МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«Санкт-Петербургский государственный университет аэрокосмического приборостроения»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

|  |  |
| --- | --- |
| на тему | Разработка web-сайта со специальными вычислительными возможностями |
|  | |
|  | |

|  |  |
| --- | --- |
| выполнена | Просянкиным Денисом Николаевичем |
| фамилия, имя, отчество студента в творительном падеже | |

|  |  |  |  |
| --- | --- | --- | --- |
| по направлению подготовки/ специальности | 38.03.05 |  | Бизнес-информатика |
|  | код |  | наименование направления подготовки/ специальности |
|  | | | |
| наименование направления подготовки/ специальности | | | |
| направленности | Бизнес-информатика | | |
|  | наименование направленности | | |
|  | | | |
| наименование направленности | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент группы № | 8721К |  |  |  | Д.Н.Просянкин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2021

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«Санкт-Петербургский государственный университет аэрокосмического приборостроения»

ЗАДАНИЕ НА ВЫПОЛНЕНИЕ   
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ БАКАЛАВРА

|  |  |  |  |
| --- | --- | --- | --- |
| студенту группы | 8721К |  | Просянкин Денис Николаевич |
|  | номер |  | фамилия, имя, отчество |

|  |  |
| --- | --- |
| на тему | Разработка web-сайта со специальными вычислительными возможностями |
|  | |
|  | |

|  |  |  |  |
| --- | --- | --- | --- |
| утвержденную приказом ГУАП от |  | № |  |

|  |  |
| --- | --- |
| Цель работы: | Заключается в развитии методов информатизации и широком |
| использовании сайтов интернета для решения прикладных задач бизнес-информатики | |
|  | |

|  |  |
| --- | --- |
| Задачи, подлежащие решению: |  |
|  | |
|  | |

|  |  |
| --- | --- |
| Содержание работы (основные разделы): |  |
|  | |
|  | |
|  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Срок сдачи работы « |  | » |  | 20 | 21 |

Задание принял(а) к исполнению

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| студент группы № | 8721К |  |  |  | Д.Н.Просянкин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Оглавление

[Введение 6](#_Toc68209556)

[1. Анализ предметной области 7](#_Toc68209557)

[1.1. Роль интернета в жизни человека 7](#_Toc68209558)

[1.2. Прирост пользователей интернета и web-сервисов 7](#_Toc68209559)

[1.3. Вычислительная сторона сайта 8](#_Toc68209560)

[1.4. Online версии desctop приложений и тенденции к этому 8](#_Toc68209561)

[1.5. PWA сайты как набирающий тренд решение 9](#_Toc68209562)

[1.6. ASP.NET 9](#_Toc68209563)

[1.7. ASP.NET – компонентная версия ASP 10](#_Toc68209564)

[1.8. Обзор альтернатив для символьной математики 14](#_Toc68209565)

[1.9. Photomath (фото режим – поддержка документов MS) 14](#_Toc68209566)

[1.10. Geometryx (Визуально понятные вычисления) 15](#_Toc68209567)

[1.11. Моя система 16](#_Toc68209568)

[1.12. Цикличные вложенные вычисления 16](#_Toc68209569)

[1.13. Работа с файлами 17](#_Toc68209570)

[1.14. Поддержка больших чисел 18](#_Toc68209571)

[1.15. Выводы: 19](#_Toc68209572)

[2. Разработка алгоритмов 19](#_Toc68209573)

[Заключение 33](#_Toc68209574)

[Тезариус 34](#_Toc68209575)

[Список литературы: 35](#_Toc68209576)

Реферат

Актуальность темы.

Цель и задачи работы, которые определяются исходя из темы ВКР.

Объект и предмет исследования или разработки.

# Введение

1. Анализ предметной области
2. Роль интернета в жизни человека

Первый в мире сайт создал Тим Бернерс-Ли в 1990 году. На нём было описание новой технологии World Wide Web, основанной на протоколе передачи данных HTTP, системе адресации URI и языке гипертекстовой разметки HTML.

С тех пор сайты проникли во все сферы человеческой деятельности. В каждой области сайт может выполнять совсем разные функции. Так же не стоит забывать о средствах, которые готовы выделять компании на разработку и поддержание web-страниц. Именно поэтому существует так много типов сайтов и с каждым годом классификация становится всё сложнее.

Один из таких типов – это web-сервисы. Такой сайт может содержать в себе online инструмент(ы) для решения конкретной задачи. Уже не мало лет такие сервисы пользуются спросом и их применение с каждым годом набирает всё большие обороты. Приведём несколько примеров, например, это – сайты отслеживающие текущую скорость соединения с интернетом, показывающие ip устройства с которого был выполнен в ход на страницу, конвертеры файлов в различные форматы и многое другое.

1. **Прирост пользователей интернета и web-сервисов**

Каждый год We Are Social и Hootsuite представляют отчет о состоянии интернет-среды – Digital 2020. В указанных данных имеется информация о 298 миллионов новых интернет-пользователей в сравнении с данными на январь 2019. Теперь, общее число пользователей оценивается в 4,54 млрд, это +7% за год.

Также, с каждым годом, число людей, пользующихся мобильными телефонами, растет, что делает интернет еще более доступным для чуть ли не ежечасного посещения. По представленной статистике в среднем по всем данным пользователь проводит в интернете 6 часов 43 минуты каждый день, включая соц.сети и прочие ресурсы. Эта цифра разница от страны к стране.

Сайты создают каждый день. Рубеж в 1 млрд сайтов был пройден в 2016 году, а уже в 2017 году общее число веб-сайтов выросло до 1,7 млрд. на 2019 год их количество практически не изменилось – общее число сайтов в мире на тот момент составляло чуть более 1,7 млрд.

1. Вычислительная сторона сайта

Вычисления производятся на каждом сайте. Для простого сайта-визитки это могут быть запросы к серверу за корректным откликом на событие через нажатие кнопки или ссылки. Сервер должен произвести простые вычисления, например, выдать нужный каталог в зависимости от номера нажатой страницы. Бывают и более сложные вычисления, например, одновременная работа с базой данных нескольких пользователей.

Есть вычисления другого типа. Например, сайты визуализации 3д моделей, решения систем уравнений или интегралов. Всё это – вычисления, скрытые от пользователя, происходящие на сервере. Такие вычисления - это часть back-end разработки сайта. Одни из языков back-end разработки являются PHP и С#.

1. Online версии desctop приложений и тенденции к этому

Эволюция приложений не стоит на месте и с нарастанием спроса на web-сервисы, решающие задачи desktop приложений, некоторые большие приложения начали адаптировать под браузерный доступ. Так, не для кого не секрет, что уже не первый год отлично работает Microsoft office online, куда входят такие приложения как: word, excel, power point и др. Данные программы доступны с некоторыми ограничениями, но каждому кто имеет аккаунт Microsoft. Уже не обязательно устанавливать офис себе на компьютер или ноутбук если вам хватает функционала web-версии. Это одно из преимуществ web-сервисов – не нужно ничего устанавливать, требуется только браузер. От сюда вытекает и второй плюс – программы совсем не занимают места, браузер по объёму меньше тех программ, к которым может дать доступ. Matlab, некоторые продукты Adobe, такие как Photoshop Express, тоже доступны в онлайн версии.

Но у web-сервисов есть один большой недостаток – online подключение. Интернет доступен еще не во всех уголках планеты, а где он есть, не всегда стабильное подключение и с недостаточная скорость передачи данных. Со всеми удобствами и преимуществами web-сервисов, такой формат всё еще не каждый может себе позволить.

1. PWA сайты как набирающий тренд решение

Одним из решений таких проблем стало относительно новое направление PWA. Для Progressive Web Apps (PWA) - нет чёткого определения. Можно сказать, это нечто среднее между сайтом и приложением в том виде, в которых мы знакомы с каждым из них. Вот три главных принципа на которых строится PWA:

Reliable – независимость от соединения с сетью интернет и статуса приложения.

Fast – быстрый взаимообмен данными, отзывчивый интерфейс.

Engaging – комфортная, удобная и приятная работа с приложением.

PWA и внешне и внутренне похож на обычное приложение. Разница лишь в том, что PWA обращается к своим ресурсам по HTTP протоколу, но хранящимся локально. Это одинаково работает на всех основных браузерах и на всех основных платформах. Стоит сказать об ещё одном отличии. Оно заключается в правах доступа к ресурсам системы, но этот вопрос можно решить, возможно, вскоре у PWA появятся новые привилегии.

1. ASP.NET

ASP.NET является частью технологии .NET Framework, принадлежащей компании Microsoft. Последняя технология является платформой не только для создания Web-сервисов и ПО, но и для их непосредственного запуска. .NET Framework это высокопроизводительная среда, основанная на стандартах и поддерживающая множество языков. Она поддерживает интеграции приложений с сервисами следующего поколения и решает задачи web-приложений, основные из которых использование и развёртывание. Данная система состоит из нескольких частей.

1. Сommon Language Runtime – общеязыковая среда выполнения.
2. Различные библиотеки классов, построенные иерархическим образом.
3. ASP.NET – компонентная версия ASP

ASP.NET используется для создания интернет приложений и отличается мощностью клиент-серверной части. Другой особенностью является создание динамических HTML страниц, что добавляет удобство и простор для разработчика. В результате объединения технологий ASP, используемых ранее и .NET Framework технологий возникла эта новая система, которая содержит разработанные и готовые к использованию элементы управления, которые позволяют достаточно быстро и удобно создавать web-сервисы. Не редко используются возможности, предоставляемые сайтами других разработчиков, для пользователей разрабатываемого сайта. ASP.NET почти не имеет границ применения в наше время, разработчик ограничен лишь собственным воображением.

Раз были упомянуты динамически создаваемые HTML коды, то стоит разобраться в чем их отличие от статических и какие преимущества несут данные возможности. Статическая HTML страница также содержит код на языке разметки HTML. В данном случае, при написании, определяется внешний вид web-страницы для всех пользователей, которые попадают на сайт. Исходя из названия, можно понять, что никакие изменения, без прямого редактирования файла HTML, происходить не будут. Это значит, что независимо от того, когда, кто и с какого устройства будет выполнен вход, страница будет всегда и для всех выглядеть одним и тем же образом. HTML документа достаточно для разметки вывода информации, которая почти не будет подвержена изменениям и не будет иметь зависимостей от устройств или других условий просмотра страницы. HTML страница – это текст из таблицы символов ASCII, который поддерживается на любой операционной системе.

Из выше сказанного следует, что, формируя страницу с помощью разметки HTML, мы добьемся совершенно статичного результата в плане содержимого. Внесенные данные при разработке – это вся информация, которая будет на странице. Внесенные пользователем в дальнейшем данные, никак не могут быть использованы для отображения нового содержимого, так как они не являются предварительно записанными в окончательный набор файлов сайта.

Совершенно ясно, что при написании статической web-страницы курсов валют, за невозможностью менять информацию сайта программно, программисту бы пришлось обновлять HTML код вручную, каждый раз, когда курс валют изменится. Понятно и то, что данный способ разработки сайта, в данном контексте, никто не использует. Динамические страницы решают проблему поддержания актуальности информации на сайте на требуемом уровне. Наполнение страницы динамическим образом – это способ обмена информацией с пользователем, учитывая интересы последнего. Такое двустороннее сообщение происходит от клиента серверу и обратно.

Веб-страницы, с динамической генерации HTML-кода, обычно проходят обработку на сервере перед отправкой клиенту. Самый тривиальный пример - это программа, которая может изменять статические страницы, запрашиваемые клиентом. Такая система использует хранилище данных и может опираться на параметры запроса. Даже такой простой пример решает ранее рассмотренную задачу. Программисту остается лишь подготовить одну html-страницу, которая будет являться шаблоном, и программно изменять её на сервере перед отправкой пользователю.

В самом начале развития интернета большая часть веб-страниц использовала статический метод написание HTML-кода. С каждым годом количество динамических страниц растёт. Это закономерность обусловлена тем, что каждый пользователь хочет видеть не только готовую информацию, но и непосредственно принимать участие в событиях, которые могут происходить каждый день. События могут являться как конкурсами и акциями, так и обновлением товаров в интернет-магазине, актуальными новостями, новыми сообщениями в блоге. Многие сайты обновляют информацию чаще чем раз в минуту. Динамическая генерации позволяет таргетировать информацию для каждого конкретного пользователя, а также быстро откликаться на его запросы через браузер. Для каждого из этих пунктов были разработаны специальные функции. Одной из таких функций является файлы cookies, которые способны сохранять пользовательские данные о настройке сайта или о личности потребителя.

Для такого рода процедур существуют языки программирования с функционалом для динамического изменение веб-сайтов. Во-первых, это скрипты на клиентской части системы. Данные скрипты можно писать на языке JS или VBscript. Особенность этих скриптов в том, что они встроены в HTML-код, который браузер получает от сервера. Такие сценарии, именно так принято называть скрипты, обрамляются HTML тегами. Благодаря этому, браузер может интерпретировать данную часть кода как скрипт и вывести результат пользователю. Такие программы не большие. Если мы отправляем запрос к базе данных, то отображаемая пользователю информация не должна превышать разумных пределов. Скрипты способны проверять корректность запроса пользователя для того, чтобы серверная часть сайта не обрабатывала запросы с ошибками. Существуют также скрипты, создающие эффекты анимации, но это другая тема. Возможности клиентской части скриптов не безграничны. Например, для отправки SMS-сообщений скрипта на JavaScript будет недостаточно. Если страница использует анимацию JavaScript, то она все равно считается статической.

Главная особенность динамический страницы в том, что она создается «на лету» или на английском RunTime, то есть во время выполнения. Изначально сервер присылают пользователю страницу статического формата. В ответ на запросы от пользователя, сервер, который хранит исполняемые файлы-программы, которые могут быть написаны на С#, С++, Delphi или PHP, посылает пользователю HTML-код, который создал специально под обрабатываемую ситуацию.

Таким ответом может быть результат поиска в браузере, переход на другой веб-сайт, сообщение об успешной отправке письма другому человеку или успешная авторизация. Сервер принимает на вход данные, которые ввёл пользователь. Результат обработки — это обновлённая web-страница. Cgi- программа подразумевает выделение отдельного процесса для каждого запроса пользователя. Это означает, что если пользователи в один момент времени сделают запрос к серверу, то в итоге в памяти будет 100 процессов. Масштабируемость такого сайта очень низкая, так как нет плавного изменения времени ответа с увеличением числа одновременных запросов.

Старая технология ASP была очень ограничена. В ней использовались языки скриптов, возможности которых нельзя было сравнить даже с самыми простыми языками программирования. Html-код в виде специальных тегов внедрялся в код asp, что создавало большую путаницу, так как asp код был разбросан по всему html-документу. Проблема ещё заключалась в том, что html-код, по большей части, создают дизайнеры, а asp код - программисты. ASP.NET решает эту проблему возможностью разделить код в разные файлы.

ASP.NET не является продолжением старой технологии ASP. Это система концептуально переработана корпорацией Microsoft, выполненное по правилам технологии .NET. ASP.NET содержит в себе максимум для более простой поддержки web-приложения. Эта технология основана на ООП, и при этом сохранила шаблоны ASP. Где раньше исполняемая программа помещалась в директорию для работы на сервере. Появилось множество новых технологий, а всё старое было переосмыслено и доработано.

Например, в asp.net теперь используется языки, которые требует компиляции до запуска. Это означает, что теперь синтаксис кода будет проверяться программой, в которой работает разработчик. Также скомпилированный код переводится в промежуточный язык. Код на данном языке выполняется быстрее и не зависит от того языка, на котором мы разрабатываем. Также стоит отметить, что он поддерживает строго типизированные языки. Это понижает риск ошибок конвертации типов, улучшая читаемость кода и имеет другие преимущества.

Компиляция кода происходит каждый раз автоматически, когда код был изменён и произошло обращение к серверу. Веб-страницы базируется на asp.net располагается платформа .NET Framework, которая взаимодействует с ОС.

В 2004 году появилась вторая версия asp.net. С приходом обновления объём кодирования сократился на 70%. На сегодняшний день последней версии является пятая. С годами разработчики поддерживали главные цели системы - высокая скорость разработки, администрирование, повышение скорости работы сервера, масштабируемости, лёгкой поддержке сайтов разделение функциональных частей кода.

ASP.NET — это технология, поддерживающая разные языки программирования, такие как С#, Visual Basic, Но именно язык С# был специально создан для данной платформы. Именно этот язык позволяет применить максимум концепций, методов и паттернов объектно-ориентированной разработки. Для начала работы с ASP.NET следует понимать основы создание сайтов на HTML CSS, а также уметь писать на Java скрипт.

1. Обзор альтернатив для символьной математики

Математические вычисления окружают любую профессию. Именно поэтому программы, облегчающие данный процесс, пользуются большой популярностью. Наиболее популярная система используется, конечно, по большей части учащимися школ и вузов. Но, понятно, что несмотря на популярность данных сервисов, это лишь малая часть вычислений, которое происходит в мире каждый день.

Посмотрим на плюсы и минусы как простых программ, рассчитанных больше на обучающихся по тем или иным профессиям или школ, так и профессиональные средства, используемые в мире крупными корпорациями и миллионами профессионалов.

1. Photomath (фото режим – поддержка документов MS)

Это популярная среди школьников и студентов программа. Она существует на всех популярных платформах, как на десктопе, так и для смартфонов. Разработчики позиционируют своё детище, как нечто большее, чем калькулятор. Данная система помогает производить вычисления по формулам, внесенным тем или иным образом.

Главной особенностью приложения такого рода является распознавание печатного текста. Это удобно тем, что пользователю не нужно вручную вводить ту или иную формулу. Достаточно навести камеру на смартфоне или загрузить фотографию на компьютере, и программа тут же расшифровывает запрос и выдаст ответ. Распознавание текста работает в 9 из 10 случаев. На случай ошибки предусмотрена возможность редактирования формулы. Распознавание текста с вычислением ответа происходит мгновенно.

Конечно, существует возможность ввести формулу вручную с помощью удобного интерфейса создания формул. Пока что данная программа не умеет распознавать рукописный текст, что не является большой проблемой, так как как всё приложение рассчитано на школьника или студента, задание для которого было выдано в печатном виде.

Программа может рассчитывать логарифмы, частично занимается интегралами и позволяет решать уравнения с двумя неизвестными. Это неплохой помощник для учащихся и иногда для их родителей, но даже на высших курсах математических вузов, программа уже не сможет справляться вычислениями, поставленными перед старшекурсниками.

1. Geometryx (Визуально понятные вычисления)

Данное приложение целиком и полностью направлено на геометрические расчёты. Оно позволяет быстро и визуально понятно произвести расчёт, связанный с основными геометрическими фигурами.

Такое приложение способно, например, вычислить объём, периметр, высоту, длину стороны, диагонали, меры углов, и многие другие геометрические задачи. Также, как и предыдущие приложения, оно обладает моментальной вычислительной способностью. Также может подсказать ввод данных, которые следует добавить, если вычисление не может быть произведено по той или иной теореме.

Данное приложение содержит изображение вычисляемых фигур с подробным обозначением каждой части объекта. Помогает визуально понять учащемуся, какие именно составляющие были введены, где они располагаются на фигуре, и какая часть объекта была вычислена.

Несмотря на немалый спектр возможностей, данное приложение нельзя назвать профессиональным. Оно, как и предыдущий вариант, является скорее помощником для решения учебных задач. Узкая направленность приложения не предусматривает не только темы за пределами геометрии, но даже в этих рамках вычисление происходят по добрым формулам, которые были отобраны для максимального покрытия геометрических задач, стоящих перед учеником. Данные формулы не поддерживают возложенных вычислений, нестандартных решений задач вне школьной или вузовской программы.

1. ru.numberempire.com
2. www.mathway.com
3. MATLAB.
4. Моя система

Вычислительных программ очень много. Каждый, из представленных систем, является представителем множества похожих друг на друга программ. Совершенно определенно, что сейчас нет смысла создавать вычислительное приложение, если оно не будет включать полезные, отличающие его от других, функциональные возможности. Такими возможностями, в моём случае, являются:

1. Цикличные вложенные вычисления

Функционал вложенных вычислений позволяет упростить формулу, посредством вынесения часто используемых, повторяющихся функций за пределы основной формулы. Такой ряд формул может складываться друг в друга в любом порядке, за исключением случаев неконтролируемый рекурсии. Это позволит создавать большие по объёму вычисления формулы, каждая из которых будет представлена в виде иерархично упорядоченного множества формул. Такой подход визуально лучше воспринимается человеком, так как каждая формула из этого множества является частью чего-то большего, и, как следствие, мала по объёму и более понятна к понимаю. Второй плюс такого подхода заключается в том, что каждое из вычислений, по вынесенной формуле, можно легко изменить в одном месте, тем самым, изменив логику расчёта в каждом месте вызова данных формул, которых может быть, как две, так и сто. И, в ином случае, пришлось бы менять сто участков в каждой формуле, что очень длительная по времени задача для человека.

Цикличное вычисление подразумевают использование формулы множество раз за одно обращение пользователя. Это значит, что пользователь может заготовить входные параметры для множества вычислений по одной формуле, и программа вычислит ответ по каждому из набора данных в пределах одного обращения пользователя к серверу. Такой подход позволяет обрабатывать большое количество данных по одной и той же формуле без необходимости каждый раз изменять изначальные данные. Длительность этих вычислений напрямую зависит от интеграций расчетов.

1. Работа с файлами

Так как я планирую браузерное приложение, рассчитанное по большей части на работу с компьютером. Я решил осуществить поддержку файлов Microsoft Office. В рассмотренном ранее приложении Photomath, использовалась система распознавания текста. Это очень удобно, поскольку не нужно вводить текст вручную, но такой способ, на мой взгляд, больше удобен для сканирования обучающих материалов. Другое большое удобство заключается в том, что в теории пользователь может отсканировать текст, который сможет написать на листе ручкой намного быстрее, чем пользуясь даже самым удобным интерфейсом программы. Но распознавание рукописного текста — это очень непростая задача, вероятность неправильной интерпретации обратно зависит от каллиграфических навыков пользователя. Из плюсов, в данном случае, стоит рассматривать лишь возможность быстро считать формулу, находящуюся на том или ином носителе информации. Вложенность формул, которую я хочу внедрить в систему, сложно реализовать через сканирование даже напечатанного текста. Так как в формуле будут присутствовать не только искомое неизвестное, но и посторонние обозначение функций от скольких-то переменных. Помимо этого, обозначение должно быть установлено в соответствии между названием и реализации, вынесенные в другое место формулы. Подробнее данная система будет рассмотрена во второй главе. Использование файлов Microsoft Office позволяет создавать, сохранять, передавать построенные формулы как от пользователя серверу, так и от пользователя к пользователю. Помимо удобного интерфейса, создания формул, присутствующих в каждой программе и входящих в Microsoft Office, данные документы могут хранить и другую полезную для пользователя информацию. Например, набор рассчитываемых параметров как для одного вычисления, так и для множества вычислений сразу. Помимо этого, можно создавать множество формул, которые не только могут быть вложены друг в друга, но и могут являться отдельными единицами, вычисляемыми в порядке очереди. Так как на сервер может быть загружено больше 1 файла одновременно, в совокупности это всё открывает большой спектр возможностей для вычисления в требуемых областях. Пользователь может создать один документ с множеством формул и описанием, примеры вычисления или реальные данные заменяемые каждой новой необходимостью. Или можно создать десятки документов, каждый из которых будет tut.by выполнять определенную функцию.

1. Поддержка больших чисел

Многие системы вычислений имеют критичные, в очень редких ситуациях, ограничения. Например, не каждая система сможет показать результат вычисления факториала от 279. Ограничение, в основном, вводится из-за высоких нагрузок на систему. А может такие вычисления понадобятся в лучшем случае одному человеку из 1000. В таких ситуациях можно обойтись примерными расчётами, рассчитав хотя бы порядок получаемого числа. Но существует направление, которое требует не только оперирование с такими большими числами, но и высокую точность вычислений в разумных временных пределах. Точность относится не только к большим числам 200 порядка и выше, но и к дробям, чья не целая часть может быть меньше 1 миллиардной. Даже простейшие операции, с такими числами, создают большую нагрузку на производительность, не могут вычисляться минуту, час и больше в зависимости от величины числа и его точности. Реализация данной системы будет рассмотрена подробнее во второй главе.

1. Выводы:
2. **Разработка алгоритмов**

Алгоритм функции FindeEquations() с помощью встроенных функций находит все формулы документа и передает объект каждой формулы в цикл. Каждая итерация цикла сверяет полученные данные с текстовыми объектами word. Если данные совпадают, то значит ранее была сохранена формула в текстовом виде. Такая формула не может быть распознана программой, формулу получится распознать только в виде математической формулы. Данная функция удалит формулу в текстовом виде и сохранит формулу в виде математическом.

Математическое выражение также является текстом, разница заключается в способе получения текста. Рассмотрим формулу,

Изначально, все объекты в файле word может получить как текст, но, если получить формулу в виде простого текста, в строке какие-то символы могут быть проигнорированы, а какие-то нераспознаваемыми. Получаемая строка будет не полная и даже эта неполная информация возможно не будет распознана. Если не производить очистку от технических стволов строка обычного текста, полученная от формулы будет выглядеть примерно так:

Согласитесь, даже половины исходных символов в данной формуле не наблюдается. Можно решить, что они зашифровались в другие знаки, но в формуле остались только технические символы, не несущие нужного нам смысла.

Решение такой проблемы - это сообщить программе границы, в которых получение текста идет в виде математической формулы. Теперь, полученная строка без обработки будет выглядеть так:

Как бы много данная формула не содержала незнакомых простому читателю символов, всё же, заметно, что она несет больше смысла чем предыдущая. И правда, по данной формуле, разбив её специальным образом, уже можно производить действия. Именно получение последнего из типов строки и занимается алгоритм на рисунке 1.

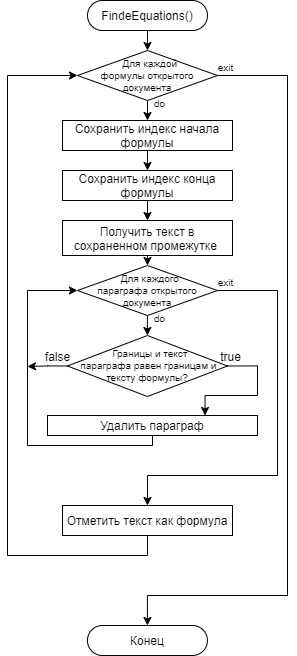
****

Рис. 1 - FindeEquations

Алгоритм функции Parse выполняет задачу установления связи между элементами. Основных элементов два:

* Параметры
* Функции

Параметры могут являться числом, набором чисел, числовой константой и другим числовым не вычисляемым значением со специальным зарезервированным именем.

Функцией могут быть параметры и операции между ними объединённые под одним именем через знак равенства.

Обязательные ограничения к имени параметров и функций:

Имя является последовательностью символов латинского алфавита без пробелов и математических операций.

Два и более имён не должны составлять одно из зарезервированных имен.

Рекомендационные ограничения:

Длина и посыл имени должна быть осмысленными. Слишком длинные имена мешают чтению формулы, а неправильный посыл может запутать читателя.

Вернемся к функции Parse, её можно разделить на три части. Первая часть находит имена функций, вторая имена параметров, третья последовательность выполнения формулы.

Поиск имени в функции происходит по шаблону. Как было сказано ранее, имя функции задается через знак равно. Каждую функцию в документе программа разделяет на части по признаку знака “=”. Если частей одна, значит знака равно не было, значит имя функции не задано. Если частей две, то проверяется сначала первая, если данная часть содержит только символы латинского алфавита, то она считается именем, если были встречены другие символы, то проверятся другая часть функции, по тому же принципу. Если в обеих частях имя не было найдено, то в отчет об ошибках помещается информация о нераспознанном имени. Пользователь в дальнейшем сможет вручную решить данную проблему указав, что имени и правда не было, и это всё единая формула равенство. Либо исправив неподходящее имя в документе.

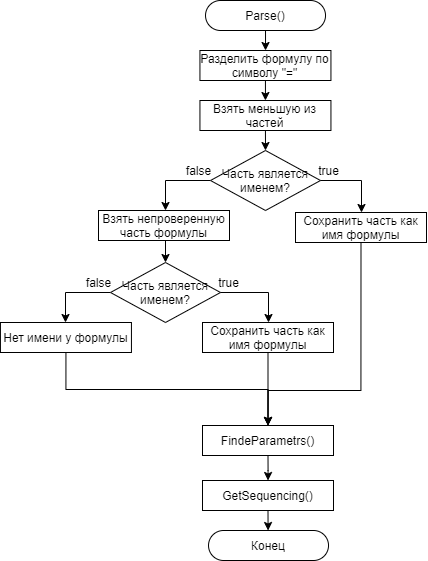


Рис. - Parse

Далее происходит вызов функции FindeParametrs. Алгоритм функции можно увидеть на рисунке 2. Для работы она требует вызов двух других функций, задачей которых является поиск параметров в двух областях: в формуле, где они использовались и в файле где записаны их значения.

Так как в формуле параметр или функция может использоваться многократно, то и найденные параметры могут содержать повторения. Также повторения могут быть и в файле, например, если значения параметра указаны в двух и более местах файла, а не только в одном. Такое может случиться при большом числе значений параметра.

Параметры объединены под единое имя, повторы удалены.

Далее, так как имена функций и параметров могут ничем не отличаться, функция сопоставляет найденные имена с именами формул и параметров и в случае неудачи сообщает пользователю что имя не было найдено. В случае успеха, имя будет иметь тип «параметр» или «формула». Таким образом установится взаимосвязь между вызовами и значениями, числовыми или вычисляемыми.

Стоит обратить внимание на второй тип ошибок, обрабатываемых в алгоритмах. Ошибка возникает если группа имён может образовать какое-то другое сохраненное имя. Например, если в файле были найдены три параметра a, b, и ab, то при обращении к параметру ab возникает неопределенность. В математических формулах принято опускать знак умножения, в такой ситуации параметр “ab” можно интерпретировать как «a\*b». Параметр a и параметр b у нас определен, не ясно к каким параметрам дет обращение, к «ab» или к «a\*b». В такой ситуации пользователю будет указано на места ошибок, и он должен будет исправить неоднозначности.

Такой проблемы можно было бы избежать, заставляя пользователя явно указывать умножение. Но в таком случае нарушается эстетическая составляющая, которая может отразиться на комфорте пользователя, привыкшего к формулам в стандартном виде. К тому же, чем больше условностей с работой программы, тем больше шанс что пользователь сделает что-то не так. У нас есть возможность обезопасить и упростить работу с программой, усложнив разработку, это необходимые меры, именно поэтому мы воспользуемся такой возможностью.

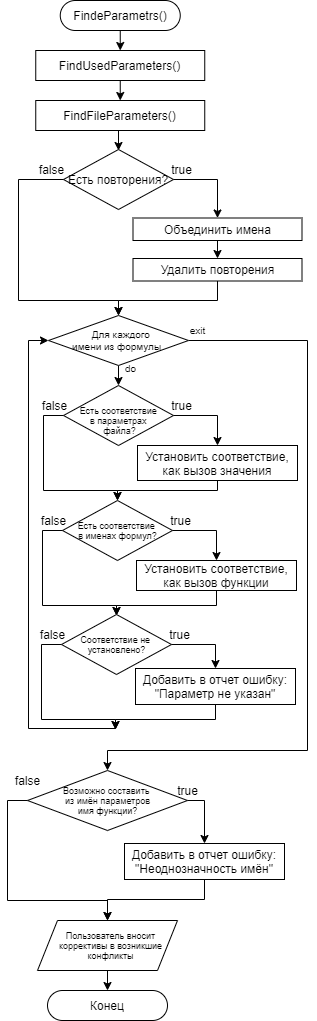
****

Рис. 3 - FindeParametrs

Функция FindUsedParametrs, алгоритм которой представлен на рисунке 3, занимается поисков всех вызываемых имен из функции. Составляет список вызываемых имен, которые в дальнейшем, по описанной выше технологии, вызываемая функция распределит по типам: параметр или функция.

Технология поиска следующая: Программа хранит список поддерживаемых операция и алфавит английского языка. При переборе каждого символа, если был встречен символ операции, то он пропускается, если встречен символ алфавита, то он добавляется одним из двух способов. Если предыдущий символ был также символом алфавита, это значит, что данный символ может является продолжением имени какого-то параметра, и должен быть записан ко всем предыдущим символам подряд. Если же до данного символа была какая-то операция, значит это определенно новое имя или новое начало имени, в любом случае сохранение идет как новые данные.

В конце функции пользователь может посмотреть на нераспознанные символы и разрешить данный тип конфликта.

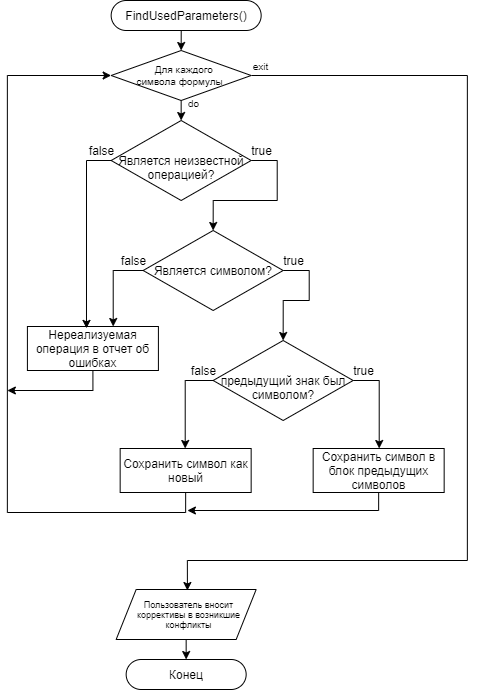
****

Рис. - FindUsedParametrs

Функция FindFileParametrs, алгоритм которой представлен на рисунке 4, занимается поиском параметров среди данных файла. В циклах происходит перечисление по каждой таблице, строке и ячейке и сохраняются не числовые значения, которые воспринимаются как имена параметров. В дальнейшем для удобства будут добавлены системы игнорирования текста, например, если пользователь решит прокомментировать параметр, программа не будет воспринимать комментарий как новое имя параметра.

Также сохраняются данные в каких ячейках находятся имена. Пользователь может расположить имена параметров таблицы в двух направлениях: по горизонтали или вертикали. У каждого направления есть две стороны - в итоге выходит 4 положения. Сверху, снизу, справа, слева. Для каждой таблицы вычисляется в каком месте были записаны имена, другие ячейки объявляются значениями параметра.

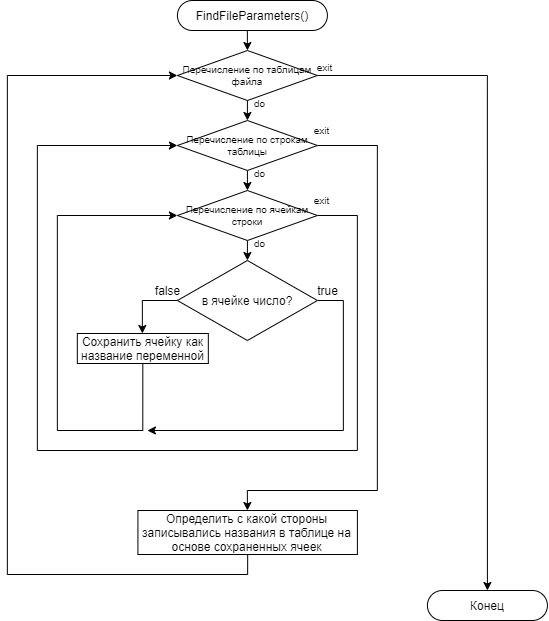


Рис. 5 - FindFileParametrs

Объединяющим все эти операции алгоритмом является ExecuteEquations, представленная на рисунке 5. Она может выполнять переданную строковую формулу. После вызова функции Parse идет перечисление по элементам пользовательского класса Math\_act. Последняя функция GetSequencing вызываемая в Parse это метод того самого класса Math\_act, который и формирует объекты своего класса. Алгоритм

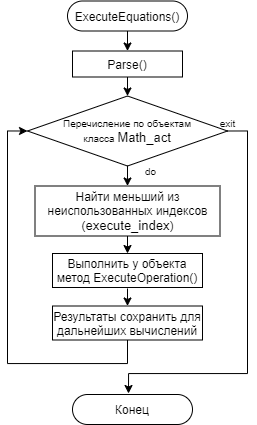
****

Рис. 6 - ExecuteEquations

Класс Math\_act содержит поля и методы.

Параметры:

1. ParamA

Хранит имя первого параметра для вычисления. Если операция унарная, то это единственный хранимый параметр. Данное имя может являться вызовом функции, тогда значение функции будет рассчитано и возращено после вызова имени.

1. ParamB

Хранит имя второго параметра для вычисления. Существует только если операция бинарная. Данное имя также может являться рассчитываемой функцией.

1. Action

Основной символ операции. Большая часть операторов занимают один символ за исключением некоторых унарных. Применяется через специальный метод класса.

1. Action\_information

Хранит дополнительный символ если того требует операция. К примеру, перед записью знака суммы присутствует дополнительный символ обозначающий, к какому конкретно выражению эта сумма применяется. Данный символ будет храниться в данной ячейке и применяться в соответствии с требованием через тот же метод класса.

1. Execute\_index

Индекс выполнения данного выражения. Каждая формула разбивается на части до тех пор, пока каждая из частей не будет хранить одну бинарную или унарную операцию. Все части будут храниться в объектах класса Math\_act. Все объекты в сумме формируют последовательность выполнения, которая зависит от данного индекса. Чем меньше число, тем выше приоритет операции. Таким образом каждая из частей формулы будет иметь индекс от 0 до количества операций.

Методы:

1. ExecuteOperation

Данный метод применяется к одному из объектов класса Math\_act, который уже содержит в себе всю нужную информацию. Получив параметры и оператор, метод находит через оператора switch нужный эквивалент и применяет его к указанным параметрам объекта. В данном методе используется самые основные операторы, на самом деле данный метод может работать с на много большим числом операторов, большая часть представлена на рисунке 0. Каждый из операторов может предполагать более сложную логику, которую не выполнить за одно действие. Поэтому алгоритм вышел бы большой и слишком детализированный. Детали реализации будут рассмотрены в разделе реализации.

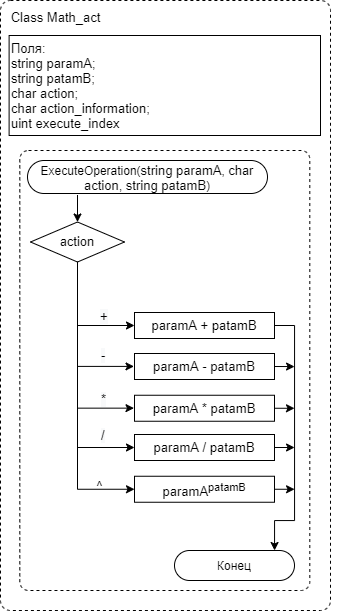


Рис. 7 - class Math\_act

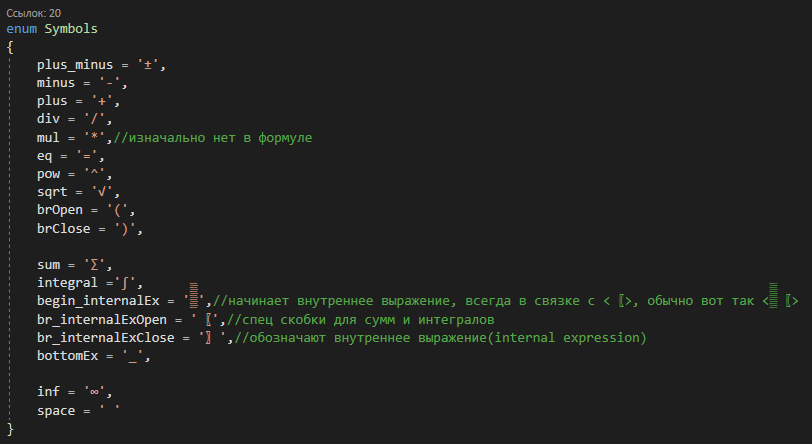


Рис. 8 - Enum c математическими операторами

1. GetSequencing

Ранее мы упоменали данную функцию, теперь рассмотим её чуть подробнее. На самом деле это лишь часть алгоритма по формированию последовательности вычисления формулы, рассмотренный очень обобщенно. Алгоритмы разделения формулы учитывают множество параметров и случаев, все эти проверки не указаны в алгоритме по причине обьема и малой целесообразности. Раздереление частей формулы и расставление приоритетов ориентировано на три основных параметра: Скобки и их вложенность, порядок действий слева на право и конечно же приоритеты математических операций как таковых. Операции бывают унарные, бинарные и тернарные. Чаще всего встречаются бинарные операции, то есть операции с двумя числами. Но, например такая операция как сумма Σ работает с одним аргуметом, при чем этот один аргумент может содержать несколько параметров и функций, но это не делает его двумя аргуметами. Для сохранения унарного оператора требуется сохранить один операнд, для бинарного два. Тернарный оператор не рассматривается в связи с отсутствием или не расспространенностью в данной области. Получаемые данные сохраняются в рассмотренные ранее поля класса.

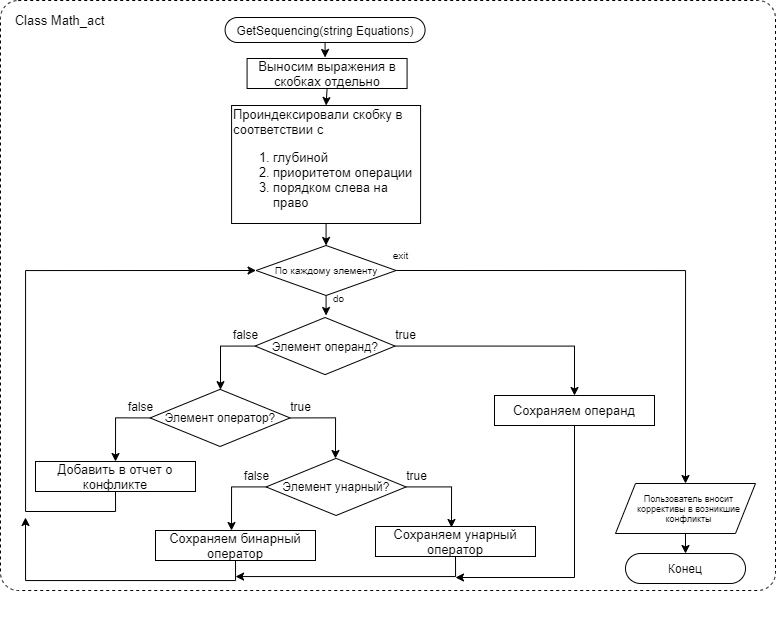
****

Рис. 9 - GetSequencing

**2.2Выводы:**

**3.Разработка программы.**

**Выводы:**

**4.Тестирование и экономическая эфктивность.**

**Вывод**

# Заключение

# Тезариус

# Список литературы:

Ссылки:

1. [Глобальная статистика интернета на 2020 год — отчет Digital о состоянии цифровых технологий в мире и в России (web-canape.ru)](https://www.web-canape.ru/business/internet-2020-globalnaya-statistika-i-trendy/)
2. [Веб-приложения вытеснят нативные? / Хабр (habr.com)](https://habr.com/ru/post/136070/)
3. [Перенос приложений "клиент-сервер" в WWW (interface.ru)](http://www.interface.ru/magazine/tcs/Archive/298/apps2www.htm)
4. [Будущее мобильной разработки — веб, а не приложения | Медиа Нетологии: образовательная платформа (netology.ru)](https://netology.ru/blog/future-web)
5. [Онлайновые вычислительные сервисы (upweek.ru)](https://upweek.ru/onlajnovye-vychislitelnye-servisy)
6. [Станут ли прогрессивные веб-приложения заменой нативным? | Nuances of programming | Яндекс Дзен (yandex.ru)](https://zen.yandex.ru/media/nuancesprog/stanut-li-progressivnye-vebprilojeniia-zamenoi-nativnym-5f64588461cbe322d9f174fb)
7. [PWA — это просто / Хабр (habr.com)](https://habr.com/ru/post/418923/)
8. [Превращаем веб-сайт в PWA | Nuances of programming | Яндекс Дзен (yandex.ru)](https://zen.yandex.ru/media/nuancesprog/prevrascaem-vebsait-v-pwa-5ef85cac6ca51851535cc722)
9. [Progressive Web Apps в WordPress • Иван Никитин и партнеры (ivannikitin.com)](https://ivannikitin.com/2018/08/17/pwa-in-wordpress/)
10. [Сайты в современном мире (proweb.ua)](https://proweb.ua/content/sajty_v_sovremennom_mire/)
11. [Классификация сайтов: 18 видов c примерами (altblog.ru)](https://altblog.ru/vidy_sajtov/)
12. [Интернет 2020 в России и мире: статистика и тренды | ADVIANA - разработка сайтов | Яндекс Дзен (yandex.ru)](https://zen.yandex.ru/media/id/5b3b4d8b04251900aa532351/internet-2020-v-rossii-i-mire-statistika-i-trendy-5e5ca553227f403871e7e957)

[Как росло количество веб-сайтов в мире – Мир – Коммерсантъ (kommersant.ru)](https://www.kommersant.ru/doc/4147760)