браузер либо каким-либо другим методом. Так, нередко страницы кода создают в Блокноте или специализированных программах для создания сайтов, а потом выкладывают их в виде файлов на сервер.

Но, как бы там ни было, первое, что нужно организовать, это ввод данных. Причем, в таком виде, чтобы впоследствии браузер смог их вывести на экран. Описанные выше технологии как раз и отвечают за ввод данных для сайта.

В разработке сайта использовались HTML и CSS языки. С помощью них написана основная часть сайта.

Стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора.

Формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Также может применяться к любым XML-документам, например, к SVG или XUL.

В работе я использовал flex, а также flexbox. Их мы можем проследить почти на каждой странице сайта. Grid показался мне менее практичным и я решил исключить его из сайта. станавливает основную ось, таким образом определяя направление flex элементов, помещаемых в flex контейнер. Flexbox — это (помимо дополнительной упаковки) концепция однонаправленного макета. Думайте о flex элементах, как о первичных раскладках в горизонтальных рядах или вертикальных столбцах.

При создании сайта, я обращался к интернет ресурсу. Видео и текстовые материалы, а также специализированные сайты по написанию сайтов.

**ОПИСАНИЕ САЙТА**

Данный сайт рассказывает вам о С++ и научит вас азам программирования на языке С++. Вы можете отследить данные материал на нескольких страницах этого сайта. Он четко распределяет в каком направлении вам стоит двигаться.

Данный сайт включает в себя интернет источники из вне, которые также помогут вам в изучении материала, видео-материал который обозначит вам все точки над и.

В данном сайте есть 7 основных страниц с информацией.

Они подразделяются на следующие темы:

* Главная
* Первая программа
* Переменные
* Типы данных
* Ввод данных
* Циклы
* Массивы

Главная старница вводит вас общую картину и концепцию данного сайта.

Первая программа - здесь описывается первая программа, то есть то с чего вам стоит начать при изучении языка С++. Она помодет вам понять, что из себя представляет С++ и разобраться непосредственно на примере вам поможет как раз данная страница моего сайта

Переменные - это очень важная тема, без которой вам не обойтись, после того как вы посмотрите первую программу, у вас и появтся вопросы. Одним из них будет, что же такое переменные и тут вам поможет эта страница сайта. Здесь все расписано и есть внешний источник на котоый вы можете перейти для более углубленного изучения данного материала.

Типы данных - это будет вашим вторым вопросом, эта страница дополнена таблицей с данными. По которой вам будет гораздо легче понять что из себя представляют эти самые типы данных.

Ввод данных, это текстовая страница сайта, которая не имеет каких-то вставок, она предназначена лишь для чтения с последующим переходом на внешний источник, который вам поможет разобрать эту тему по полочкам.

Циклы- эта страница также является текстовой и предназначена для прочтения. Она также оснащенна внешним источником с информацией.

Массивы - это странцица включает в себя помимо внешней информации также видео материал, который помодет вам понять тему более глубже. В данном видео уроке будет рассказывааться непосрественно о массивах. Урок этот будет вести проффесионал своего дела, который знает с чего лучше начать урок. Также вы можете перейдя по ссылке посмотреть еще больше видеоуроков которые он преподает.Ввод данных, это текстовая страница сайта, которая не имеет каких-то вставок, она предназначена лишь для чтения с последующим переходом на внешний источник, который вам поможет разобрать эту тему по полочкам.

Также внутри сайта установлена ссылка с иконкой, то есть она представляет собой главную иконку сайта, и если вы на нее нажмете левой кнопкой мыши то вы перейдете на главную страницу сайта, порой это многое упрощает.

Также я добавил бегущую полоску при наведении на раздел сайта, то есть название его страницы.

**ЗАКЛЮЧЕНИЕ**

В эпоху бурного развития информационных технологий, электронное представительство в сети Интернет необходимо любому человеку для изучения, расслабления и просто досуга в жизни.

В настоящее время сайт - необходимое средство коммуникации для любой компании, для любого учреждения, человека. Каждой компании - будь то новичок на рынке или огромная корпорация - необходимо доносить до своих клиентов информацию.

Постановка задачи данного дипломного проекта включала в себя следующие основные этапы:

- формирование направлений работы создания сайта;

- выбор инструментального средства для создания сайта;

- разработка структуры сайта;

- разработка стиля сайта;

- заполнение соответствующим материалом страниц;

- отладка работы.

Формирование направлений работы с сайтом включает в себя составление плана действий, обзор материала для сайта.

Выбор инструментального средства для сайта включает в себя обзор программных средств для реализации сайта.

Данный сайт предназначен для изучения С++ новичкам.

С++ будет актуален очень многие годы, и пока мы молодое поколение научимся азам его магии, появитсяя множество новых фишек. С++ не стоит на месте, там есть усовершенствования и о также будет расширяться. Мой сайт предназначен для изучения всех тонкостей этого языка программирования.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Расселл, Д.Ф. Структура С++ [Текст] / Д.Ф. Расселл, И.Б. Байсун, Е.А. Кохн. – М.: Физматлит, 2014. – 98 с.
2. Стелтинг, С.А. С++ без сбоев [Текст] / А.А. Букейтов, С.А. Гуда. – М.: Прогресс, 2019. – 486 с.
3. Адамчик, Л.Н. Прикладная иформатика[Текст] / Л.Н. Демидов. – М.: Прометей, 2010. – 398 с.
4. Герберт, Ш. С++ базовый курс [Текст] / Г. Шилдт, – М.: Форум, 2020. – 190 с.
5. Эккель Б. А. Философия С++ [Текст] / Г.А. Лисьев, Ю.И. Эккель, Б. А., Аскерко, П.Ю. Романов. – М.: Инфра-М, 2019. – 145 с.
6. Власова, В.Г. Объектно-ориентированное программирование [Текст] / В.Г. Олифер, Н.А. Олифер. – СПб.: Питер, 2017. – 632 с.
7. Сергеев, А.Н. Microsoft.NET [Текст] / А.Н. Сергеев. – СПб.: Лань, 2015. – 184 с.
8. Берель Б. В. Перемены в С++ [Текст] / Г.А. Бакоев, Ю.И. Берель, Б. В., Касопов, В.Ю. Росомахин. – М.: Инфра-М, 2009. – 130 с.