МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Кафедра «Информационная безопасность систем и технологий»

Курсовая работа
по дисциплине «Технологии и методы программирования»
на тему «Программная реализация сетевого сервера »
ПГУ.100502.С.1.О.23.КР.22ПТ107.01.ПЗ

Специальность — 10.05.02 Информационная безопасность телекоммуникационных систем.
Специализация — Разработка защищенных телекомуникацинных систем

Выполнил студент: Крестина С.Д.
Группа: 22ПТ1
Руководитель: Лупанов М.Ю.

2023

Цель работы

Целью данной курсовой работы является разработка серверной программы для клиент-серверной системы обработки данных.

Актульность работы

В современном мире сетевые серверы играют важную роль в передаче и обработке данных между клиентами и серверными системами. В связи с этим, разработка эффективных и надежных серверных программ становится необходимой задачей для многих компаний и организаций.

Задачи работы

Анализ требований к программе.

Построение UML-диаграмм вариантов использования и проектирование пользовательского интерфейса

Построение UML-диаграмм классов и планирование модулей.

Построение UML-диаграмм последовательностей.

Построение UML-диаграмм деятельности.

Разработка серверной программы.

Разработка модульных тестов и проведение модульного тестирования.

Разработка функциональных тестов и проведение приемочного тестирования.

Проведение документирования кода программы с использованием Doxygen.

Проектирование

В данной главе проведено проектирование программы. Были разработаны UML диаграммы следующих типов:

Диаграмма вариантов использования(прецедентов)

Диаграмма классов

Диаграмма деятельности

Диаграмма последовательности

Диаграмма вариантов использования

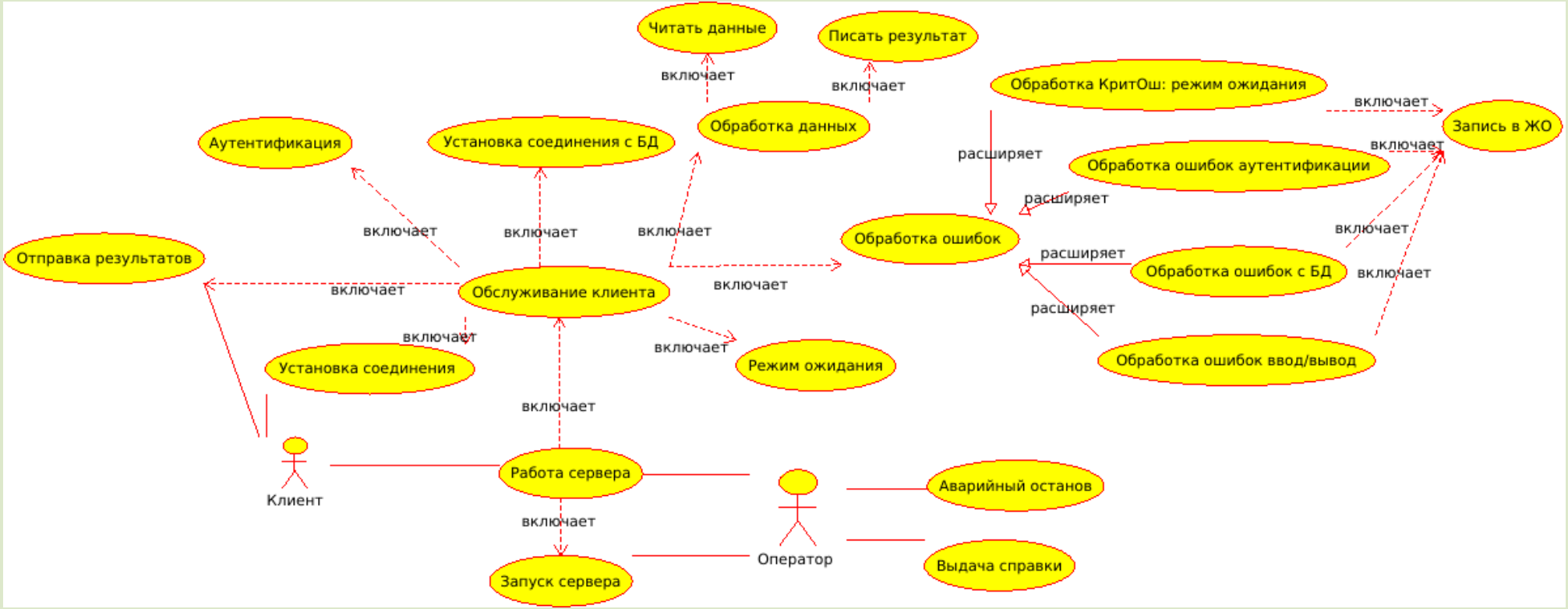


Диаграмма классов

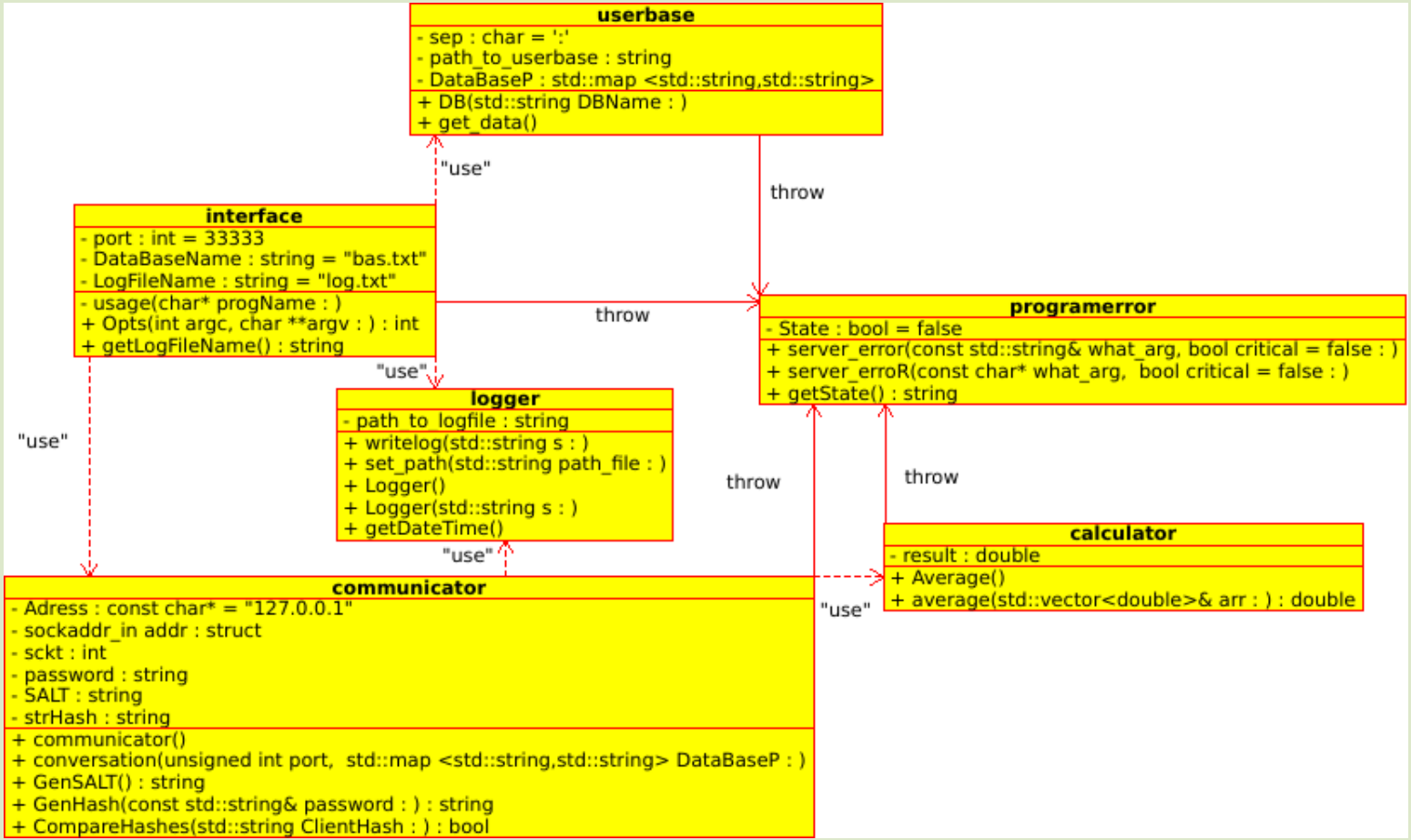


Диаграмма последовательности

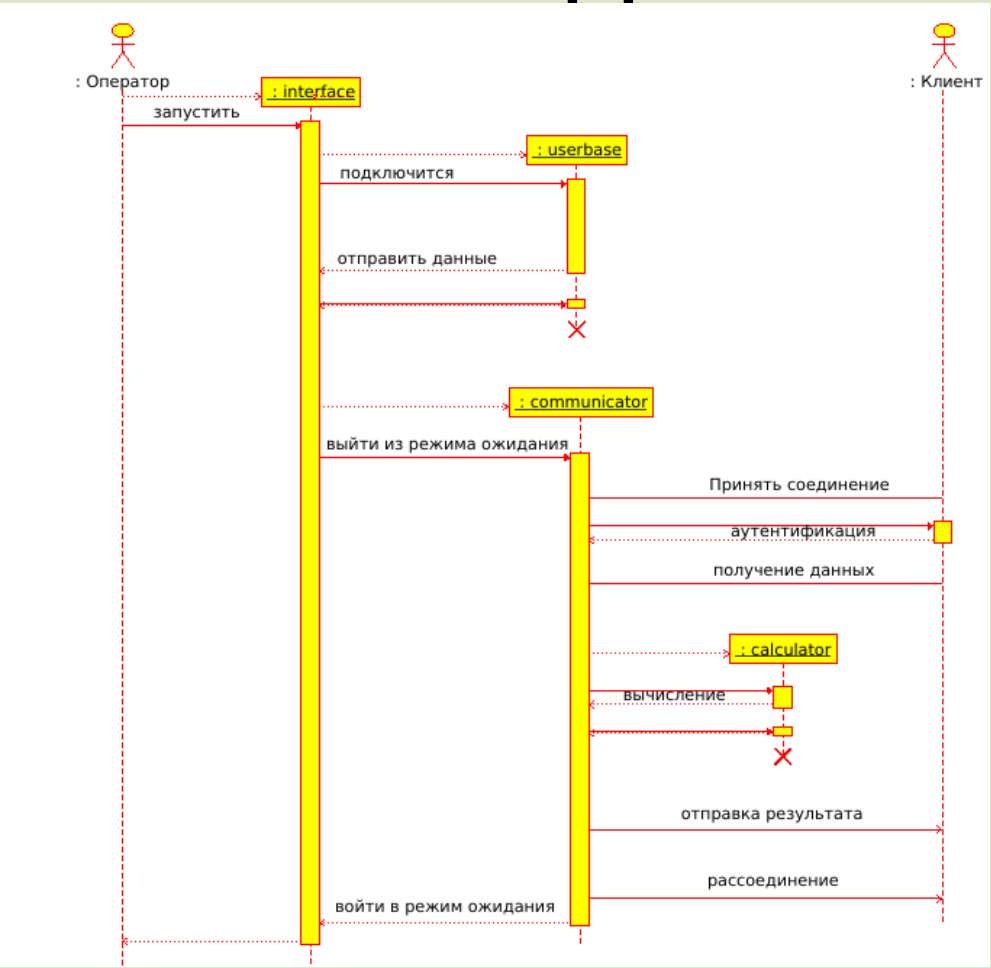
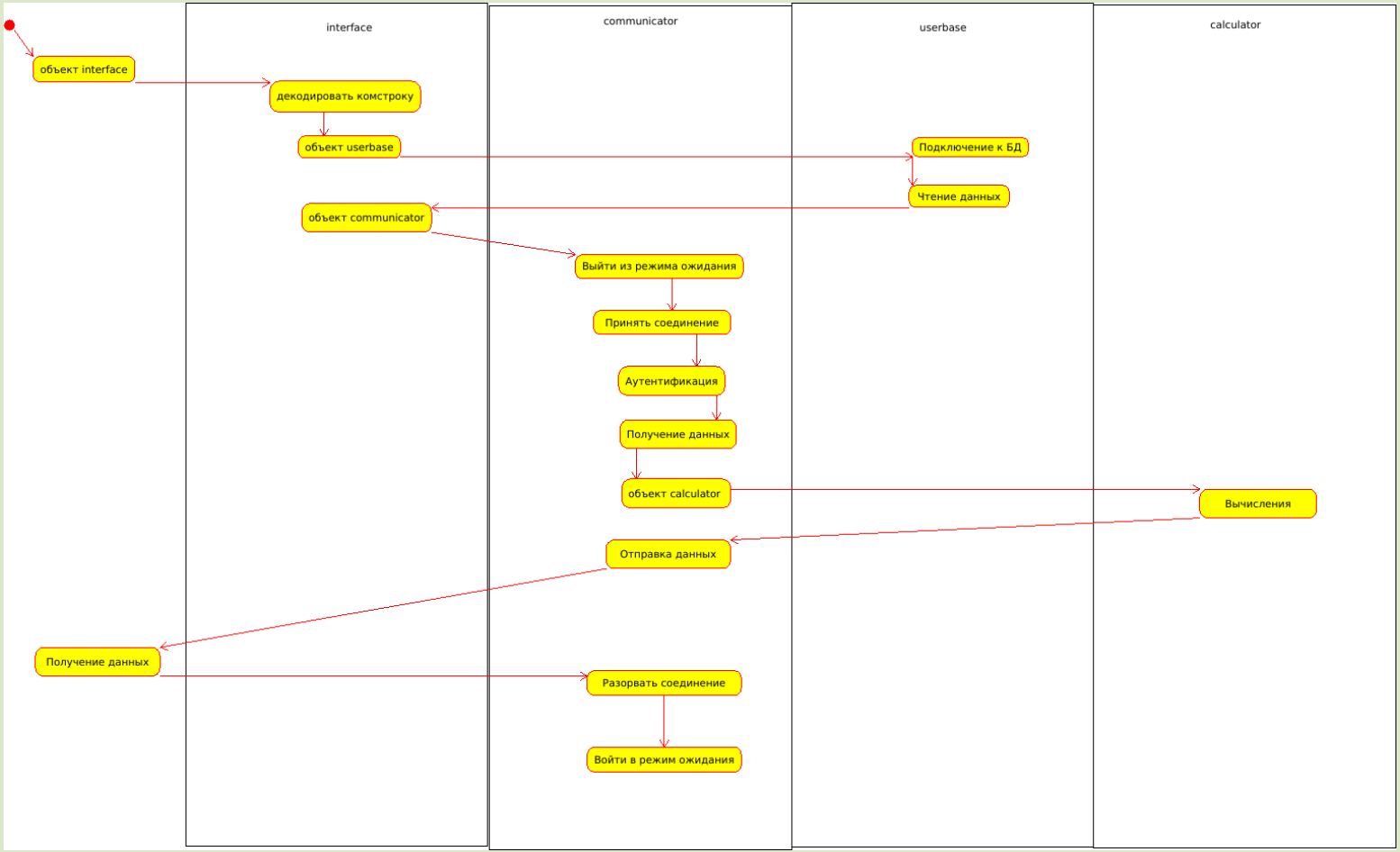


Диаграмма деятельности



Проектирование интерфейса

Таблица 1 - Проект командного интерфейса

Параметр	Значение	Функции	Умолчание	Зависимость
-b	string	Путь к файлу с базой клиентов	base.txt	Нет
-l	string	Путь у файлу с журналом работом	log.txt	Нет
-p	int от 1023 до 65536	Порт сервера	33333	Нет
-h	нет	Справка	При ошибке	Приоритет

Разработка программы

На данном этапе проведена разработка программы сервера на языке программирования C++ с применением классовой структуры.

Программа выполняет следующие функции:

чтение базы пользователей при старте программы;

обеспечение возможности подключения клиента в течении всего времени функционирования;

обработка запросов клиентов в однопоточном режиме, в том числе:

идентификаию и аутентификацию подключаемого клиента;

выполнение вычислений на данными, передаваемыми клиентом;

операция, выполняемая над данными — среднее арифметическое.

Разработка программы

В ходе работы были разработаны следующие модули:

calculator — отвечает за вычисления

communicator — отвечает за взаимодействия с клиентом

interface — отвечает за разбор ПКМ и включение модулей

logger — отвечает за запись логов

programmerror — обработка ошибок

userbase — отвечает за получение данных из базы данных

Модульное тестирование

При создании курсовой, на данном этапе, были разработаны модульные тесты для проверки корректности работы программы.

Модульное тестирование

Таблица 2 - Сценарии тестирования модуля коммуникации

№	Тест	Входные данные	Ожидаемый результат	Полученный результат
1.1	SIZE_SALT	Нет	true	true
1.2	BAD_SALT	«0000»	server_error	server_error
1.3	EMPTY_SALT	«»	server_error	server_error
1.4	BAD_PASS	«badpassword»	server_error	server_error
1.5	EMPTY_PASS	«»	server_error	server_error
1.6	NORM	«password»	true	true

Примечания:

«» в путь к файлу не входят

```
//Тесты сравнения хешей CompareHashes в модуле аутентификации
SUITE(HAHA){
    TEST(SIZE_SALT){
        communicator cm;
        cm.getpass("password");
        std::string salt = cm.GenSALT();
        CHECK(salt.size() == 16);
    }
    TEST(BAD_SALT){
```

Таблица 3 - Сценарии тестирования модуля вычисления

№	Тест	Входные данные	Ожидаемый результат	Полученный результат
2.1	EMPTY_VEC	Нет	server_error	server_error
2.2	MAX_OVER_VEC	{1.79769e+308, 1.79769e+308, 1.79769e+308}	inf	inf
2.3	MIN_OVER_VEC	{-1.79769e+308, -1.79769e+308, -1.79769e+308}	-inf	-inf
2.4	NORM_VEC	{1, 2, 3}	2	2

Модульное тестирование

Таблица 4 - Сценарии тестирования модуля подключения к базе данных

№	Тест	Входные данные	Ожидаемый результат	Полученный результат
3.1	EMPTY_PATH	«»	server_error	server_error
3.2	BAD_PATH	«no/base/file.txt»	server_error	server_error

Примечания:

«» в путь к файлу не входят

```
//Тесты открытия базы данных
SUITE(CLIENTIC){
    TEST(EMPTY_PATH){
         CHECK_THROW(DB(""), server_error);
    }

    TEST(BAD_PATH){
        CHECK_THROW(DB("no/base/file.txt"), server_error);
    }
}
```

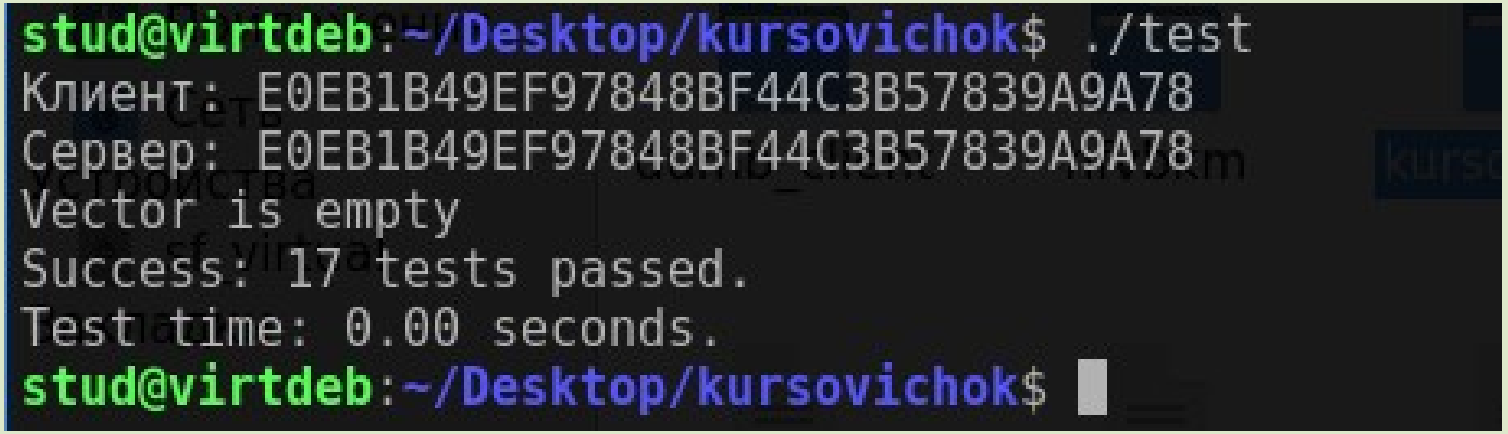
Таблица 5 - Сценарии тестирования модуля записи логов

№	Тест	Входные данные	Ожидаемый результат	Полученный результат
4.1	EMPTY_PATH_FILE	«»	invalid_argument	invalid_argument
4.2	VERY_BAD_PATH	«Путь/к/лог/файлу.txt»	invalid_argument	invalid_argument
4.3	EMPTY_MESSAGE	Нет	invalid_argument	invalid_argument
4.4	VREMYA	Нет	true	true
4.5	NORM	«Test message»	0	0

Примечания:

«» в путь к файлу не входят

Модульное тестирование



Функциональное тестирование

Таблица 6 - Сценарии тестирования разбора ПКС

№	Тест	Входные данные	Ожидаемый результат	Полученный результат
1.1	Все параметры	-l log.txt -b bas.txt -p 7777	Без исключений	Без исключений
1.2	Неверный путь до БД	-b ne/basa.txt	Исключение server_error	Исключение server_error
1.3	Неверный путь до ЖР	-l ne/log/log.txt	Исключение server_error	Исключение server_error
1.4	Некорректный порт	-p 88888888	Вывод справки	Вывод справки
1.5	Передача несуществующего параметра	-o	Исключение invalid option Вывод справки	Исключение invalid option Вывод справки
1.6	Вывод справки	-h	Вывод справки	Вывод справки

Функциональное тестирование

Таблица 7 - Сценарии тестирования стабильности работы сервера

№	Тест	Входные данные	Ожидаемый результат	Полученный результат
2.1	Клиент с типом данных double	Нет	Стабильная работа	Стабильная работа
2.2	Клиент с типом данных int32_t	Нет	Стабильная работа	Стабильная работа

```
stud@virtdeb:~/Desktop/kursovichok/code$ for i in range{1..1000}; do ./client_double; done
```

```
stud@virtdeb:~/Desktop/kursovichok/code$ for i in range{1..1000}; do ./client_int32_t; done
```

Документирование

- Пример документирования

communicator.cpp × calculator.h ×

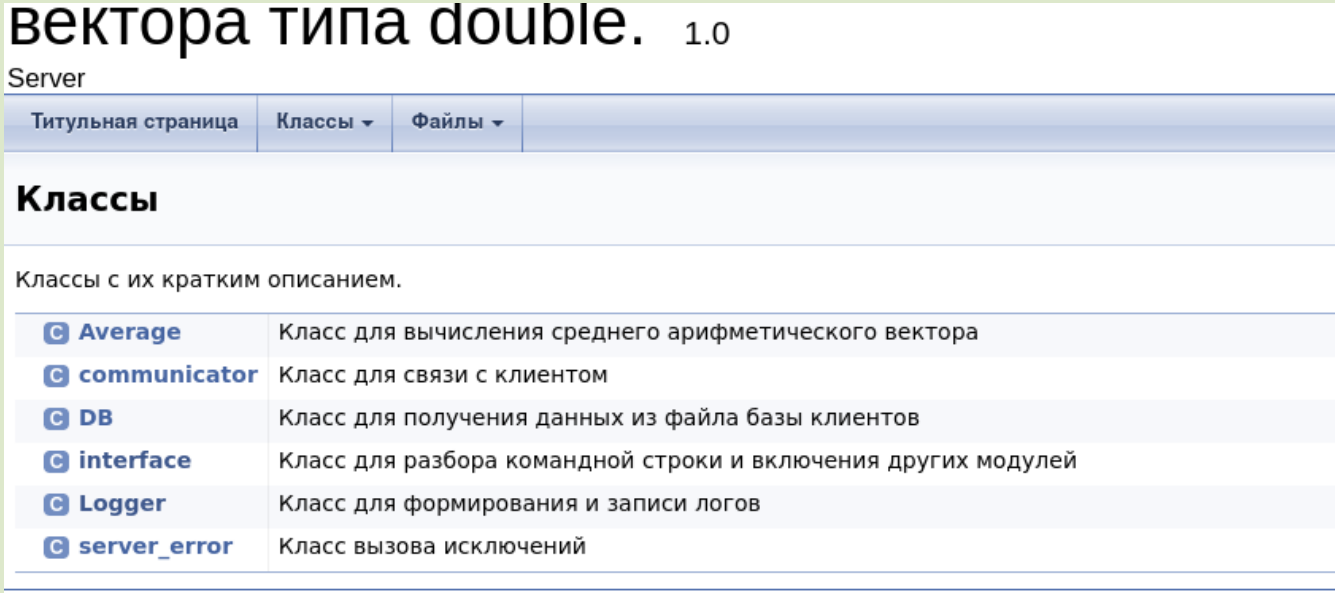
/home/stud/Desktop/kursovichok/code/

```
/**
* @file calculator.h
* @author Крестина С.Д.
* @version 1.0
* @brief Заголовочный файл для модуля calculator, отвечающий за вычисления
*/

#pragma once
#include <vector>
#include <iostream>
#include <exception>
#include <limits>
#include "programmerror.h"


/**
 * @brief Класс для вычисления среднего арифметического вектора
 * @details Метод для вычисления
 */
class Average
{
private:
    double result; ///< Результат вычислений
public:
    /**
    * @brief Конструтор класса без параметров
    */
    Average(){};
     /**
    * @brief Функция вычисления среднего арифметического вектора
    * @param [in] vector<double>& arr вектор со значениями типа double
    * @return result
    */
    double average(std::vector<double>& arr);
};
```

Документирование



Программная реализация сетевого сервера. Сервер, вычисляющий среднее арифметическое вектора типа double.

1.0

Создано системой Doxygen 1.9.1

Заключение

В ходе работы над курсовым проектом была осуществлена программная реализация сетевого сервера. Было построено 4 диаграммы и реализовано 6 модулей.

В первой главе проведено проектирование программы.

Во второй главе проведена разработка кода программы.

В третей главе проведено модульное тестирование.

В четвёртой главе проведено функциональное тестирование программы

В пятой главе написана документация программы.

Во время выполнения курсовой работы было разработано и протестировано серверное приложение с использованием TCP протокола и сокетов в Linux. В результате работы были созданы диаграммы, написан код сервера и тесты для проверки его работоспособности, а также составлена техническая документация и отчёт о проделанной работе. Все компоненты программы были сохранены на GitHub.

Репозиторий GitHub: https://github.com/KrestinaSD/kursovichok

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Кафедра «Информационная безопасность систем и технологий»

Курсовая работа
по дисциплине «Технологии и методы программирования»
на тему «Программная реализация сетевого сервера »
ПГУ.100502.С.1.О.23.КР.22ПТ107.01.ПЗ

Специальность — 10.05.02 Информационная безопасность телекоммуникационных систем.
Специализация — Разработка защищенных телекомуникацинных систем

Выполнил студент: Крестина С.Д.
Группа: 22ПТ1
Руководитель: Лупанов М.Ю.

2023