

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С. П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

Дополнительное задание по курсу “Операционные Системы”

Создание приложения мессенджера для личного кабинета Самарского
университета на основе HTTP запросов.

Выполнил:
Аброськин А.М.
группа 6304-090301D

Проверил:
Востокин С.В.

Самара 2022

1. Функциональность приложения

- Вход при помощи имени пользователя и пароля.
- Просмотр сообщений в выбранном чате.
- Отправка сообщений в выбранный чат.

2. Средства программирования

Приложение написано на языке C# и использует следующие библиотеки:

- .NET Framework 4.7.2
- Json.NET Newtonsoft

3. Структура программы

Работа программы основана на классе System.Net.HttpWebRequest из библиотеки .NET Framework 4.7.2. Также использован класс JsonConvert пакета Json.NET Newtonsoft для преобразования объектов языка C# в объекты JSON и наоборот. Основные методы реализуют POST и GET запросы.

POST запрос:

```
public HttpStatusCode Post(string uri, object data, string contentType) {
    byte[] dataBytes =
        Encoding.UTF8.GetBytes(JsonConvert.SerializeObject(data));
    HttpRequest request = (HttpRequest)WebRequest.Create(uri);
    request.ContentLength = dataBytes.Length;
    request.ContentType = contentType;
    request.Headers.Set("Cookie", cookie);
    request.Method = "POST";
    using (Stream requestBody = request.GetRequestStream())
    {
        requestBody.Write(dataBytes, 0, dataBytes.Length);
    }

    using (HttpWebResponse response =
        (HttpWebResponse)request.GetResponse())
    using (Stream stream = response.GetResponseStream())
    using (StreamReader reader = new StreamReader(stream))
    {
        cookie = response.GetResponseHeader("Set-Cookie");
        return response.StatusCode;
    }
}
```

GET запрос:

```
public string GetRequestWithCookie(string uri)
{
    HttpRequest request = (HttpRequest)WebRequest.Create(uri);
```

```

request.Method = "GET";
request.Headers.Set("Cookie", cookie);
using (HttpWebResponse response =
    (HttpWebResponse)request.GetResponse())
using (Stream stream = response.GetResponseStream())
using (StreamReader reader = new StreamReader(stream))
{
    cookie = response.GetResponseHeader("Set-Cookie");
    return reader.ReadToEnd();
}
}

```

Для входа в личный кабинет используется POST запрос с именем пользователя и паролем по ссылке <https://cabinet.ssau.ru/login>. Из ответа сервера запоминаем куки, которые используются во всех дальнейших запросах. Логин и пароль являются JSON объектом со структурой:

```
{ login : "имя пользователя" , password : "пароль" }
```

Для получения списка чатов используется GET запрос по ссылке <https://cabinet.ssau.ru/api/chats>.

Для получения сообщений в чате используется GET запрос по ссылке https://cabinet.ssau.ru/api/chats/CHAT_ID, где CHAT_ID это число-индикатор чата.

Для отправки сообщений используется POST запрос по ссылке https://cabinet.ssau.ru/api/chats/CHAT_ID, где CHAT_ID это число-индикатор чата. Сообщение является объектом JSON структура которого выглядит так:

```
{ message : "сообщение" }
```

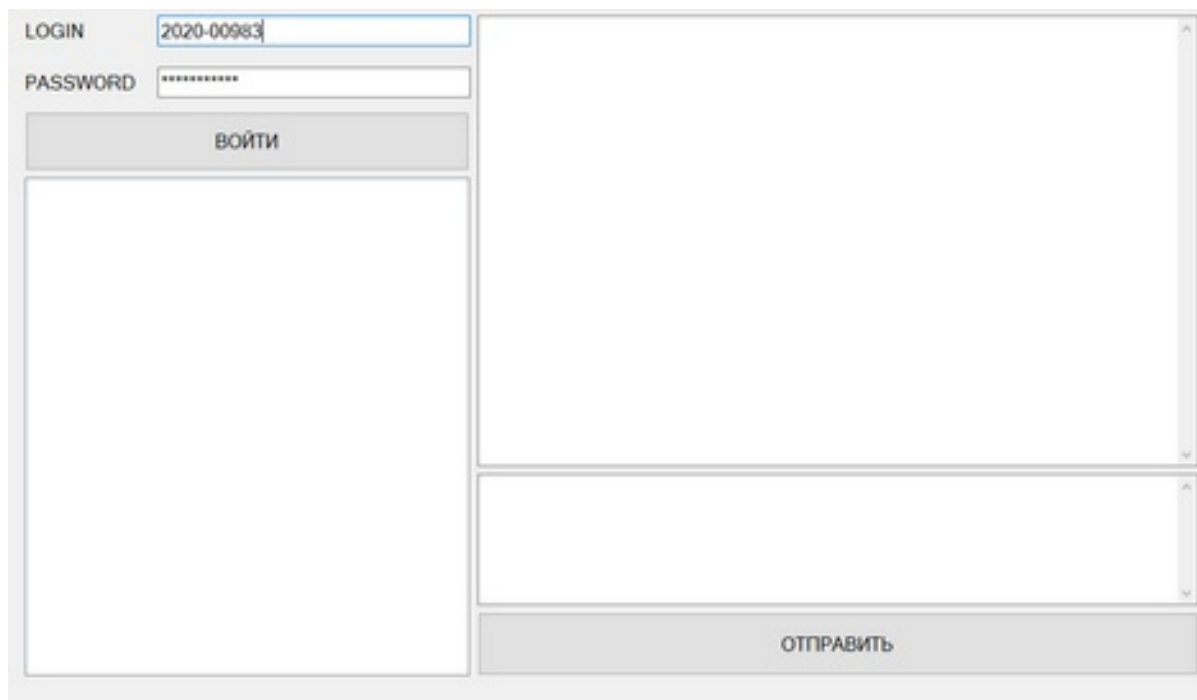
Классы программы:

- Form1 – интерфейс приложения.
- Site – методы и поля для http запросов.
- Chats – Аналог JSON структуры массива чатов.
- Chat – Аналог JSON структуры массива сообщений в чате.

Код всех классов можно найти в приложении.

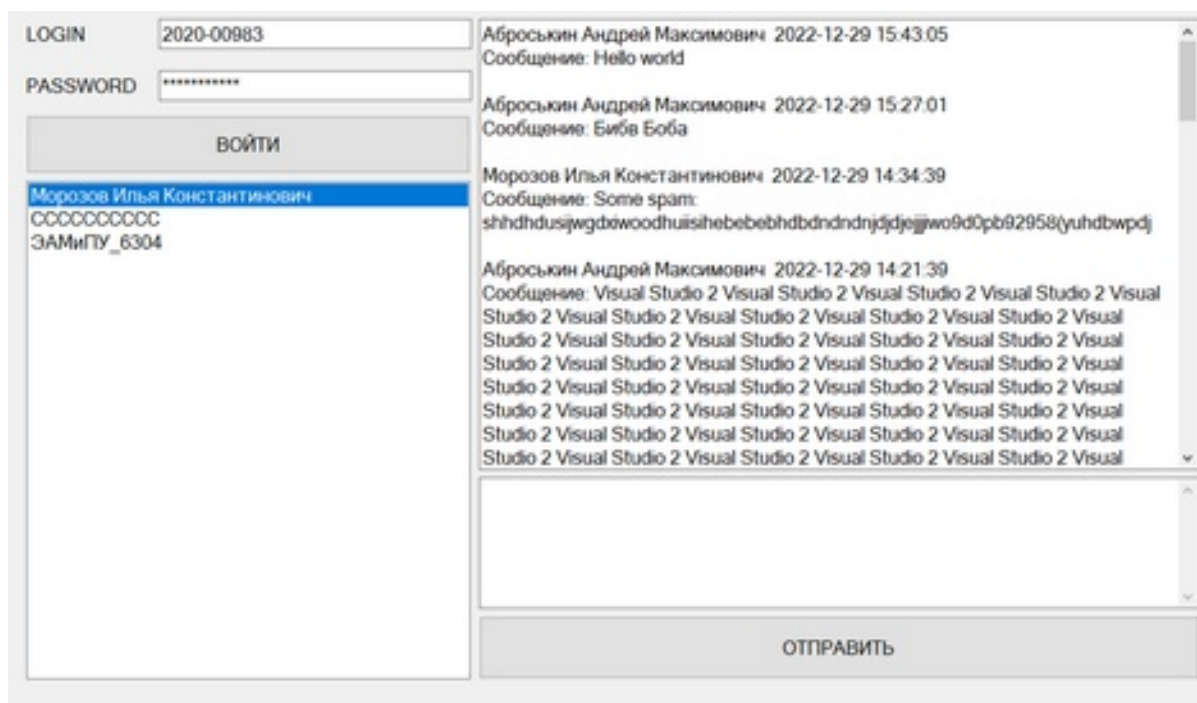
4. Демонстрация работы

Сначала от пользователя требуется ввести его логин и пароль от личного кабинета и нажать кнопку Войти.



The screenshot shows a login interface. On the left, there are two input fields: 'LOGIN' containing the text '2020-00983' and 'PASSWORD' containing a series of asterisks. Below these fields is a button labeled 'ВОЙТИ'. To the right of the login fields is a large, empty rectangular area, likely intended for displaying chat messages. At the bottom right of the interface is a button labeled 'ОТПРАВИТЬ'.

После этого требуется выбрать чат, нажав на его название, и сообщения отобразятся в правом окне.



The screenshot shows the chat interface after login. On the left, the 'LOGIN' and 'PASSWORD' fields are still visible, along with the 'ВОЙТИ' button. Below them is a list of chat titles. The first title is highlighted in blue: 'Морозов Илья Константинович' with a sub-label 'ЗАМПУ_6304'. Other titles include 'Аброськин Андрей Максимович' and 'Морозов Илья Константинович'. On the right, a message history is displayed, showing messages from 'Аброськин Андрей Максимович' and 'Морозов Илья Константинович' with their respective timestamps and message content. At the bottom right, there is an empty text input field and the 'ОТПРАВИТЬ' button.

Для отправки сообщения в данный чат, требуется ввести текст в окно под чатом и нажать кнопку отправить.

Все

☆ ☆

Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2
Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2
Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2
Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2
Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2
Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2
Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2
Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2 Visual Studio 2

14:21 ✓

Морозов Илья Константинович

☆ ☆

Some spam: shhdhdusjwgdxiwoodhuisihebebehdndndndj(djjejjwo9d0pb92958(yuhdbwpcj)
14:34

Все

☆ ☆

Еще Софа

15:27 ✓

Все

☆ ☆

Hello world

15:43 ✓

Все

☆ ☆ 1

Мой софусе

19:59 ✓

Приложение

Класс Chat

```
public class Chat
{
    public int id { get; set; }
    public int chatId { get; set; }
    public string text { get; set; }
    public Author author { get; set; }
    public string createdAt { get; set; }
    public bool isRead { get; set; }
    public object isPinned { get; set; }
    public bool isImportant { get; set; }
    public List<object> links { get; set; }

    public class Author
    {
        public int id { get; set; }
        public string name { get; set; }
    }
}
```

Класс Chats

```
public class Chats
{
    public int id { get; set; }
    public string name { get; set; }
    public string type { get; set; }
    public string avatar { get; set; }
    public List<ChatMember> chatMembers { get; set; }
    public LastMessage lastMessage { get; set; }
    public string createdAt { get; set; }
    public string lastActivity { get; set; }
    public Creator creator { get; set; }
    public bool isArchive { get; set; }
    public bool isDisableNotification { get; set; }
    public int isImportant { get; set; }
    public int unreadMessagesCount { get; set; }

    public class Author
    {
        public int id { get; set; }
        public string name { get; set; }
    }

    public class ChatMember
    {
        public int id { get; set; }
        public string name { get; set; }
    }

    public class Creator
    {
        public int id { get; set; }
        public bool isTeacher { get; set; }
    }

    public class LastMessage
    {
        public int id { get; set; }
        public int chatId { get; set; }
    }
}
```

```

    public string text { get; set; }
    public Author author { get; set; }
    public string createdAt { get; set; }
    public bool isRead { get; set; }
    public object isPinned { get; set; }
    public bool isImportant { get; set; }
    public List<object> links { get; set; }
}
}

```

Класс Site

```

internal class Site
{
    string cookie;
    string loginURL = "https://cabinet.ssau.ru/login";
    string chatsURL = "https://cabinet.ssau.ru/api/chats";

    public Site2()
    {
        string cookie = GetRequestCookey(loginURL);
    }

    public bool sendMessage(string chatId, string mess)
    {
        return (Post(chatsURL + "/" + chatId, new { message = mess }, "application/json") == HttpStatusCode.OK);
    }

    public string[] getChat(string chatId)
    {
        List<Chat> chat = JsonConvert.DeserializeObject<List<Chat>>(GetRequestWithCookie(chatsURL + "/" +
chatId + "?limit=100&offset=0"));
        string[] result = new string[chat.Count];
        for (int i = 0; i < chat.Count; i++)
        {
            result[i] = chat[i].author.name + " " + chat[i].createdAt + "\r\n" + "Сообщение: " + chat[i].text + "\r\n\r\n";
        }
        return result;
    }

    public string[,] getChats()
    {
        List<Chats> chats = JsonConvert.DeserializeObject<List<Chats>>(GetRequestWithCookie(chatsURL));
        string[,] result = new string[chats.Count,2];
        for (int i = 0; i < result.GetLength(0); i++)
        {
            result[i,0] = chats[i].name;
            result[i,1] = chats[i].id.ToString();
        }
        return result;
    }

    public bool Login(string login, string password)
    {
        return (Post(loginURL, new { name = login, password = password }, "application/json") ==
HttpStatusCode.OK);
    }

    public HttpStatusCode Post(string uri, object data, string contentType)
    {
        byte[] dataBytes = Encoding.UTF8.GetBytes(JsonConvert.SerializeObject(data));
        HttpRequest request = (HttpRequest)WebRequest.Create(uri);
        request.AutomaticDecompression = DecompressionMethods.GZip | DecompressionMethods.Deflate;
    }
}

```



```

request.ContentLength = dataBytes.Length;
request.ContentType = contentType;
request.Headers.Set("Cookie", cookie);
request.Method = "POST";
using (Stream requestBody = request.GetRequestStream())
{
    requestBody.Write(dataBytes, 0, dataBytes.Length);
}

using (HttpWebResponse response = (HttpWebResponse)request.GetResponse())
using (Stream stream = response.GetResponseStream())
using (StreamReader reader = new StreamReader(stream))
{
    cookie = response.GetResponseHeader("Set-Cookie");
    return response.StatusCode;
}
}

public string GetRequestWithCookie(string uri)
{
    HttpRequest request = (HttpRequest)WebRequest.Create(uri);
    request.AutomaticDecompression = DecompressionMethods.GZip | DecompressionMethods.Deflate;
    request.Method = "GET";
    request.Headers.Set("Cookie", cookie);
    using (HttpWebResponse response = (HttpWebResponse)request.GetResponse())
    using (Stream stream = response.GetResponseStream())
    using (StreamReader reader = new StreamReader(stream))
    {
        cookie = response.GetResponseHeader("Set-Cookie");
        return reader.ReadToEnd();
    }
}

public string GetRequestCookey(string uri)
{
    HttpRequest request = (HttpRequest)WebRequest.Create(uri);
    request.AutomaticDecompression = DecompressionMethods.GZip | DecompressionMethods.Deflate;
    request.Method = "GET";
    using (HttpWebResponse response = (HttpWebResponse)request.GetResponse())
    using (Stream stream = response.GetResponseStream())
    using (StreamReader reader = new StreamReader(stream))
    {
        return response.GetResponseHeader("Set-Cookie");
    }
}
}

```

Класс Form1

```

public partial class Form1 : Form
{
    string log = "2020-00983";
    Site2 site;
    string[,] chats;
    public Form1()
    {
        InitializeComponent();
        textBoxLogin.Text = log;
    }

    private void button1_Click(object sender, EventArgs e)

```

```
{
    site = new Site2();
    if (site.Login(textBoxLogin.Text, textBoxPass.Text))
    {
        chats = site.getChats();
        listBoxChats.Items.Clear();
        for (int i = 0; i < chats.GetLength(0); i++)
        {
            listBoxChats.Items.Add(chats[i,0]);
        }
    }
}

private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    textBox1.Text = "";
    string[] chat = site.getChat(chats[listBoxChats.SelectedIndex, 1]);
    foreach (string message in chat)
    {
        textBox1.Text += message;
    }
}

private void button2_Click(object sender, EventArgs e)
{
    if (site.sendMessage(chats[listBoxChats.SelectedIndex, 1], textBoxMessage.Text))
    {
        textBoxMessage.Text = "";
        listBox1_SelectedIndexChanged(null, null);
    }
}
}
```