

# Лабораторная работа № 7

## Программирование приложений с использованием WebSocket

### 1. Теоретические сведения

Протокол WebSocket (стандарт RFC 6455) предназначен для решения любых задач и снятия ограничений обмена данными между браузером и сервером.

Он позволяет пересылать любые данные, на любой домен, безопасно и почти без лишнего сетевого трафика.

Пример браузерного кода

Для открытия соединения достаточно создать объект WebSocket, указав в нём специальный протокол ws.:

```
var socket = new WebSocket("ws://javascript.ru/ws");
```

У объекта socket есть четыре коллбэка: один при получении данных и три — при изменениях в состоянии соединения:

```
socket.onopen = function() {  
    alert("Соединение установлено.");  
};
```

```
socket.onclose = function(event) {  
    if (event.wasClean) {  
        alert('Соединение закрыто чисто');  
    } else {  
        alert('Обрыв соединения'); // например, "убит" процесс сервера  
    }  
    alert('Код: ' + event.code + ' причина: ' + event.reason);  
};
```

```
socket.onmessage = function(event) {  
    alert("Получены данные " + event.data);  
};
```

```
socket.onerror = function(error) {  
    alert("Ошибка " + error.message);  
};
```

Для отправки данных используется метод `socket.send(data)`. Пересылать можно любые данные.

Например, строку:

```
socket.send("Привет");
```

...Или файл, выбранный в форме:

```
socket.send(form.elements[0].file);
```

Просто, не правда ли? Выбираем, что переслать, и `socket.send()`.

Для того, чтобы коммуникация была успешной, сервер должен поддерживать протокол WebSocket.

Чтобы лучше понимать происходящее — посмотрим, как он устроен.

### Установка WebSocket-соединения

Протокол WebSocket работает *над* HTTP.

Это означает, что при соединении браузер отправляет специальные заголовки, спрашивая: «поддерживает ли сервер WebSocket?».

Если сервер в ответных заголовках отвечает «да, поддерживаю», то дальше HTTP прекращается и общение идёт на специальном протоколе WebSocket, который уже не имеет с HTTP ничего общего.

### Установка соединения

Пример запроса от браузера при создании нового объекта `new WebSocket("ws://server.example.com/chat")`:

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Origin: http://javascript.ru
Sec-WebSocket-Key: Iv8io/9s+IYFgZWcXczP8Q==
Sec-WebSocket-Version: 13
```

Описания заголовков:

GET, Host

Стандартные HTTP-заголовки из URL запроса

Upgrade, Connection

Указывают, что браузер хочет перейти на websocket.

Origin

Протокол, домен и порт, откуда отправлен запрос.

Sec-WebSocket-Key

Случайный ключ, который генерируется браузером: 16 байт в кодировке Base64.

Sec-WebSocket-Version

Версия протокола. Текущая версия: 13.

Все заголовки, кроме GET и Host, браузер генерирует сам, без возможности вмешательства JavaScript.

Такой XMLHttpRequest создать нельзя

Создать подобный XMLHttpRequest-запрос (подделать WebSocket) невозможно, по одной простой причине: указанные выше заголовки запрещены к установке методом `setRequestHeader`.

Сервер может проанализировать эти заголовки и решить, разрешает ли он WebSocket с данного домена Origin.

Ответ сервера, если он понимает и разрешает WebSocket-подключение:

HTTP/1.1 101 Switching Protocols

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Accept: hsBlbuDTkk24srzEOTBUIZAIC2g=

Здесь строка Sec-WebSocket-Accept представляет собой перекодированный по специальному алгоритму ключ Sec-WebSocket-Key. Браузер использует её для проверки, что ответ предназначен именно ему.

Затем данные передаются по специальному протоколу, структура которого («фреймы») изложена далее. И это уже совсем не HTTP.

### Расширения и подпротоколы

Также возможны дополнительные заголовки Sec-WebSocket-Extensions и Sec-WebSocket-Protocol, описывающие расширения и подпротоколы (subprotocol), которые поддерживает данный клиент.

Посмотрим разницу между ними на двух примерах:

- Заголовок Sec-WebSocket-Extensions: deflate-frame означает, что браузер поддерживает модификацию протокола, обеспечивающую сжатие данных.
- Это говорит не о самих данных, а об улучшении способа их передачи. Браузер сам формирует этот заголовок.
- Заголовок Sec-WebSocket-Protocol: soap, wamp говорит о том, что по WebSocket браузер собирается передавать не просто какие-то данные, а данные в протоколах SOAP или WAMP («The WebSocket Application Messaging Protocol»). Стандартные подпротоколы регистрируются в специальном каталоге IANA.
- Этот заголовок браузер поставит, если указать второй необязательный параметр WebSocket:
- ```
var socket = new WebSocket("ws://javascript.ru/ws", ["soap", "wamp"]);
```

При наличии таких заголовков сервер может выбрать расширения и подпротоколы, которые он поддерживает, и ответить с ними.

Например, запрос:

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Origin: http://javascript.ru
Sec-WebSocket-Key: Iv8io/9s+IYFgZWcXczP8Q==
Sec-WebSocket-Version: 13
Sec-WebSocket-Extensions: deflate-frame
Sec-WebSocket-Protocol: soap, wamp
```

Ответ

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: hsBlbuDTkk24srzEOTBUIZAIC2g=
Sec-WebSocket-Extensions: deflate-frame
Sec-WebSocket-Protocol: soap
```

В ответе выше сервер указывает, что поддерживает расширение deflate-frame, а из запрошенных подпротоколов — только SOAP.

## WSS

Соединение WebSocket можно открывать как WS:// или как WSS://. Протокол WSS представляет собой WebSocket над HTTPS.

Кроме большей безопасности, у WSS есть важное преимущество перед обычным WS — большая вероятность соединения.

Дело в том, что HTTPS шифрует трафик от клиента к серверу, а HTTP — нет.

Если между клиентом и сервером есть прокси, то в случае с HTTP все WebSocket-заголовки и данные передаются через него. Прокси имеет к ним доступ, ведь они никак не шифруются, и может расценить происходящее как нарушение протокола HTTP, обрезать заголовки или оборвать передачу.

А в случае с WSS весь трафик сразу кодируется и через прокси проходит уже в закодированном виде. Поэтому заголовки гарантированно пройдут, и общая вероятность соединения через WSS выше, чем через WS.

## Пример программы на JavaEE 7

```
package com.byteslounge.websockets;

import java.io.IOException;
import java.util.Collections;
import java.util.HashSet;
import java.util.Set;

import javax.websocket.OnClose;
import javax.websocket.OnError;
import javax.websocket.OnMessage;
import javax.websocket.OnOpen;
import javax.websocket.Session;
import javax.websocket.server.ServerEndpoint;

//http://www.hascode.com/2013/08/creating-a-chat-application-using-java-ee-7-
//websockets-and-glassfish-4/

@ServerEndpoint("/websockmulty")
public class WebSocketMulty {

    //private static Set<Session> clients = Collections.synchronizedSet(new
    HashSet<Session>());
    private static Set<Session> clients = new HashSet<Session>();

    @OnMessage
```

```

public void onMessage(String message, Session session) {

    synchronized(clients){
        // Iterate over the connected sessions
        // and broadcast the received message

        for(Session client : clients){
            if (!client.equals(session)){

                try {
                    client.getBasicRemote().sendText(message);
                } catch(IOException ex) {
                    System.out.println("client error");
                    clients.remove(client);
                }
            }
        }
    }

}

private static int nextId;
@OnOpen
public void onOpen (Session session) {
    // Add session to the connected sessions set
    session.getUserProperties().put("username", "user_"+nextId++);

    synchronized(clients) {
        clients.add(session);
    }

}

@OnClose
public void onClose (Session session) {
    // Remove session from the connected sessions set
    synchronized(clients) {
        clients.remove(session);
    }

    System.out.println("client disconnected: " +
session.getUserProperties().get("username"));
}

@OnError
public void onError(Session session, Throwable thr) {
    System.out.println("error socket, maybe someone disconnected");
}

}

```

## Клиентская часть

```

<!DOCTYPE html>
<html>

```

```

<head>
<title>Testing websockets</title>
</head>
<body>
  <div>
    <input id="inputmessage" type="text" />
  </div>
  <div>
    <input type="submit" value="Broadcast message" onclick="send()" />
  </div>
  <div id="messages"></div>
  <script type="text/javascript">
    var webSocket = new
WebSocket('ws://localhost:8080/example-websockets/websockmulty');

    webSocket.onerror = function(event) {
      onError(event)
    };

    webSocket.onopen = function(event) {
      onOpen(event)
    };

    webSocket.onmessage = function(event) {
      onMessage(event)
    };

    function onMessage(event) {
      document.getElementById('messages').innerHTML
        += '<br />Received message: ' + event.data;
    }

    function onOpen(event) {
      document.getElementById('messages').innerHTML
        = 'Connection established';
    }

    function onError(event) {
      alert(event.data);
    }

    function send() {
      var txt = document.getElementById('inputmessage').value;
      webSocket.send(txt);
      return false;
    }
  </script>
</body>
</html>

```

## 2. Задание на лабораторную работу.

Разработать интерактивное веб-приложение с использованием технологии WebSocket. (рекомендуется использовать .Net, Java или Python, можно использовать другие языки программирования по согласованию с преподавателем)

### Отчёт должен содержать:

- Краткое описание алгоритма
- Листинги серверного кода, отправки сообщений клиенту
- Листинги клиентского кода с обработкой событий с сервера
- Верификация

\* Все задания должны выполняться в реальном времени, без перезагрузки страницы браузера. И без использования пуллинга (опрашивания изменения на сервере по таймеру).

### Варианты заданий

|   |                                                                                                                                                                                                                                      |
|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Общий чат                                                                                                                                                                                                                            |
| 2 | Погода, сервер генерирует изменения погоды - клиенты входят на сайт и получают обновление прогноза погоды в реальном времени.                                                                                                        |
| 3 | Курсы валют, сервер генерирует изменения курсов валют - клиенты входят на сайт и получают обновленные курсы валют в реальном времени.                                                                                                |
| 4 | Новости, сервер генерирует случайные новости. Клиенты вошедшие на сайт, в реальном времени получают извещение.                                                                                                                       |
| 5 | Чат с комнатами. При входе на сайт отображается чат с доступными комнатами (комната существует до тех пор, пока в ней есть хотя бы один человек)                                                                                     |
| 6 | Обмен кодом (с подсветкой) - по антологии с публичным чатом, но ключевые слова должны быть подсвечены.                                                                                                                               |
| 7 | Чат, в котором общение происходит при помощи обмена текстовыми сообщениями и картинками (смайликами)                                                                                                                                 |
| 8 | Bro-чат (приложение из сериала "Кремневая долина", только веб-версия)<br><br><a href="https://play.google.com/store/apps/details?id=com.silicon.valley.bro">https://play.google.com/store/apps/details?id=com.silicon.valley.bro</a> |
| 9 | Чат, в котором общение происходит при помощи обмена смайлами (Смайлик-чат)                                                                                                                                                           |



|    |                                                                                                                                                                                                       |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10 | Public stream. Streamer заходит на главную страницу, вводит логин и начинает писать. Подписчики вводят url стрима. Изменения информации на стриме происходит в реальном времени.                      |
| 11 | public stream - с сохранением истории (историю хранить на сервере в БД)                                                                                                                               |
| 12 | мат. задачи, сервер через случайное время шлет задачу. На сервере есть список задач. Задачи можно добавлять, зайдя по другому URL                                                                     |
| 13 | брейн-ринг. Ведущий входит по специальному url и задает вопросы. Участники получают вопрос, и посылают ответ. Очки получает первый кто ответил правильно на вопрос.                                   |
| 14 | Движение объектов (использовать HTML5 Canvas), у каждого клиента подключенного есть объект , которым он может управлять. При подключении нового клиента, добавляется объект на поле.                  |
| 15 | Движение объектов (использовать HTML5 Canvas), у каждого подключенного клиента есть объект , одного из двух типов. Одни охотники, другие убегают.                                                     |
| 16 | Онлайн-аукцион, предложения на товар обновляются динамически. Товар сменяются после пришествия времени. Товар отдается клиенту с самой высокой ставкой (просто выводится сообщение, кому было отдано) |
| 17 | Магазин, с моментальным отображением, количества товара (наличия)                                                                                                                                     |
| 18 | Сдача квартир. В реальном времени происходит обновление новых объявлений.                                                                                                                             |
| 19 | Продажа билетов на поезда. В реальном времени отображать количество не проданных билетов на конкретный поезд.                                                                                         |
| 20 | Продажа туристических путевок. Пользователь может войти на сайт и заказать тур, у остальных пользователей присутствующих в данный момент на сайте отражается данное изменение.                        |
| 21 | Номера гостиницы. Можно войти на сайт, увидеть свободные номера. Можно выбрать любой номер. В реальном времени у всех других пользователей должно отобразиться изменение.                             |
| 22 | Заказ столиков в ресторане. Можно войти на сайт, увидеть свободные столики. Можно выбрать любой столик. В реальном времени у всех других пользователей должно отобразиться изменение.                 |
| 23 | Заказ столиков в ресторане, с прорисовкой столиков, с возможностью выбирать кликами.                                                                                                                  |
| 24 | Места в кинотеатре, с прорисовкой, с возможностью кликать по местам                                                                                                                                   |
| 25 | Запись к врачу. В реальном времени должны отображаться свободные места (время)                                                                                                                        |
| 26 | Онлайн консультация. По аналогии с приватным чатом, есть пользователи, к которым можно подключиться и общаться.                                                                                       |
| 27 | Вызов такси. Можно открыть сайт, и получить список свободных такси. Заказы отображаются в реальном времени для всех подключенных клиентов.                                                            |

|    |                                                                                                                                                                                                                                                                                         |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 28 | Выбор места в самолете. Пользователь заходит на сайт, на котором показана схематичная рассадка в самолете. При выборе места у остальных пользователей отображается данное действие.                                                                                                     |
| 29 | Погода, сервер генерирует изменения погоды - клиенты входят на сайт и получают обновление прогноза погоды в реальном времени. Прогноз погоды нужно получать с любого публичного сервиса.                                                                                                |
| 30 | Курсы валют, сервер генерирует изменения курсов валют - клиенты входят на сайт и получают обновленные курсы валют в реальном времени. Курсы валют нужно получать с любого публичного сервиса.                                                                                           |
| 31 | Новостной сервис. У всех пользователей сайта в реальном времени показываются новости (в текстовом формате), которые нужно получать с любого публичного новостного сервиса.                                                                                                              |
| 32 | Заказ мест в театре. При входе на сайт пользователь видит схему рассадки в театре. При выборе места это действие должно показываться у других пользователей.                                                                                                                            |
| 33 | Результаты футбольных матчей в реальном времени отображаются всем пользователям на сайте.                                                                                                                                                                                               |
| 34 | Результаты скачек в реальном времени отображаются всем пользователям на сайте.                                                                                                                                                                                                          |
| 35 | Текстовый стрим баскетбольного матча. Streamer заходит на главную страницу, вводит логин и начинает писать. Подписчики вводят url стрима. Изменения информации на стриме происходит в реальном времени.                                                                                 |
| 36 | Колесо фортуны. Все пользователи делают ставки в течение определенного промежутка времени пока сервер крутит колесо и показывает результат. Всем пользователям показывается логин призера. Затем после небольшого ожидания отсчет запускается заново.                                   |
| 37 | Решение интегралов. Один из пользователей (любой находящийся на сайте) задает пределы интегрирования, точность и ключевое слово для решения (например, calculate), всем отображается введенная информация. Сервер решает задачу и отправляет результат всем находящимся на сайте.       |
| 38 | Ставки, с динамическим расчетом коэффициентов (боксерский поединок). В зависимости от людей сделавших ставку (ставку может сделать любой посетитель сайта выбрав одного из двух боксеров), происходит расчет коэффициентов (учитывается общая сумма и количество ставок на спортсмена). |
| 39 | Текстовый стрим бейсбольного матча. Streamer заходит на главную страницу, вводит логин и начинает писать. Подписчики вводят url стрима. Изменения информации на стриме происходит в реальном времени.                                                                                   |
| 40 | Рулетка. Все пользователи делают ставки в течение определенного промежутка времени пока сервер крутит рулетку и показывает результат. Всем пользователям показывается логин призера. Затем после небольшого ожидания отсчет запускается заново.                                         |
| 41 | Спорт-лото. Все пользователи делают выбирают шесть чисел в диапазоне от 1                                                                                                                                                                                                               |

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                         |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | до 36, в течение определенного промежутка времени, затем ставки не принимаются и сервер выдает последовательность из шести чисел. Победителем считается тот, у кого есть хотя бы одно совпадение. Всем пользователям показываются логины призеров. Затем после небольшого ожидания игра возобновляется. |
| <p>Для всех заданий представленных ниже, цель задания та же что и в предыдущих лабораторных работах. Клиент входит на сайт, отправляет задачу на сервер (одну или несколько). И в реальном времени получает ответ от сервера. Не должно происходить блокировки после отправления задачи на сервер. Т.е. во время решения задачи, можно отправлять еще задачи. После того как сервер решил задачу, он отправляет результат клиенту, который должен сразу же отобразиться у него (на более высокую оценку отправка результата от клиента и получение результата должно отображаться у всех пользователей).</p> |                                                                                                                                                                                                                                                                                                         |
| 42                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Вычисление псевдослучайного числа используя регистр сдвига с линейной обратной связью (LFSR)                                                                                                                                                                                                            |
| 43                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Одномерная оптимизация функции методом деления пополам                                                                                                                                                                                                                                                  |
| 44                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Mergesort                                                                                                                                                                                                                                                                                               |
| 46                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Quicksort                                                                                                                                                                                                                                                                                               |
| 47                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Heapsort                                                                                                                                                                                                                                                                                                |
| 50                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Умножение матриц                                                                                                                                                                                                                                                                                        |
| 52                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Treesort                                                                                                                                                                                                                                                                                                |
| 53                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Одномерная оптимизация функции методом касательных                                                                                                                                                                                                                                                      |
| 54                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Разбор математического выражения и вычисление с учетом скобок и приоритетов операций (+ - * / ) . Использовать обратную польскую запись                                                                                                                                                                 |
| 55                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Вычисление синуса (без использования математических библиотек, используя разложение в ряд Маклоррена)                                                                                                                                                                                                   |
| 56                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Вычисление экспоненты (без использования математических библиотек, используя разложение в ряд Маклоррена)                                                                                                                                                                                               |
| 57                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Разбор математического выражения с учетом приоритетов операций (+ - * / ) , и тригонометрических функций (sin, cos, tg) . Использовать обратную польскую запись.                                                                                                                                        |
| 58                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Вычислить число ПИ до 50 знака методом Монте-Карло                                                                                                                                                                                                                                                      |
| 59                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Вычислить число ПИ до 50 знака методом иглы Буффона                                                                                                                                                                                                                                                     |
| 60                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Вычислить число ПИ до 50 знака разложением в ряд Нилаканта                                                                                                                                                                                                                                              |
| 61                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Вычислить квадратный корень из числа, используя итерационную формулу Герона                                                                                                                                                                                                                             |

|    |                                                                                                                                                                                                                                                                   |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 62 | Вычислить квадратный корень из числа, используя алгоритм Ньютона                                                                                                                                                                                                  |
| 63 | Вычисление натурального логарифма (без использования математических библиотек, используя разложение в ряд Маклоррена)                                                                                                                                             |
| 64 | Вычисление гиперболического синуса(без использования математических библиотек, используя разложение в ряд Маклоррена)                                                                                                                                             |
| 65 | Вычислить число $\pi$ до 50 знака разложением в ряд Лейбница                                                                                                                                                                                                      |
| 66 | Решение СЛАУ через обратную матрицу                                                                                                                                                                                                                               |
| 67 | <p>Разработать конечный автомат для анализа корректности email.</p> <p>Найти все email в файле отправленном пользователем.</p> <p>Клиенту возвращается список всех найденных email в тексте</p>                                                                   |
| 68 | Найти обратную матрицу (используя метод Гаусса)                                                                                                                                                                                                                   |
| 69 | Разбор математического выражения и вычисление с учетом скобок и приоритетов операций (+ - * / ) . Использовать дерево выражений                                                                                                                                   |
| 70 | Граф хранится на сервере, пользователь указывает название точки отправления и точки назначения, в ответ получает кратчайший путь. Для поиска пути использовать Алгоритм Флойда — Уоршелла                                                                         |
| 71 | Граф хранится на сервере, пользователь указывает название точки отправления и точки назначения, в ответ получает кратчайший путь. Для поиска пути использовать Алгоритм Дейкстры                                                                                  |
| 72 | Граф хранится на сервере, пользователь указывает название точки отправления и точки назначения, в ответ получает кратчайший путь. Для поиска пути использовать Алгоритм Беллмана — Форда                                                                          |
| 73 | Решение СЛАУ методом Гаусса                                                                                                                                                                                                                                       |
| 74 | Решение СЛАУ методом Жардана-Гауса                                                                                                                                                                                                                                |
| 75 | Найти определитель матрицы                                                                                                                                                                                                                                        |
| 76 | $LDL^T$ разложение матрицы                                                                                                                                                                                                                                        |
| 77 | Найти обратную матрицу (через единичную)                                                                                                                                                                                                                          |
| 78 | Разложение Холецкого                                                                                                                                                                                                                                              |
| 79 | LU разложение матрицы                                                                                                                                                                                                                                             |
| 80 | Шифрование строки алгоритмом RSA                                                                                                                                                                                                                                  |
| 81 | <p>Разработать конечный автомат для анализа корректности номера телефона в формате.[ +XXX (YY) XXX-XX-XX ]</p> <p>Найти все номера телефонов в файле отправленном пользователем.</p> <p>Клиенту возвращается список всех найденных номеров телефонов в тексте</p> |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 82 | <p data-bbox="316 168 1316 235">Разработать конечный автомат для анализа надежности и корректности пароля</p> <p data-bbox="316 275 1348 380">Длина пароля не менее 6 символов. Разрешены только латинские буквы, цифры и знаки: пробел _ - . В пароле обязательно должны присутствовать знаки в верхнем и нижнем регистре и цифры.</p> <p data-bbox="316 421 1332 488">Пользователю возвращается информация о том, можно ли использовать указанный пароль</p> |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|