Учреждение образования

«Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

Лабораторная работа №4-6

**Распределенные информационные системы**

Выполнили:

Студенты 3 курса 3 группы ФИТ

Авхачёв Евгений,

Шедько Евгений

**Лабораторная работа №4**

В ходе лабораторной работы №4, было разработано 3 программы:

* CentralizedServer;
* FirstClient
* SecondClien

В рамках этой лабораторной работы, было реализовано протоколирование в 3 программа, которое выводит информацию о выполнении тех или иных действий.

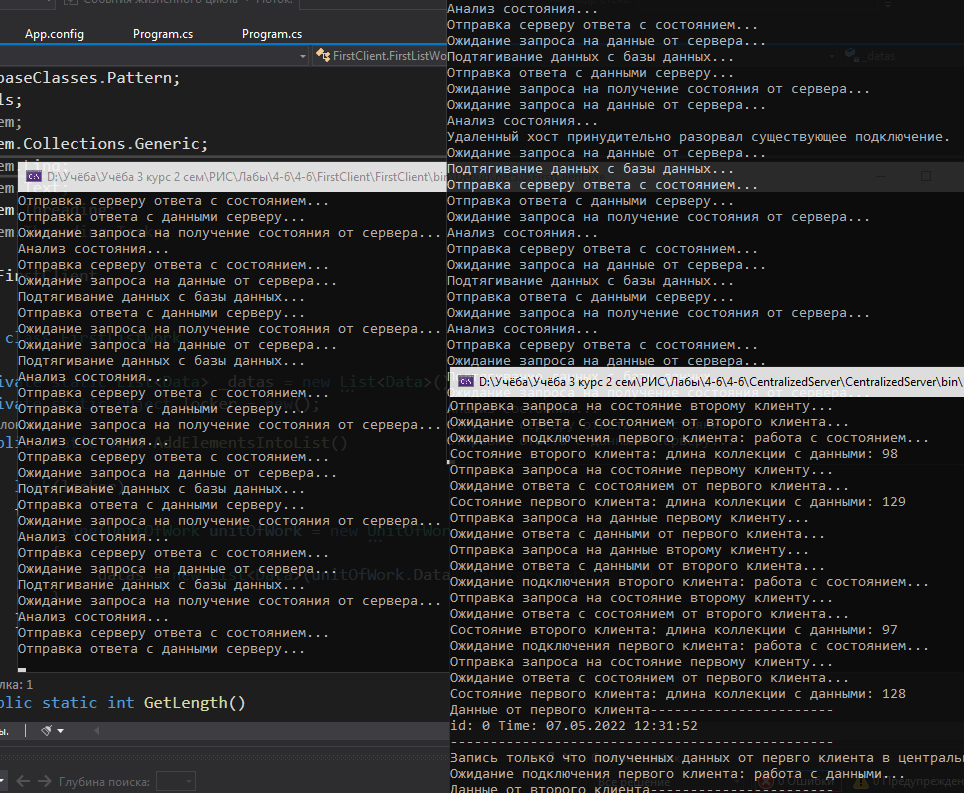


Рисунок 1 – Протоколирование

При запуске сервера, программа ожидает подключения клиентов, для обмена данными.

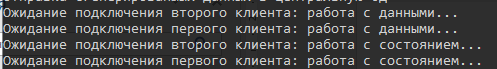


Рисунок 2 – Ожидание сервером подключения клиентов

После того, как было произведено подключение клиентов к серверу, начинается процесс обмена данными в рамках клиент-сервер.

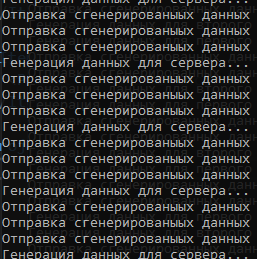


Рисунок 3 – Отправка данных клиентам, сгенерированных на сервере

Так же была создана обработка исключительных ситуаций, которая обрабатывает все внештатные ситуации, произошедшие во время работы программ.

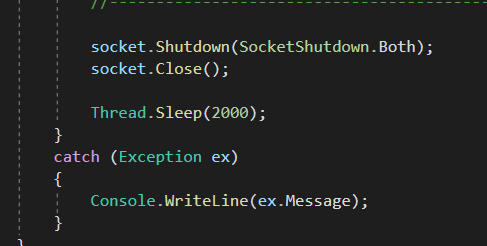


Рисунок 4 – Реализация обработки исключительных ситуаций

На следующем рисунке показана обработка ошибки, при неудавшемся подключении клиента к серверу.

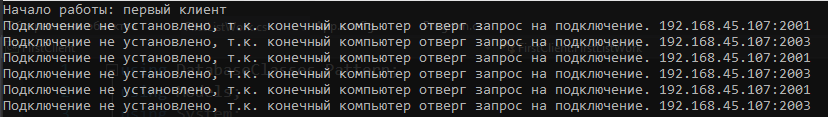


Рисунок 5 – Обработка ошибки неудавшегося подключения клиента к серверу

**Лабораторная работа №5**

Чтобы реализовать вытягивающую репликацию данных, были созданы 3 базы данных, одна из которых предназначается для сервера, вторая для первого клиента, а третья – для второго клиента.

Так же в каждой из баз данных была создана таблица для записи данных.

Для того, чтобы реализовать случайное заполнение данными, была разработана ещё одна программа, которая генерирует случайные данные и записывает их в базы данных.

Для этого был реализован паттерн UnitOfWork, с помощью которого в отдельном потоке создаются записи в базе данных.

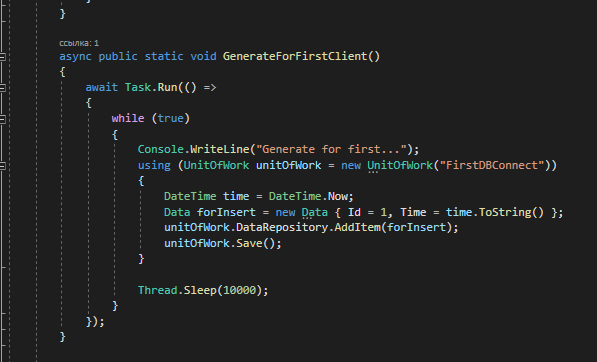


Рисунок 6 – листинг кода добавления записи в бд

Чтобы реализовать вытягивающую репликацию, в программе сервера было добавлено 2 потока, которые обмениваются данными с клиентами.

Чтобы получить данные для репликации, сервер отправляет запрос клиентам, которые, в свою очередь, отправляют ответ с данными в формате json.

Сервер переводит из формата json, в необходимый тип данных, и проверяет состояние, которое указывает, появились ли новые записи в базе данных клиента.

Если же отправленное состояние базы данных клиентом указывает, что появились новые данные, то сервер создаёт новый объект и из него отправляет данные в свою базу данны.

На рисунке 7 показана детальная информация, о выполнении сервером операций соединения с клиентами и записи данных в базу данных.

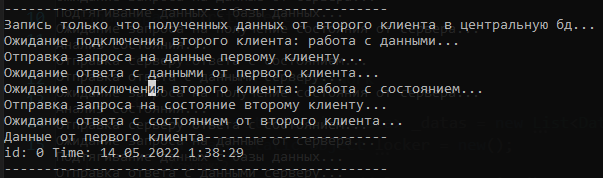


Рисунок 7 – Вывод информации сервером

Если вдруг возникнет исключительная ситуация во время подключения, то сервер будет выводить сообщение об ошибке такое, как показано на рисуноке 8.

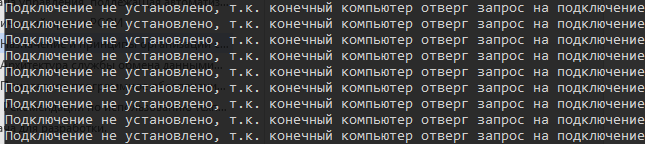


Рисунок 6 – вывод информации, при возникновении ошибок

**Лабораторная работа №6**

В шестой лабораторной работе, необходимо было реализовать репликацию данных.

Для этого в программе сервера был добавлен код, который связывается с базами данных клиентов и отправляет им сгенерированные данные.

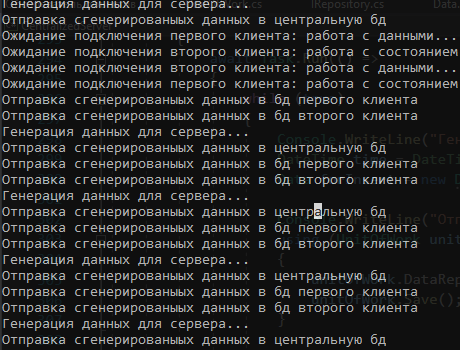


Рисунок 7 – Вывод данных о выталкивающей репликации данных

На рисунке 8 показаны записи, которые были добавлены в базы данных клиентов и сервера. Как видно по результату, то сервер действительно выполняет вытягивающую и выталкивающую репликацию данных.

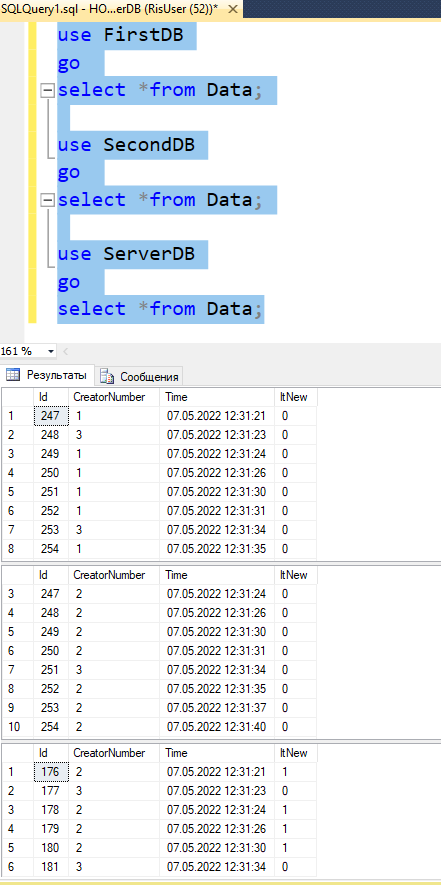


Рисунок 8 – добавление данных сервером в свою базу данных

**Вывод**: в результате лабораторной работы были реализованы два клиента, сервера. Была реализована запись в базу данных клиентами и сервером. Так же была реализована выталкивающая и вытягивающая репликации данных базы данных.