**Создание сервера и клиента для распределённых вычислений**

Автор: Завёрткин Михаил Александрович,

МБОУ Лицей №136

Руководитель: Валюхова Светлана Викторовна,

учитель информатики, высшая категория,

8-952-949-37-32

Новосибирск, 2021

Оглавление

[Введение 3](#_Toc88743135)

[Цели 3](#_Toc88743136)

[Задачи 3](#_Toc88743137)

[Первая программа на PascalABC.NET 4](#_Toc88743138)

[Разрабатываемая версия 4](#_Toc88743139)

[Архитектура 5](#_Toc88743140)

[Сервер 5](#_Toc88743141)

[Клиент 5](#_Toc88743142)

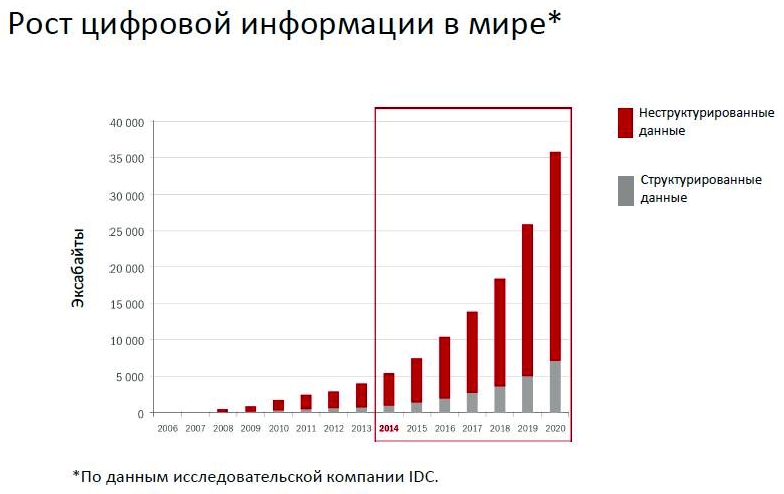
[Заключение 6](#_Toc88743143)

[Список использованной литературы 7](#_Toc88743144)

[Приложения 8](#_Toc88743145)

# Введение

В наши дни количество получаемой информации всё увеличивается, и эту информацию надо обрабатывать. Делать это всё труднее по мере усложнения алгоритмов обработки и увеличения наборов данных. Один из методов повышения эффективности вычислений – это их распределение. А учитывая тенденцию к росту сбора информации, отрасль её обработки не останется невостребованной.



# Цели

* Создать клиент-серверное приложение для проведения распределённых вычислений.

# Задачи

* Изучение недостающей для решения информации
* Написание программ сервера и клиента
* Отладка программы

# Первая программа на PascalABC.NET

Несколько лет назад я уже писал аналогичную программу, но на языке программирования PascalABC.NET. С начала 2020 года я начал изучать C# и программирование и увидел следующие проблемы:

* Программа была написана в функциональном стиле, что снижало возможности по её расширению
* Для изменения обрабатываемой задачи требовалось вручную менять файлы как на сервере, так и на клиенте
* Отсутствие возможности разными клиентами обрабатывать разные задачи

Исходники можно просмотреть по следующей [ссылке](https://disk.yandex.ru/d/mdxvyt-SWMIwzg)

С целью устранения данных недостатков я начал разработку полноценной программы, но на языке C# и с учётом новых знаний о программировании.

# Разрабатываемая версия

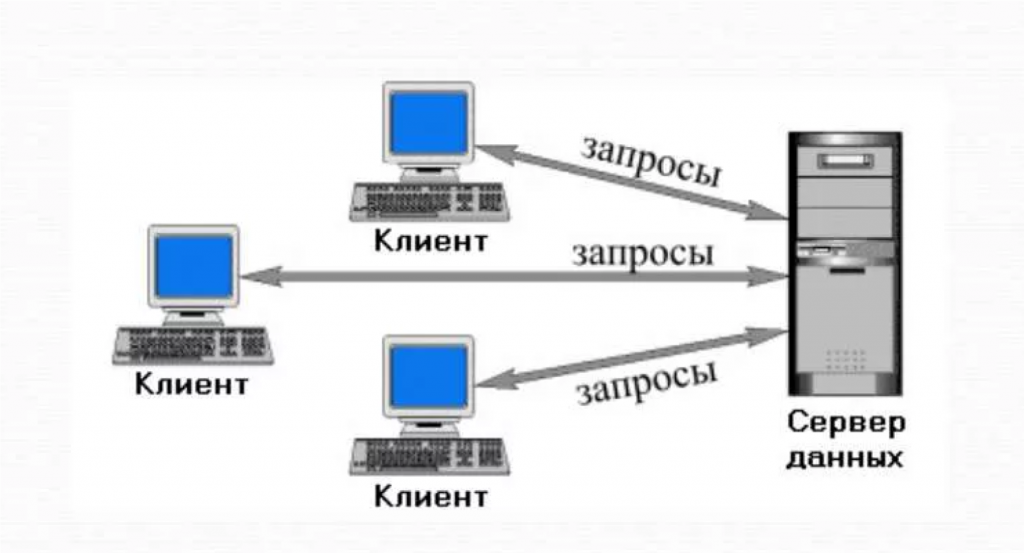
На данный момент готов код соединения между сервером и клиентом, а также набросок интерфейса

Новая версия будет лишена изъянов предшественника, так как она

* Написана с применением ООП
* Клиент способен сам менять обрабатываемую задачу по решению пользователя
* Сервер спроектирован с учётом возможности координации обработки нескольких задач одновременно

## Архитектура

Проект имеет клиент-серверную архитектуру, что означает наличие двух частей – сервера и клиента. Сервер координирует выполнение задач клиентами, которые используют вычислительные мощности локальных машин



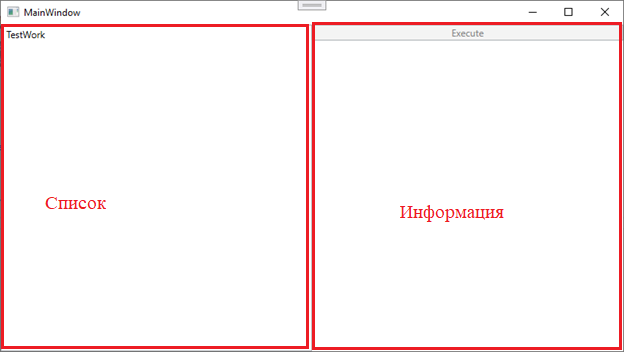
### Сервер

После запуска сервер находится в состоянии прослушивания входящих подключений от клиента. При появлении такого подключения запросы клиента обслуживаются до тех пор, пока он не отключится

### Клиент

Клиент построен с применением паттерна MVVM, что позволяет повысить гибкость приложения.   
При запуске происходит подключение к серверу и запрос списка имеющихся на сервере задач с их хэшем. Я использовал хэш-функцию SHA256. Далее по выбору пользователя одна из задач запускается. Если она уже загружена клиентом ранее, то с помощью сравнения хэшей, полученного от сервера и вычисленного клиентом, проверяется её целостность. В случае обнаружения повреждения данных, или их отсутствия, они загружаются с сервера и, выполняется задача.

Интерфейс разбит на несколько частей – список задач и подробная информация о выбранной задаче.



# Заключение

На данный момент имеется уже работающий прототип, который нетрудно довести до готового к пользовательскому применению состояния.

# Список использованной литературы

[Сайт, по которому я изучал C#](https://metanit.com/sharp/)

[Документация по C# и .NET](https://docs.microsoft.com/ru-ru/dotnet/csharp/)

Форумы – [StackOverflow](https://stackoverflow.com/) и [Cyberforum](https://www.cyberforum.ru/net-framework/)

Язык программирования C# 7 и платформы .NET и .NET Core - Джепикс Филипп, Троелсен Эндрю

# Приложения

[Репозиторий проекта](https://github.com/ProMix0/ProjectCalculis)