Муниципальное бюджетное общеобразовательное учреждение

города Новосибирска «Лицей № 136 имени Героя Российской Федерации

Сидорова Романа Викторовича»

Образовательная область: технология

Предмет: информатика

Создание сервера и клиента для распределённых вычислений

Исполнитель: Завёрткин Михаил Александрович,

11 класс

Руководитель: Валюхова Светлана Викторовна,

учитель информатики высшей категории

г. Новосибирск, 2022 г.

Оглавление

[Тезаурус 3](#_Toc95526587)

[Введение 4](#_Toc95526588)

[Актуальность 4](#_Toc95526589)

[Цель 4](#_Toc95526590)

[Задачи 4](#_Toc95526591)

[Распределённые вычисления 4](#_Toc95526592)

[1. Первая программа на PascalABC.NET 5](#_Toc95526593)

[2. Разрабатываемая версия 5](#_Toc95526594)

[2.1. Вспомогательные инструменты 6](#_Toc95526595)

[2.2. Архитектура 6](#_Toc95526596)

[2.2.1. Сервер 7](#_Toc95526597)

[2.2.2. Клиент 7](#_Toc95526598)

[Заключение 9](#_Toc95526599)

[Список использованной литературы 10](#_Toc95526600)

[Приложения 11](#_Toc95526601)

# Тезаурус

ООП – объектно-ориентированное программирование

Паттерн – шаблон

Абстракция – использование только тех характеристик, которые точно характеризуют объект в данной системе

Инверсия зависимостей – подход, при котором класс получает зависимости извне, вместо самостоятельного их разрешения

Логирование – запись информации о событиях, происходящих в рамках процесса

Флопс – единица измерения вычислительной мощности устройства. Измеряется в операциях с плавающей точкой в секунду (**FL**at **OP**eration**S**)

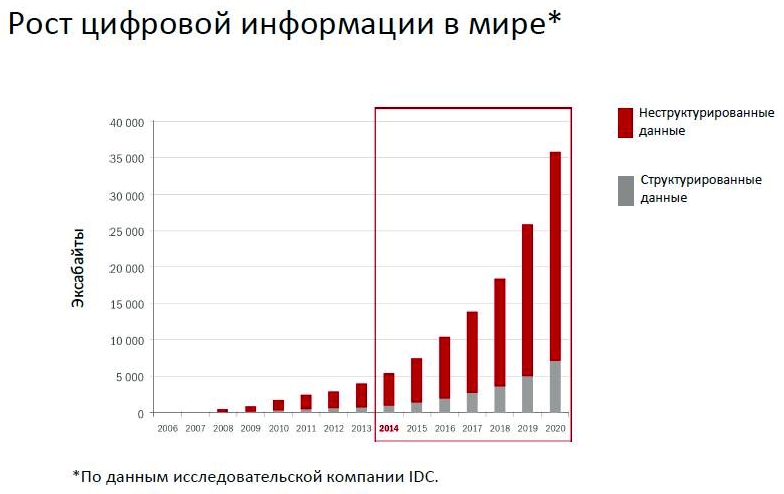
Петафлопс – флопс

Класс – набор данных и методов, описывающих реальный объект

# Введение

## Актуальность

По данным исследовательской компании IDC объёмы сбора и хранения цифровой информации в мире неуклонно увеличивается. А чем больше информации, тем больше требуется времени на её обработку. Следовательно, сейчас актуальность повышения эффективности вычислений велика как никогда.



## Цель

Написать приложение для проведения распределённых вычислений.

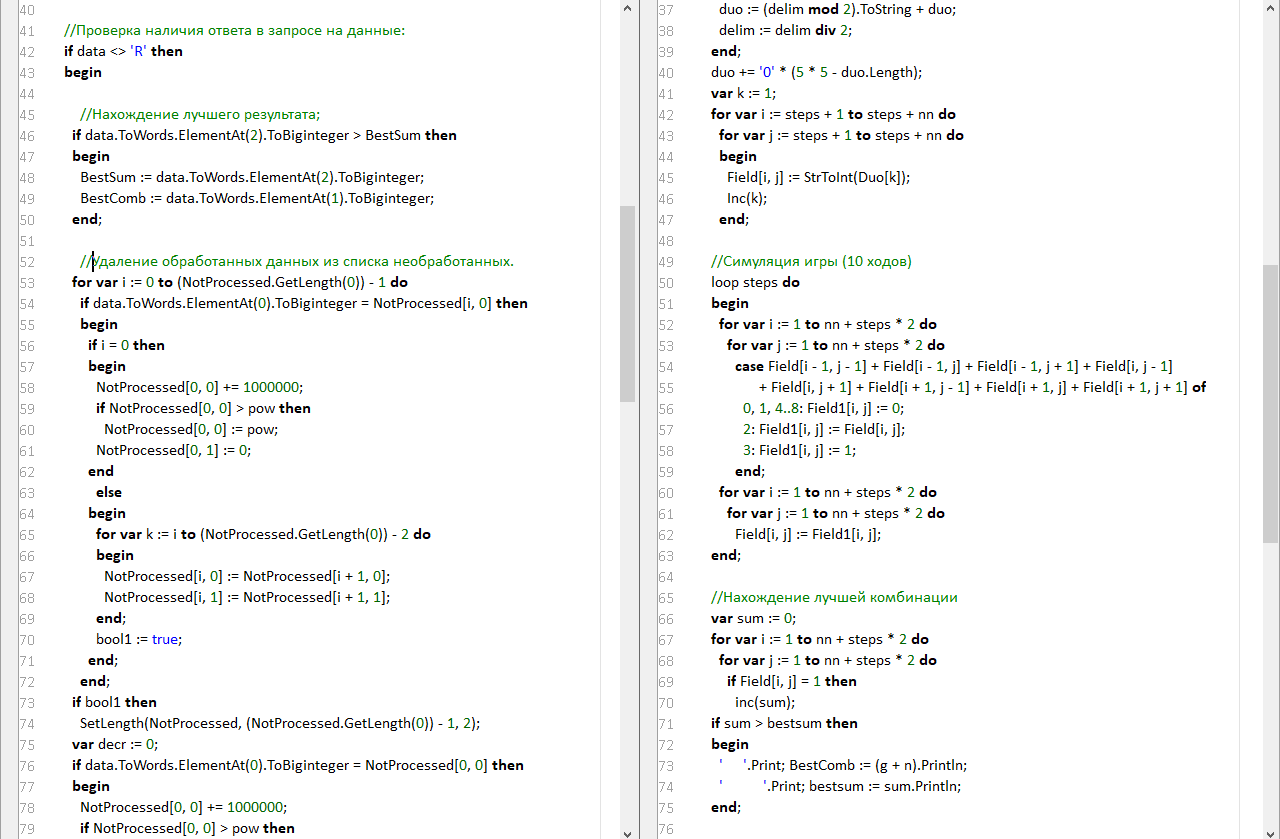
## Задачи

* Продумать структуру проекта
* Написать реализацию
* Найти и исправить ошибки

## Распределённые вычисления

Распределённые вычисления - это способ решения трудоёмких вычислительных задач с использованием нескольких компьютеров. Например, платформа BOINC объединяет домашние компьютеры обычных пользователей. Так можно добиться вычислительных мощностей, близких к таковым у суперкомпьютеров, но с куда меньшими затратами.

# Первая программа на PascalABC.NET

Несколько лет назад я уже писал аналогичное приложение, но на языке программирования PascalABC.NET. Это была программа в функциональном стиле, что сильно снижало её гибкость, из-за чего я пришёл к решению повторить функционал, но в виде, лучше отвечающем современным практикам программирования. 

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

# Разрабатываемая версия

Сейчас мною разрабатывается новое приложение, ориентированное на гибкость кодовой базы. Она достигается путём:

* использования ООП, в частности интерфейсов и абстракций;
* применения инъекции зависимостей с помощью стандартного DI-контейнера для реализации инверсии контроля;
* использования структурных паттернов, таких как MVVM, Factory.
* Внедрение логирования позволяет упростить процесс отладки.

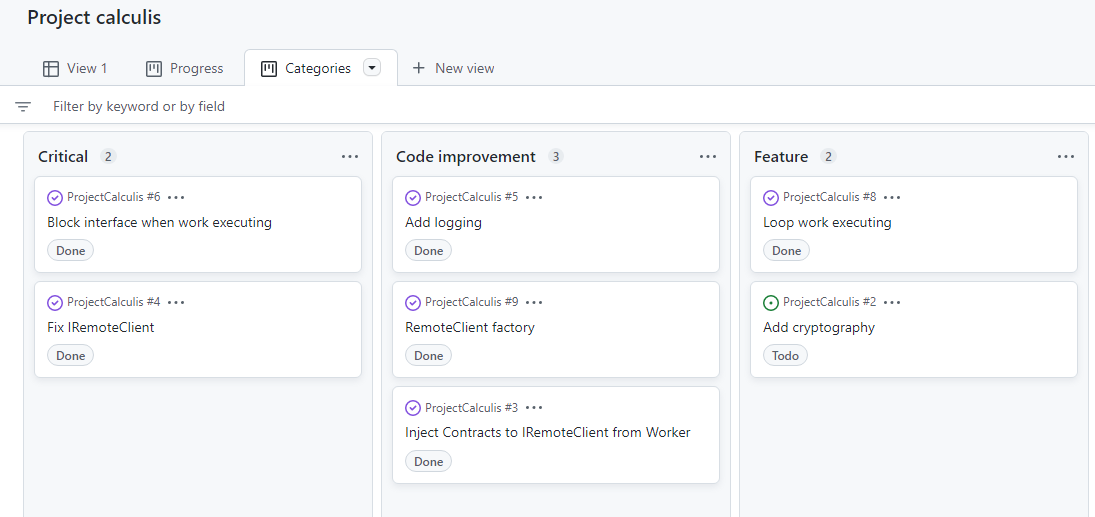
Широкое применение абстракций в программе позволяет избавиться от деталей реализации каждого компонента, что даёт возможность легко их менять, не нарушая работоспособность программы.

Инверсия контроля – подход, при котором класс не разрешает свои зависимости самостоятельно, а получает их извне. В сочетании с абстракциями позволяет менять компоненты, не затрагивая код. Контейнеры инъекции зависимости применяются для упрощения кода в системах, где количество классов может достигать сотен, а то и тысяч единиц.

Паттерны проектирования значительно упрощают разработку и упрощают коммуникацию между разработчиками.

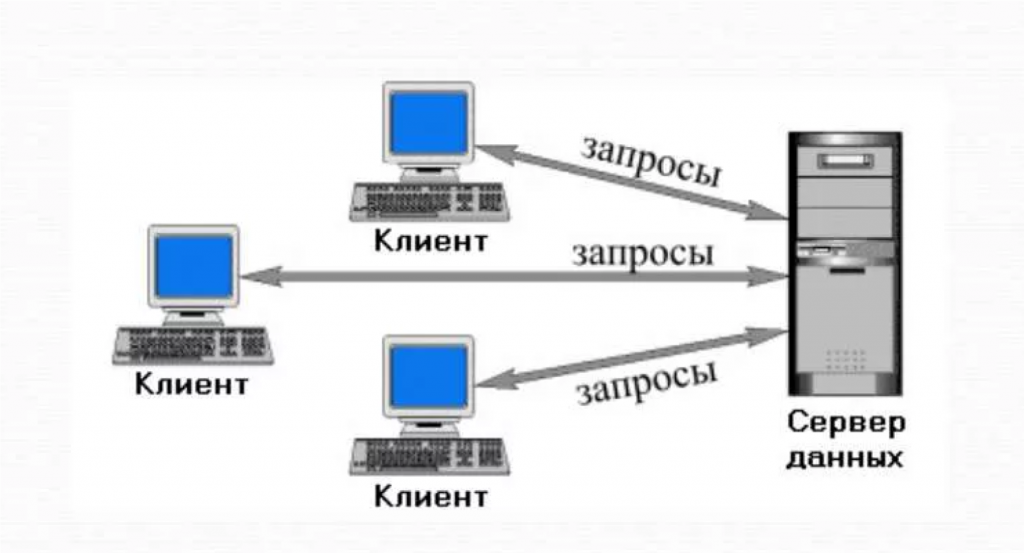
## Вспомогательные инструменты

Для планирования и организации работы широко использовались возможности GitHub, например, проекты.

Они позволяют группировать задачи по различным свойствам и организовывать их выполнение.

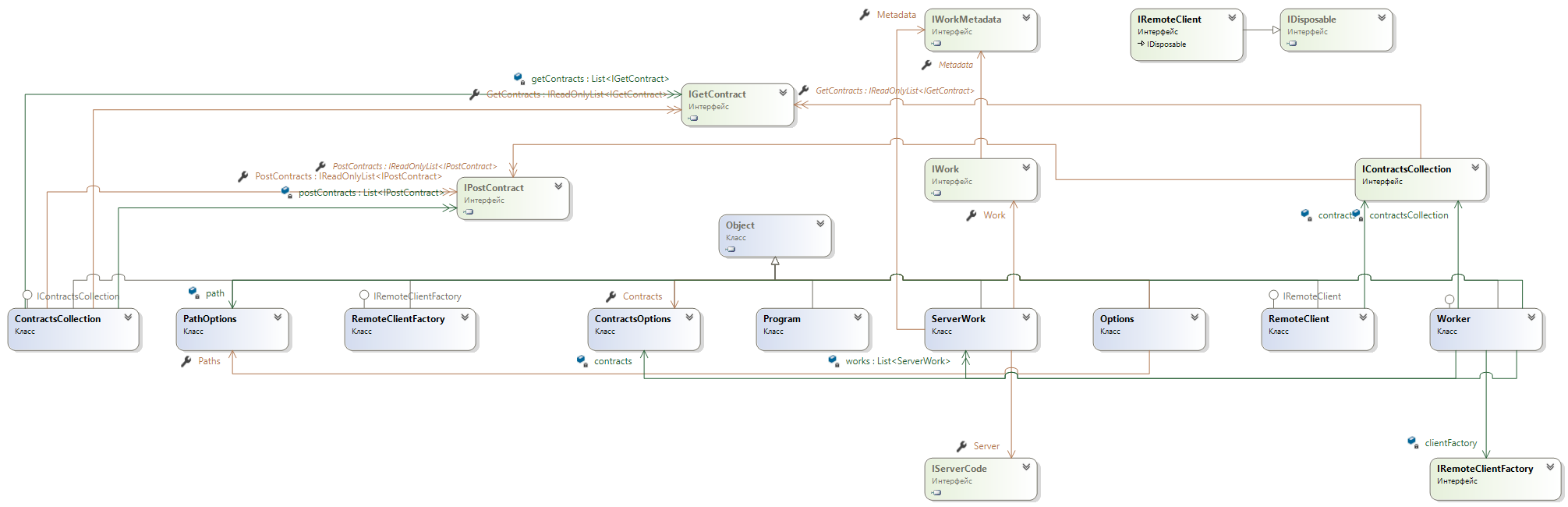
## Архитектура

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

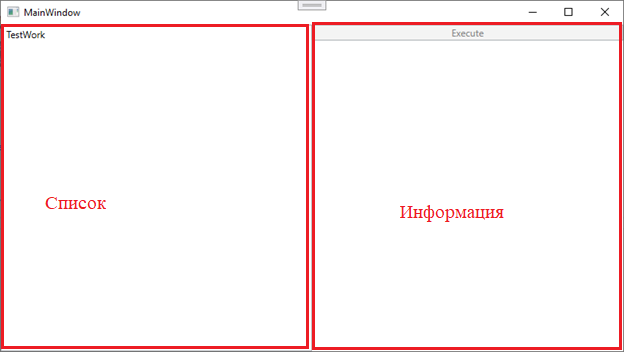
После запуска сервер находится в состоянии прослушивания входящих подключений от клиентов. При их появлении запрос обслуживается, и ожидается следующий. Благодаря многопоточности можно обслуживать сразу множество клиентов.



### Клиент

Клиент построен с применением паттерна MVVM, что позволяет сделать интерфейс и модель более слабосвязанными.

Интерфейс разбит на несколько визуальных частей – список задач и подробная информация о выбранной задаче. После запуска пользователь может выбрать из списка желаемую задачу для выполнения.



# Заключение

На данный момент разработан гибкий и простой для изменения рабочий прототип с большим потенциалом для улучшения. Моя программа может найти свою сферу применения как в любительских задачах, так и в более серьёзных исследовательских задачах.

# Список использованной литературы

[Metanit](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)