УДК 004.93

# СЕТЕВАЯ ИНФОРМАЦИОННАЯ СИСТЕМА «РАСПИСАНИЕ ЗАНЯТИЙ»

Белявская Н. С., Мироненко К. А.
Научный руководитель - Белова С.В.

Цель работы – проектирование и реализация информационной системы «Расписание занятий».

Программа представляет собой клиент-серверное многопользовательское приложение, в котором реализованы следующие функции:

- вход в систему под логином и паролем;
- просмотр всего расписания;
- просмотр расписания по определенному предмету, фамилии преподавателя, номеру группы.

Логины (в качестве них взяты номера зачеток студентов) и пароли к ним хранятся в текстовом файле на сервере. Также на сервере хранится файл, содержащий расписание занятий. Клиент и сервер обмениваются сообщениями, кодированными при помощи стандарта UTF-8.

В проекте используется соединение по протоколу TCP. Преимуществами данного протокола являются строгий порядок и надежность передачи данных.

В процессе работы были реализована технология применения асинхронных сокетов. Асинхронность в программировании — выполнение процесса в неблокирующем режиме системного вызова, что позволяет потоку программы продолжить обработку.Главная идея асинхронного программирования заключается в том, чтобы запускать отдельные вызовы методов и параллельно продолжать выполнять другую работу без ожидания окончания вызовов.

Программный код серверной части разбит на несколько модулей, реализующих соответствующие своему назначению методы.

В модуле Listener.cs создается сокет listener по протоколу TCP, который будет использоваться для прослушивания попыток соединения клиента. При помощи команд Bind и Listen и функции обратного вызова AcceptCallback происходит поиск клиента и соединение с ним. В этой части программного кода используется метод WaitOne(), отвечающий за блокировку данного потока и метод Set(), устанавливающий, что операция в данном потоке завершена. Методы обратного вызова требуются асинхронным операциям, чтобы вернуть результат.

В модуле ClientController.cs в список клиентов List<ClientObject> происходит добавление и удаление клиентов.

В модуле Sender.cs происходит кодирование считанных из файла данных и пересылка их клиенту. Здесь используется функция обратного вызова SendCallback. Также в методе этого модуля происходит вывод на консоль сервера, какие данные были высланы какому клиенту (по его имени). Аналогична было произведено проектирование модуля ReceivePackage.cs, который отвечает за получение и раскодирование данных.

Интерфейс клиентской части содержит в себе обработчик событий, возникающих в процессе взаимодействия пользователя с интерфейсом. Обработчик на стороне клиента обеспечивает получение данных пользовательского ввода, кодирование и отправку на сервер. При появлении нового сообщения (списка занятий) он обновляет соответствующие поля пользовательского интерфейса.

Основной клиентский модуль - модуль Form1.cs и его Windows-форма. Они отвечают за получение запросов клиента и связь клиент-сервер. Функция, обрабатывающая нажатие на клавишу «Показать», запускает методы Send() и Receive(), которые обмениваются с сервером необходимой информацией, используя блокирование потоков при помощи команды WaitOne(). Здесь также используются callback-функции.

UML-диаграмма этого класса и класса Listenerсерверной части представлена на рисунке 1.

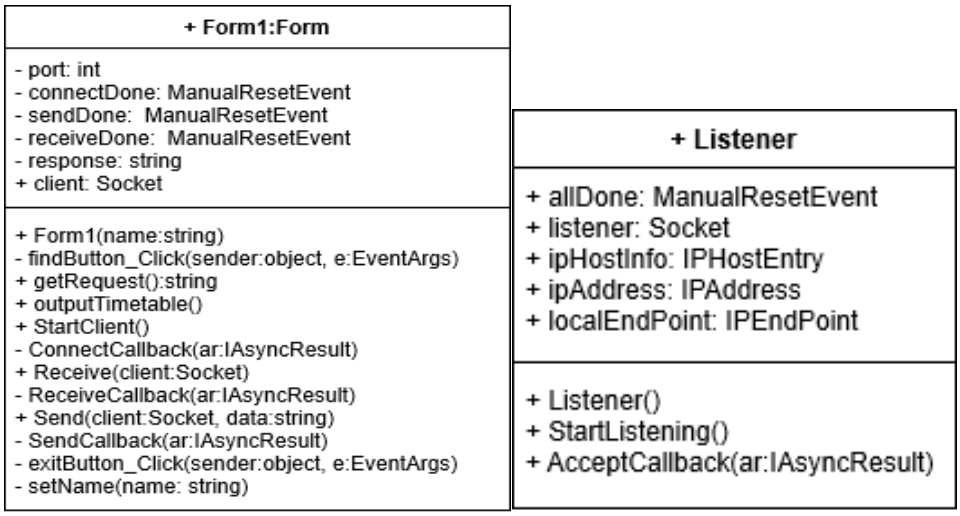


Рисунок 1 – UML-диаграммы классов Listener и Form1

Приложение содержит user-friendly интерфейс. Для входа в систему (рисунок 2) требуется ввести свой логин (который будет отображаться на сервере) и пароль, а также IPсервера.

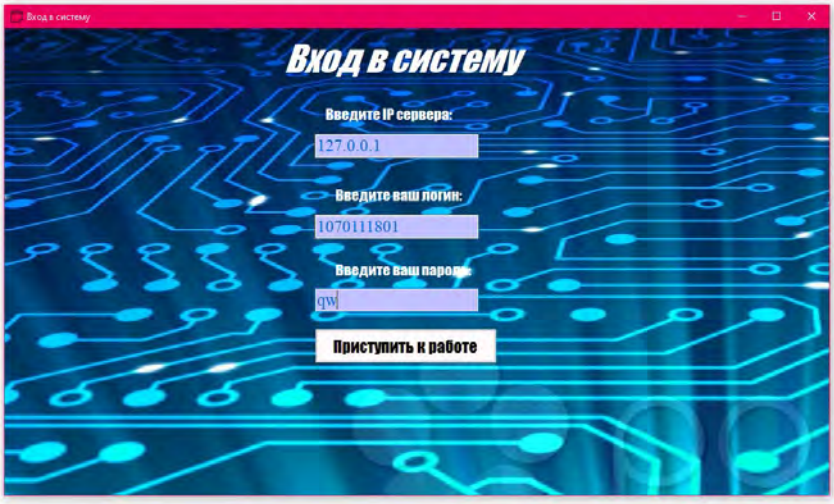


Рисунок 2 - графический интерфейс начального окна программы

На следующей форме (рисунок 3) можно выбрать группу, предмет, преподавателя или несколько из этих факторов отбора и также нажать на кнопку «Показать» для отображения расписания. Расписание выводится в удобной форме в виде таблицы.


Рисунок 3 – основная форма приложения

Разработанный программный продукт предназначен для хранения и выборки для просмотра необходимой информации по расписанию занятий и может быть использован на практике в процессе обучения.