

Федеральное государственное автономное образовательное учреждение
высшего образования

Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)

Кафедра МИТ

ОТЧЕТ
по практической работе №5

по дисциплине "Объектно-ориентированное программирование"

на тему «Создание сервера»

Выполнил: студент группы 4376
ФИО: Шульгин А.Н.

Санкт-Петербург

2025

Введение:

Цель работы: Создать распределенное приложение, включающее клиентскую и серверную части, взаимодействующие посредством сетевого обмена сообщениями для работы с полиномами с комплексными коэффициентами.

Постановка задачи:

Требования к приложению:

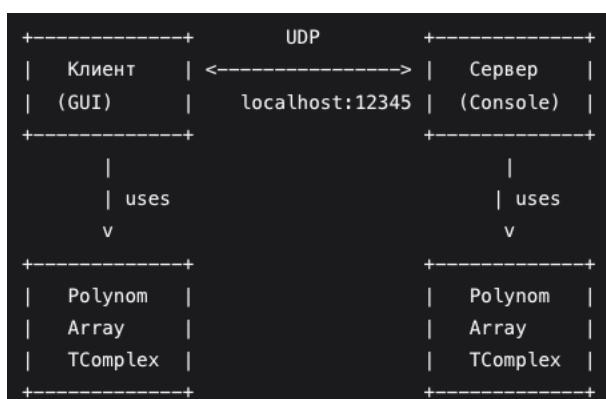
- Клиентская часть - GUI приложение с интерфейсом аналогичным работе №5
- Серверная часть - консольное приложение для выполнения функций над полиномами
- Взаимодействие через сетевой обмен сообщениями по протоколу UDP
- Использование классов polynom, array, complex из предыдущей работы
- Возможность расширения функциональности без изменения существующей реализации

Функциональные требования:

- Ввод и отображение полинома в двух формах (коэффициенты и разложение на множители)
- Вычисление значения полинома в точке
- Добавление и удаление корней полином
- Сетевое взаимодействие между клиентом и сервером

Архитектура системы:

Компонентная диаграмма:



Протокол взаимодействия:

- Транспортный протокол: UDP
- Порт по умолчанию: 12345
- Формат сообщений: QDataStream с сериализацией Qt

Диаграмма классов:

Клиентская часть:

```
MainWindow (QMainWindow)
  +-- NetworkInterface (наследник)
    |   +-- Communicator
    |   |   +-- QUdpSocket
    |   |   +-- QTimer
    |   +-- GUI компоненты
  +-- Polynom (через сеть)
```

Серверная часть:

```
Server (QObject)
  +-- Communicator
    |   +-- QUdpSocket
  +-- Polynom
    +-- Array
      +-- std::vector<number>
  +-- TComplex
```

Диаграмма классов Polynom:

```
└─ aN: number
└─ roots: Array
└─ coeffs: Array
└─ computeCoeffs(): void
└─ valueAt(x: number): number
└─ getForm1String(): string
└─ getForm2String(): string
```

Реализация:

1. Расширение существующих классов

Обоснование изменений: Для обеспечения сетевого взаимодействия были добавлены методы сериализации в класс TComplex:

// В complex.h добавлены:

```
friend QDataStream &operator<<(QDataStream &out, const TComplex &complex);
friend QDataStream &operator>>(QDataStream &in, TComplex &complex);
```

2. Новые классы для сетевого взаимодействия

Communicator - абстракция над QUdpSocket:

```
class Communicator : public QObject {
```

// Обеспечивает отправку и получение UDP-пакетов

// Инкапсулирует логику работы с сокетом

```
};
```

Server - обработчик запросов:

```
class Server : public QObject {
```

// Обрабатывает команды: SET_POLYNOM, GET_FORM1, GET_FORM2, CALCULATE и др.

// Выполняет вычисления с полиномами

```
};
```

NetworkInterface - клиентский интерфейс:

```
class NetworkInterface : public MainWindow {  
    // Наследует GUI от MainWindow  
    // Добавляет сетевую функциональность  
    // Обрабатывает таймауты и ошибки сети  
};
```

3. Протокол обмена сообщениями

Команды клиента → сервера:

- **SET_POLYNOM** - установить полином
- **GET_FORM1** - получить представление в форме коэффициентов
- **GET_FORM2** - получить представление в форме корней
- **CALCULATE** - вычислить значение в точке
- **ADD_ROOT** - добавить корень
- **REMOVE_ROOT** - удалить корень

Ответы сервера → клиента:

- **SUCCESS** - операция выполнена успешно
- **FORM1 + строка** - представление полинома
- **FORM2 + строка** - представление полинома
- **RESULT + number** - результат вычисления
- **ERROR + строка** - сообщение об ошибке

Тестирование:

Полином: $p(x) = 1 * (x - (1+0i)) * (x - (2+0i))$

Вычисление: $p(3) = (3-1)*(3-2) = 2$

Результат: Совпадает с ожидаемым

Действие: Отключение сервера во время работы клиента

Результат: Клиент обнаруживает таймаут (5 сек)

Показывает сообщение об ошибке

Восстанавливается при перезапуске сервера

Заключение:

Достигнутые результаты:

- Разработано полнофункциональное распределенное приложение
- Реализовано сетевое взаимодействие по протоколу UDP
- Обеспечена совместимость с существующими классами
- Реализован графический интерфейс пользователя
- Обеспечена обработка ошибок и таймаутов

Разработанное приложение успешно решает поставленную задачу создания распределенной системы для работы с полиномами. Архитектура "клиент-сервер" с использованием UDP протокола доказала свою эффективность для данного типа приложений. Все функциональные требования выполнены, система готова к использованию.