Министерство образования и науки Российской
Федерации
Федеральное государственное бюджетное
образовательное учреждение высшего образования
"Петрозаводский государственный университет"
Институт математики и информационных технологий
Кафедра информатики и математического обеспечения

ОТЧЕТ
по дисциплине "Верификация ПО"
на тему: "Тестирование TCP/IP-чата"

Выполнила студентка
группы 22608:

Дятлова Александра Михайловна

Преподаватель:

Кулаков Кирилл Александрович

Петрозаводск
2018

Оглавление

План тестирования

Объект тестирования

Объектом тестирования является TCP-чат на C# и .NET, с использованием TCP сокетов. Приложение работает по типу ICQ и позволяет пользователям мгновенно обмениваться текстовыми сообщениями. Клиенты работают на платформах ОС Windows. Через сервер осуществляется поиск и связь с другими клиентами, а также обмен сообщениями между пользователями. Помимо отправки текстовых сообщений пользователями в общий чат, есть возможность посылать приватные сообщения конкретным пользователям. В ходе работы будут тестироваться клиентская и серверная части.

Проект C# SGSserverTCP и SGSclientTCP предназначен для платформы .NETFramework, Version=v2.0.

Функциональность объекта тестирования

Приложение подразумевает подключение нескольких пользователей, каждый из которых подключается по IP-адресу сервера и заходит под своим ником.

Общий функционал приложения:

1. Авторизация
2. Выход из системы
3. Отправка сообщений в чат всем пользователям
4. Отправка приватных сообщений конкретному пользователю
5. Получение сообщений в чате
6. Получение списка всех пользователей чата
7. Звуковое оповещение при входе в систему, отправке и получении сообщений

<u>В аттестационном тестировании принимают участие все вышеперечисленные функции.</u>

Интерфейс приложения

На рисунках ниже представление интерфейс приложения.

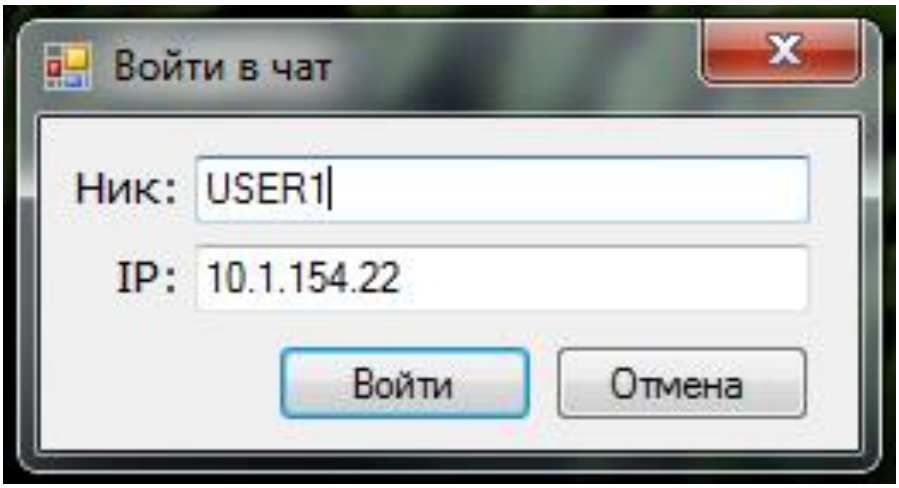


Рис. 1: Окно авторизации клиента в чате

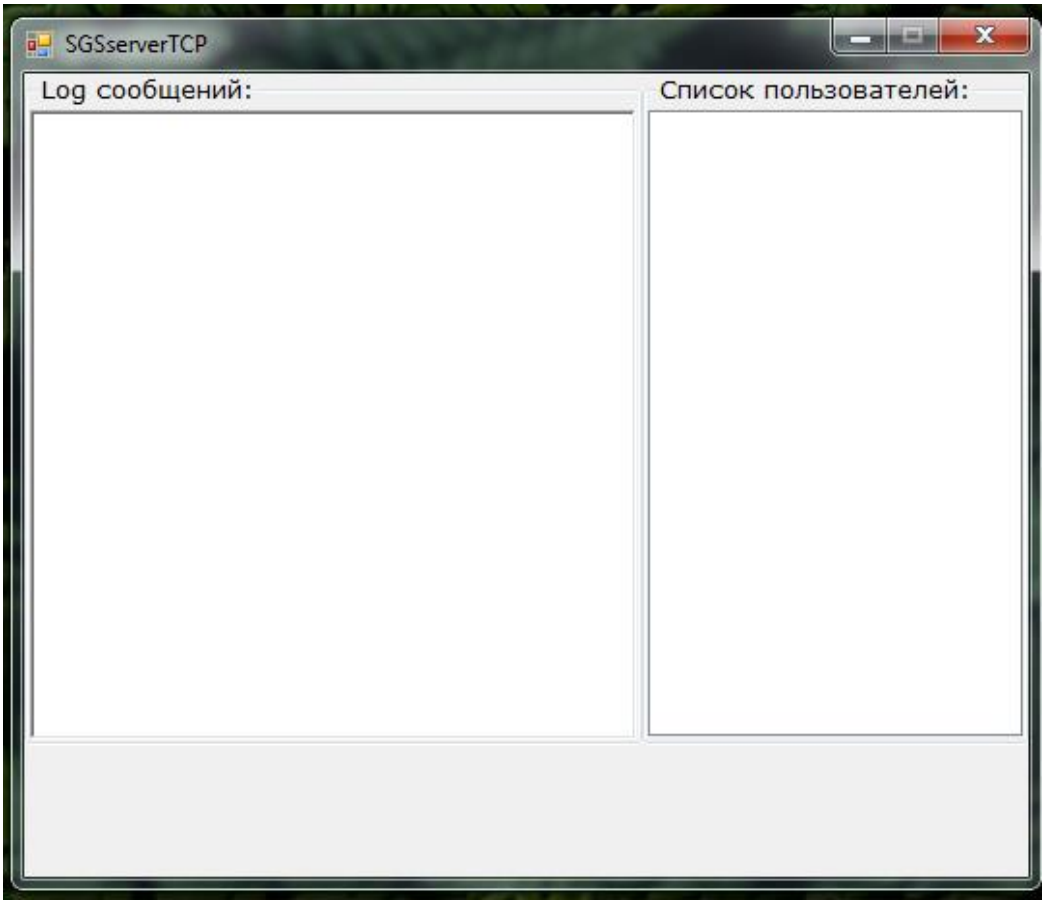


Рис. 2: Сервер (отображение логов сообщений и списка подключенных пользователей)

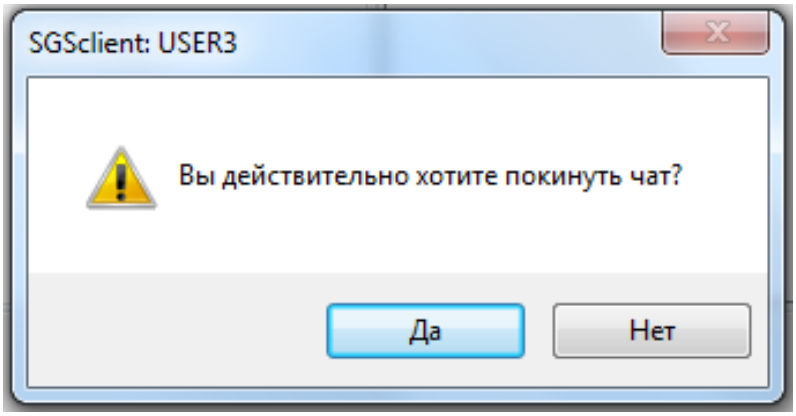


Рис. 3: Окно выхода клиента из чата

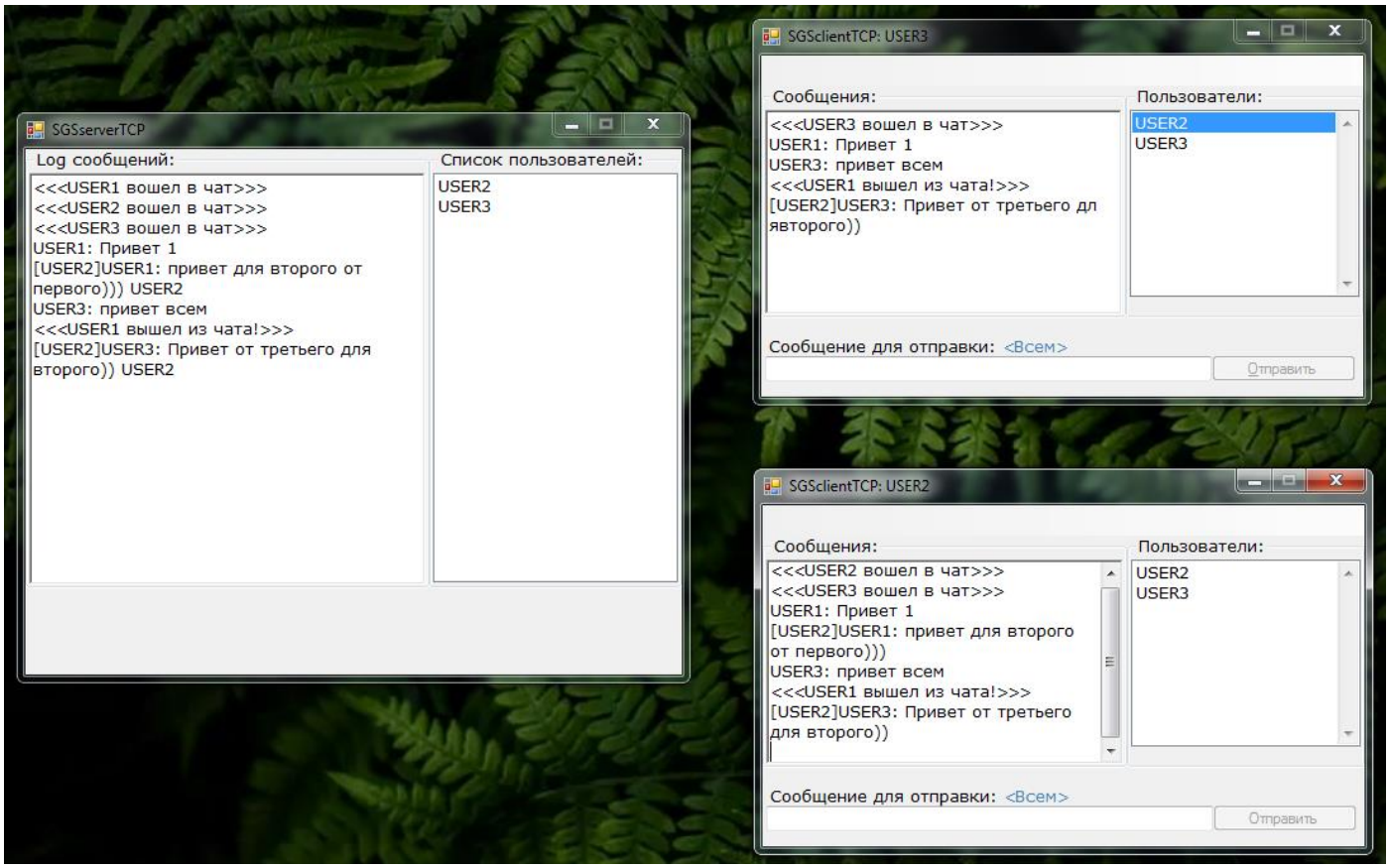
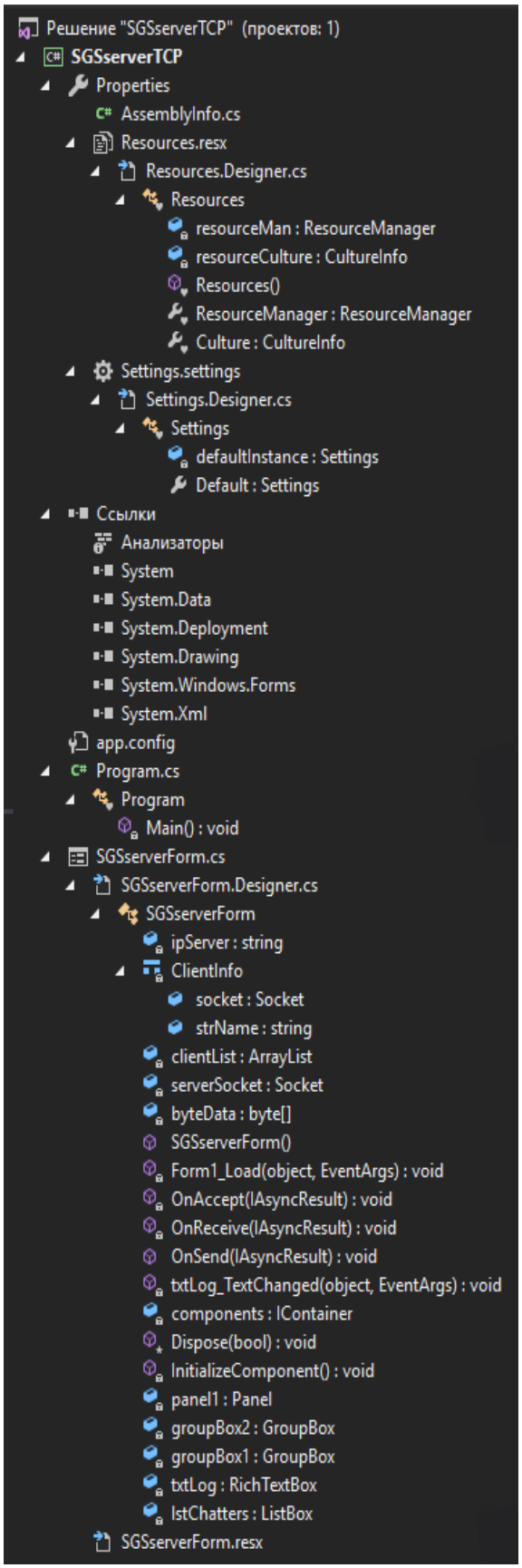


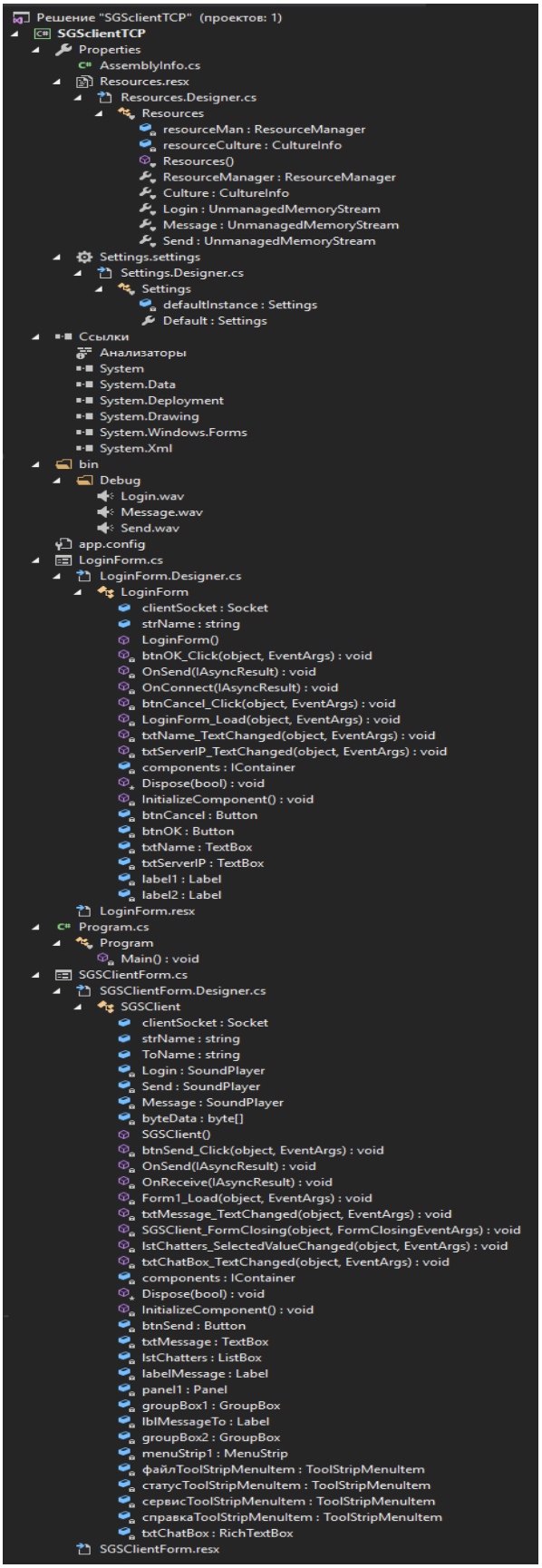
Рис. 4: Чат (Взаимодействие 2 клиентов с сервером)

Стратегия тестирования

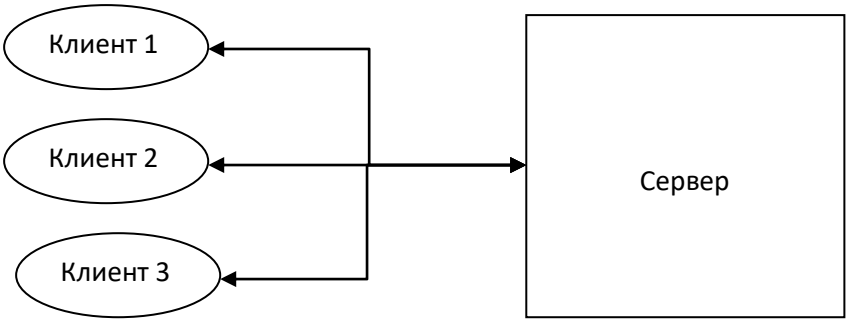
Структура объекта тестирования (Сервер):



и Структура объекта тестирования (Клиент):



Архитектура приложения



Архитектура серверной части (SGSserverTCP)

Файл Program.cs внутри Main(): void

```
namespace Server
   static class Program
      static void Main() – Метод по умолчанию. Точка входа в программу.
```

Файл SGSserverForm.cs внутри SGSserverForm.Designer.cs и SGSserverForm.resx

SGSserverForm.Designer.cs:

```
namespace Server
   partial class SGSserverForm
      protected override void Dispose(bool disposing)
      private void InitializeComponent()
//Инициализация компонентов пользовательского интерфейса сервера----------
      private System.Windows.Forms.Panel panel1;
      private System.Windows.Forms.GroupBox groupBox2;
      private System.Windows.Forms.GroupBox groupBox1;
      private System.Windows.Forms.RichTextBox txtLog;
      private System.Windows.Forms.ListBox lstChatters;
//-------------------------------------------------------------------------------------------------------------
```

SGSserverForm.resx:

```
namespace Server
  enum Command – Внутри описаны команды для взаимодействия между сервером и клиентом
  public partial class SGSserverForm : Form
            struct ClientInfo
            public SGSserverForm()
            private void Form1_Load(object sender, EventArgs e) – Загрузка главного окна (формы) сервера. Не участвует в
блочном тестировании
```

private void OnAccept(IAsyncResult ar) – Метод ожидания входящих подключений от клиентов. Участвует в блочном тестировании.

private void OnReceive(IAsyncResult ar) – Метод обработки сообщения от клиента. Участвует в блочном тестировании.

public void OnSend(IAsyncResult ar) – Метод передачи данных клиенту через tcp-сокет. Участвует в блочном тестировании.

private void txtLog_TextChanged(object sender, EventArgs e) – Метод логирования действий сервера в текстовый блок. Не участвует в блочном тестировании.

class Data - Структура данных, с помощью которого сервер и клиент взаимодействуют друг с другом

public Data() - конструктор по умолчанию

public Data(byte[] data) - Преобразует байты в объект типа данных

public byte[] ToByte() - Преобразует данные структуры в массив байт

public string strName; - Имя, под которым клиент входит в комнату

public string strTo; - Имя пользователя которому отправлено сообщение

public string strMessage; - Текстовое сообщение

public Command cmdCommand; - Тип команды (вход, выход, отправить сообщение, и т.д.)

Архитектура клиентской части (SGSclientTCP)

Файл Program.cs внутри Main(): void

```
namespace SGSclient
  static class Program
    static void Main() – Метод по умолчанию. Точка входа в программу.
```

Файл SGSclientForm.cs внутри SGSclientForm.Designer.cs SGSclientForm.resx

SGSclientForm.Designer.cs :

```
namespace SGSclient
  partial class SGSClient
    private void InitializeComponent()
//Инициализация компонентов пользовательского интерфейса клиента------------------------------
    private System.Windows.Forms.Button btnSend;
    private System.Windows.Forms.TextBox txtMessage;
    private System.Windows.Forms.ListBox lstChatters;
    private System.Windows.Forms.Label labelMessage;
    private System.Windows.Forms.Panel panel1;
    private System.Windows.Forms.GroupBox groupBox1;
    private System.Windows.Forms.Label lblMessageTo;
    private System.Windows.Forms.GroupBox groupBox2;
    private System.Windows.Forms.MenuStrip menuStrip1;
    private System.Windows.Forms.ToolStripMenuItem файлToolStripMenuItem;
    private System.Windows.Forms.ToolStripMenuItem статусToolStripMenuItem;
```

private System.Windows.Forms.ToolStripMenuItem сервисToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem справкаToolStripMenuItem;

private System.Windows.Forms.RichTextBox txtChatBox;

//----------

SGSclientForm.resx :

namespace SGSclient

enum Command - Команды для взаимодействия между сервером и клиентом

public partial class SGSClient : Form

public Socket clientSocket; - Основной сокет клиента

public string strName; - Имя, под которым пользователь подключается к чату

public string ToName; - Имя пользователя которому отправляется сообщение

public SGSClient() – Конструктор по умолчанию.

private void btnSend_Click(object sender, EventArgs e) – Метод обработки события нажатия кнопки «Отправить». Не участвует в блочном тестировании.

private void OnSend(IAsyncResult ar)- Метод отправки данных от клиента серверу через tcp-сокет. Не участвует в блочном тестировании.

private void OnReceive(IAsyncResult ar) – Метод обработки сообщений от сервера. Не участвует в блочном тестировании.

private void Form1_Load(object sender, EventArgs e) – Метод загрузки главного окна (формы) пользовательского интерфейса клиента. Не участвует в блочном тестировании.

private void txtMessage_TextChanged(object sender, EventArgs e) - Проверка на не пустое поле отправляемого сообщения. Не участвует в блочном тестировании.

private void SGSClient_FormClosing(object sender, FormClosingEventArgs e) – Метод закрытия главного окна (формы) пользовательского интерфейса. Не участвует в блочном тестировании.

private void lstChatters_SelectedValueChanged(object sender, EventArgs e) – Метод выбора пользователя для отправки приватного сообщения. Не участвует в блочном тестировании.

private void txtChatBox_TextChanged(object sender, EventArgs e)

class Data - Структура данных, содержащая информацию о передаваемом сообщении.

public Data() - конструктор по умолчанию

public Data(byte[] data) - Преобразует байты в объект типа данных

public byte[] ToByte() - Преобразует данные структуры в массив байт

public string strName; - Имя, под которым клиент входит в комнату

public string strTo; - Имя пользователя которому отправлено сообщение

public string strMessage; - Текстовое сообщение

public Command cmdCommand; - Тип команды (вход, выход, отправить сообщение, и т.д.)

Основные положения процедуры проведения тестирования

1. Тест считается успешным, если в процессе тестирования не было обнаружено ошибок, то есть ожидаемый и фактический результат теста совпадают.
2. Этап тестирования считается завершенным в том случае, когда все тесты текущего этапа успешно пройдены.
3. Переход к следующему этапу тестирования может быть совершен только в том случае, если предыдущий этап тестирования завершен.
4. Процедура проведения тестирования может быть остановлена только в случае ошибки, блокирующей выполнение тестов.
5. Возобновление процедуры проведения тестирования происходит после исправления блокирующей ошибки, которая помешала предыдущему выполнению.

Инструменты тестирования

Для проведения блочного и интеграционного тестирования будут использованы следующие инструменты автоматического тестирования C# Test . Аттестационные и системные тесты будут проводиться в ручном режиме.

Описание методов и обоснование необходимости их применения

Блочное тестирование

Тестирование отдельных методов, используя вспомогательный код. Проведение является необходимым для минимизации ошибок при интеграции.

№ теста	Описание	Входные данные	Ожидаемые результаты
Б1	public bool createSocket(string ip, int port) – функция создания сокета	String ip int port	Возвращает: TRUE- если сокет создался нормально FALSE- если возникла ошибка или исключение
Б2	public configServer _configServer(string path) – метод возвращающий конфигурацию сервера в виде объекта класса configServer.	String path	Возвращает: 1.Экземпляр класса configServer с полями: ip и port, для создания подключения. 2.NULL – если не удалось выполнить чтение файла конфигурации или возникло необрабатываемое исключение
Б3	private void OnAccept(IAsyncResult ar) – Метод создающий новый сокет для подключения клиента	IAsyncResult ar – состояние асинхронной опреации	Успешно: Создается новый сокет для подключенного клиента. Ошибка:

			Создается исключение и выводится сообщение об ошибке.
Б4	private void OnReceive(IAsyncResult ar) – Метод обработки сообщений от клиентов	IAsyncResult ar – состояние асинхронной опреации	Успешно: Метод принимает сообщение клиента, преобразовывает в структуру данных (имя клиента, команда, сообщение). Выполняет переданную команду и отправляет сообщение об успешном выполнении операции. Ошибка: Создается исключение и выводится сообщение об ошибке.
Б5	private void OnSend(IAsyncResult ar) – Метод асинхронной передачи данных через tcp-сокет	IAsyncResult ar – состояние асинхронной операции	Успешно: Возвращает результат выполнения асинхронной операции передачи данных Ошибка: Создается исключение и выводится сообщение об ошибке.
Б6	public byte[] ToByte(String data) – Преобразует текстовую строку в массив байт	String data – строка передаваемого сообщения	Возвращает: Успешно: Структурированный массив байт содержащий информацию о передаваемом сообщении. Ошибка: значение NULL
Б7	private bool SendMessage(String message) – Метод отправки сообщения на сервер	String message – сообщение	Возвращает: TRUE – если сообщение успешно сформировано и отправлено на сервер. FALSE – Если возникло необрабатываемое исключение. Выводится сообщение об исключении.
Б8	public void OnConnect(IAsyncResult ar) – Метод отправки на сервер данных для идентификации нового пользователя	IAsyncResult ar – состояние асинхронной операции	Успешно: Формируется и передается сообщение для сервера, содержащее имя клиента и команду Login, сообщая о новом подключившемся клиенте. Ошибка: Создается обрабатываемое исключение и выводится сообщение об ошибке.

Интеграционное тестирование

Включает в себя тестирование подсистем. Также является важным этапом тестирования для проверки корректности взаимодействия модулей подсистем.

1. Восходящие тестирование подключения подсистемы Client к подсистеме Server.

- Метод Connect() класса LoginForm модуля Client:
  - Создает tcp-сокет.
  - Настраивает сокет для подключения к серверу указав ip-адрес и порт сервера.
  - Открывает подключение к серверу через созданный сокет.
- Метод OnAccept() класса SGSserverForm модуля Server:
  - Создает tcp-сокет для прослушивания подключений клиентов.
  - Принимает подключение клиента.
  - Запускает асинхронный метод обработки сообщений от подключенного клиента.

№ теста	Входные данные	Тип теста	Ожидаемые результаты
И1.1	Модуль Server запущен и ожидает подключений. Пользователь указал правильный ip-адрес сервера на форме авторизации	Позитивный	Подключение модуля Client через созданный tcp-сокет к модулю Server прошло успешно.
И1.2	Модуль Server запущен и ожидает подключений. Пользователь указал неправильный ip-адрес сервера на форме авторизации	Негативный	Модуль Client не смог подключиться к модулю Server через созданный tcp-сокет.
И1.3	Модуль Server не запущен. Пользователь указал правильный ip-адрес сервера на форме авторизации.	Негативный	Модуль Client не смог подключиться к модулю Server через созданный tcp-сокет.

2. Восходящее тестирование передачи сообщения о подключении модуля Client и обработка сообщения о подключении модулем Server.

- Метод OnConnect() класса LoginForm модуля Client.
  - Формирует сообщение для передачи в модуль Server. Сообщение содержит информацию: отправитель сообщения, команда которую должен выполнить Server, текст сообщения.
  - Формирует массив байт с помощью метода ToByte().
  - Выполняет передачу сообщения в модуль Server.
- Метод OnReceive класса SGSServerForm модуля Server.
  - Принимает сообщение от модуля Client.
  - Преобразовывает массив байт полученных от клиента в объект типа данных Data.
  - Получает из объекта данных необходимую для выполнения команду Login.
  - Записывает данные о новом пользователе в структуру данных.
  - Формирует сообщение о новом пользователе.
  - Отправляет сообщение всем подключенным клиентам информацию о новом пользователе.

№ теста	Входные данные	Тип теста	Ожидаемые результаты
И2.1	Модуль Client подключен к модулю Server через tcp-сокет. Пользователь указал уникальное имя на форме авторизации.	Позитивный	Модуль Server добавил нового клиента в список подключенных пользователей и отправил сообщение о новом подключении всем клиентам.
И2.2	Модуль Client подключен к модулю Server через tcp-сокет. Пользователь указал существующее имя на форме авторизации.	Негативный	Модуль Server отклонил команду Login, отправил в модуль Client сообщение об исключении уникального имени и разорвал соединение с модулем Client.
И2.3	Модуль Client подключен к модулю Server через tcp-сокет. Пользователь не указал имя на форме авторизации.	Негативный	Модуль Server отклонил команду Login, отправил в модуль Client сообщение об исключении уникального имени и разорвал соединение с модулем Client.

3. Нисходящее тестирование передачи сообщения от модуля Server в модуль Client.

- Метод OnSend() класса SGSServer модуля Server
  - Передает данные через tcp-сокет в модуль Client.
- Метод OnReceive() класса SGSClient модуля Client:
  - Принимает сообщение от модуля Server.
  - Преобразовывает массив байт полученных от клиента в объект типа данных Data.
  - Получает из объекта данных необходимую для выполнения команду.
  - Выполняет инструкции, соответствующие команде.

№ теста	Входные данные	Тип теста	Ожидаемые результаты
И3.1	Новый пользователь выполнил подключение к чату с уникальным именем User3.	Позитивный	Модуль Client получил команду Login от модуля Server. В окне пользовательского интерфейса была добавлена строка в список подключенных клиентов.
И3.2	Пользователь User3 отправил сообщение «Hello, World!» для всех пользователей.	Позитивный	Модуль Client получил команду Message от модуля Server. В окне пользовательского интерфейса была добавлена строка в текстовое поле сообщений чата

Аттестационное тестирование

Тестирование системы в целом. На данном этапе проверяется соответствие разрабатываемого продукта требованиям заказчика. Поэтому данный этап является обязательным.

№ теста	Описание	Входные данные	Ожидаемые результаты
А1	Авторизация	Ник+IP	Успешный вход в систему под указанным ником
А2	Выход из системы	Красный крестик	Успешный выход из системы
А3.1	Воспроизведение звуков при входе в систему	Вход в систему	Воспроизведение звуков при входе в систему
А3.2	Воспроизведение звуков при отправке сообщения	Отправка сообщения	Воспроизведение звуков при отправке сообщения
А3.3	Воспроизведение звуков при получении сообщения	получение сообщения	Воспроизведение звуков при получении сообщения
А4.1	Отправка текстового сообщения в общий чат	Текст + кнопка ОТПРАВИТЬ	Отправка текстового сообщения в общий чат
А4.2	Отправка текстового сообщения приватно пользователю	Текст + выбор пользователя + кнопка ОТПРАВИТЬ	Отправка текстового сообщения в приватный чат
А5.1	Получение текстового сообщения для общего чата	нет	Появление текстового сообщения в общем чате
А5.2	Получение приватного текстового сообщения	нет	Появление текстового сообщения в приватном чате
А6	Получение списка всех пользователей чата	нет	Отображение списка всех пользователей чата

Специальное тестирование

В качестве специального тестирования будет рассматриваться Системное тестирование — тестирование графического интерфейса.

№ теста	Описание	Входные данные	Ожидаемые результаты
С1.1-П	Растягивание экрана до нестандартных размеров	Окно клиента	Не должно растягиваться
С1.2-П		Окно сервера	Не должно растягиваться
С2.1-П	Проверка инсталляции и конфигурации на разных платформах	Запуск на Windows 10	Запуск без ошибок
С2.2-П		Запуск на Windows 7	Запуск без ошибок
С2.3-П		Запуск на Linux	Запуск без ошибок
С3.1-Н	Отправка нетиповых текстовых	22 383 символов в	Предупреждение о большой длине

	сообщений	тексте сообщения	вводимого текста
С3.2-П		Иероглифы в тексте сообщения	Отправится без предупреждений
С3.3-Н		Пустое сообщение	Невозможно отправить
С4-П	Проверка на предельных объемах нагрузки входного потока	Подключение к чату более 40 пользователей	Работа сервера не остановится

Реализация тестирования

Пример реализации блочного тестирования

Код UnitTest-а для теста № Б1.2:

```
[TestClass]
  public class ServerState
  {
    [TestMethod]
    public void TestCreateSocket()
    {
      string ip = "127.0.0.1";
      int port = 135;

      bool expResult = true;

      SGSserverForm sserverForm = new SGSserverForm();

      bool current = sserverForm.createSocket(ip, port);

      Assert.AreEqual(expResult, current);
    }
  }
```

Журнал тестирования

Тест № + тип	Дата	Тестируемый модуль с описанием	Входные данные	Кол-во запусков	Найденные ошибки	Результат теста
Б1.1-П	03.12.18	public bool createSocket (string ip, int port) – функция создания сокета	ip=‘127.0.0.1’ port=‘11000’	1	0	Пройден успешно
Б1.2-П	03.12.18		ip=‘127.0.0.1’ port=‘135’	1	1	Найдена ошибка (см. отчет №1)
Б1.3-Н	03.12.18		ip=‘R2tdЙgd7’ port=‘11000’	1	0	Пройден успешно
Б1.4-Н	03.12.18		ip=‘R2tdЙsgd7’ port=‘135’	1	1	Найдена ошибка (см. отчет №2)

Б2.1-П	03.12.18	public configServer _configServer(string path) – метод возвращающий конфигурацию сервера в виде объекта класса configServer.	Path = “\\config.ini”	1	0	Пройден успешно
Б2.2-Н	03.12.18		Path = “config.ini”	1	1	Найдена ошибка (см. отчет №3)
Б3.1-П	03.12.18	private void OnAccept(IAsyncResult ar) – Метод создающий новый сокет для подключения клиента	Подключение Client к Server.	1	0	Пройден успешно
Б3.2-П	03.12.18		Подключение Client к Server.	1	0	Пройден успешно
Б4.1-П	03.12.18	private void OnReceive(IAsyncResult ar) – Метод обработки сообщений от клиентов	Сообщение от Client: Name= user1 Message=”Hello” Command=Message	1	0	Пройден успешно
Б4.2-П	03.12.18		Сообщение от Client: Name= user1 Message=”” Command=Logout	1	0	Пройден успешно
Б5.1-П	03.12.18	private void OnSend(IAsyncResult ar) – Метод асинхронной передачи данных через tcp-сокет	Отправка данных от Server на Client	1	0	Пройден успешно
Б5.2-П	03.12.18		Отправка данных от Server на Client	1	0	Пройден успешно
Б6.1-П	03.12.18	public byte[] ToByte(String data) – Преобразует текстовую строку в массив байт	Строка «Привет, Мир»	1	0	Пройден успешно
Б6.2-П	03.12.18		Строка «Пользователь1»	1	0	Пройден успешно
Б7.1-П	03.12.18	private bool SendMessage(String message) – Метод отправки	Message=”Привет, Мир!”	1	0	Пройден успешно
Б7.2-П	03.12.18		Message=””	1	0	Пройден успешно

Б7.3-П	03.12.18	сообщения на сервер	Message=”000”	1	0	Пройден успешно
Б8.1-П	03.12.18	public void OnConnect(IAsync Result ar) – Метод отправки на сервер данных для идентификации нового пользователя	username=”user1”	1	0	Пройден успешно
Б8.2-П	03.12.18		Username=”user”	1	1	Найдена ошибка (см. отчет №4)
А1.1-П	04.12.18	Авторизация	Ник = ‘USER1’ IP = ‘127.0.0.1’	1	0	Пройден успешно
А1.2-Н	04.12.18		Ник = ‘USER2’ IP = ‘rYue73’	1	0	Пройден успешно
А1.3-Н	04.12.18		Ник = ‘%frtgyh6’ IP = ‘255/0/255/0’	1	0	Пройден успешно
А2.1-П	04.12.18	Выход из системы	Нажатие на красный крестик	1	0	Пройден успешно
А3.1-П	04.12.18	Воспроизведение звуков при входе в систему	Пользователь1 авторизовался, как: Ник = ‘USER1’ IP = ‘127.0.0.1’	1	0	Пройден успешно
А3.2-П	04.12.18	Воспроизведение звуков при отправке сообщения	Пользователь1 ввел сообщение = ’Hello’ + кнопка ОТПРАВИТЬ	1	0	Пройден успешно
А3.3-П	04.12.18	Воспроизведение звуков при получении сообщения	Пользователь2 получил сообщение= ’Hello’ от пользователя1	1	0	Пройден успешно
А4.1.1-П	04.12.18	Отправка текстового сообщения в общий чат	Текст=’Hello’ + кнопка ОТПРАВИТЬ	1	0	Пройден успешно
А4.1.2-Н	04.12.18		Текст=NULL + кнопка ОТПРАВИТЬ	1	0	Пройден успешно
А4.1.3-Н	04.12.18		Текст=’jtyue456….56 yhtr5’ + кнопка ОТПРАВИТЬ	1	0	Пройден успешно
А4.2.1-П	04.12.18	Отправка текстового сообщения приватно	Текст=’Hello’ + выбор пользователя в сети + кнопка	1	0	Пройден успешно

		пользователю	ОТПРАВИТЬ			
А4.2.2-Н	04.12.18		Текст=NULL + выбор пользователя в сети + кнопка ОТПРАВИТЬ	1	0	Пройден успешно
А4.2.3-Н	04.12.18		Текст=’jtyue456….56 yhtr5’ + выбор пользователя в сети + кнопка ОТПРАВИТЬ	1	0	Пройден успешно
А5.1-П	04.12.18	Получение текстового сообщения в общем чате	Пользователь1 Текст=’Hello’ + кнопка ОТПРАВИТЬ	1	0	Пройден успешно
А5.2-П	04.12.18	Получение приватного текстового сообщения	Пользователь1 Текст=’Hello’ + выбор пользователя2 в сети + кнопка ОТПРАВИТЬ	1	0	Пройден успешно
А6-О	04.12.18	Получение списка всех пользователей чата	Отображение списка всех пользователей в чате	1	0	Пройден успешно
И1.1-П	05.12.18	Подключение модуля Client к модулю Server	Модуль Server запущен. Пользователь указал ip-адрес: 127.0.0.1	1	0	Пройден успешно
И1.2-Н	05.12.18		Модуль Server запущен. Пользователь указал ip-адрес: 192.168.0.111	1	0	Пройден успешно
И1.3-Н	05.12.18		Модуль Server не запущен. Пользователь указал ip-адрес: 127.0.0.1	1	0	Пройден успешно
И2.1-П	05.12.18	Передача сообщения об авторизации	Подключение клиента 1: Пользователь	1	0	Пройден успешно

		пользователя и обработка сообщения в модуле Server	указал имя: «User1»			
И2.2-П	05.12.18		Подключение клиента 2: Пользователь указал имя: «User1»	1	0	Пройден успешно
И2.3-Н	05.12.18		Подключение клиента 1: Пользователь указал имя: «»	1	0	Пройден успешно
И3.1-П	05.12.18	Передача сообщения от модуля Server в модуль Client и обработка сообщения	Пользователь подключился к чату с именем «User3»	1	0	Пройден успешно
И3.2-Н	05.12.18		Пользователь User3 отправил сообщение «Hello, World!»	1	0	Пройден успешно
С1.1-П	06.12.18	Растягивание экрана до нестандартных размеров	Окно клиента	1	0	Пройден успешно
С1.2-П	06.12.18		Окно сервера	1	0	Пройден успешно
С2.1-П	06.12.18	Проверка инсталляции и конфигурации на разных платформах	Запуск на Windows 10	1	0	Пройден успешно
С2.2-П	06.12.18		Запуск на Windows 7	1	0	Пройден успешно
С2.3-П	06.12.18		Запуск на Linux	1	0	Пройден успешно
С3.1-Н	06.12.18	Отправка нетиповых текстовых сообщений	22 383 символов в тексте сообщения	1	0	Пройден успешно
С3.2-П	06.12.18		Попытка отправки текста сообщения=’临摹盗袭图画盗袭’	1	0	Пройден успешно
С3.3-Н	06.12.18		Попытка отправки текста сообщения=NULL	1	0	Пройден успешно
С4-П	06.12.18	Проверка на предельных объемах нагрузки входного потока	Подключение к чату более 40 пользователей	1	0	Пройден успешно

Журнал найденных ошибок

Номер отчета об ошибке	Дата составления отчета	Номер теста	Входные данные	Ожидаемый результат	Фактический результат
№ 1	03.12.18	Б1.2 - П	ip='127.0.0.1' port='135'	TRUE - сокет создался без ошибок	Вылезло необрабатываемое исключение, т.к. пытаемся использовать занятый порт
№ 2	03.12.18	Б1.4-Н	ip='R2tdЙsgd7' port='135'	TRUE - сокет создался без ошибок	Вылезло необрабатываемое исключение, т.к. пытаемся использовать занятый порт
№ 3	03.12.18	Б2.2-П	Path = "config.ini"	Экземпляр класса configServer с полями	Исключение, что не удалось найти указанный файл
№ 4	03.12.18	Б8.2-П	Username = "user"	Формируется и передается сообщение для сервера, содержащее имя клиента и команду Login, сообщая о новом подключившемся клиенте.	Вылезло необрабатываемое исключение, т.к. пользователь с таким именем уже существует

Отчеты об ошибках

Отчет №1

Краткое описание: public bool createSocket(string ip, int port) – функция создания сокета
Входные данные: ip='127.0.0.1' port='135'
Ожидаемый результат: Создастся сокет без ошибок
Фактический результат: Вылезло необрабатываемое исключение, т.к. пытаемся использовать занятый порт

Отчет №2

Краткое описание: public bool createSocket(string ip, int port) – функция создания сокета
Входные данные: ip='R2tdЙsgd7' port='135'
Ожидаемый результат: Создастся сокет без ошибок
Фактический результат: Вылезло необрабатываемое исключение, т.к. пытаемся использовать занятый порт

Отчет №3

Краткое описание: public configServer _configServer(string path) – метод возвращающий конфигурацию сервера в виде объекта класса configServer.
Входные данные: Path = "config.ini"
Ожидаемый результат: Создастся сокет без ошибок
Фактический результат: Вылезло необрабатываемое исключение, т.к. пользователь с таким именем уже существует

Отчет №4

Краткое описание: public void OnConnect(IAsyncResult ar) – Метод отправки на сервер данных для идентификации нового пользователя
Входные данные: Username="user"
Ожидаемый результат: Создастся сокет без ошибок
Фактический результат: Вылезло необрабатываемое исключение, т.к. пользователь с таким именем уже существует

Метрики

Всего запланировано тестов: 44
Выполнено: 44
Успешно: 40
Провалено: 4
Исправлено: 0

Покрытие исполняемого кода тестами

Расчет тестового покрытия относительно исполняемого кода ПО проводится по формуле:

X = A/B*100%

где:

- X - тестовое покрытие;
- A - количество строк кода, покрытых тестами;
- B - общее количество строк кода;

Тестовое покрытие = X = 628/804 * 100% = 78 %

Результаты

Программа успешно прошла тестирование. Было найдено 4 ошибки. Проведенное тестирование позволяет надеяться на невысокую вероятность возникновения новых серьезных ошибок в работе текущей версии приложения.