



**Московский авиационный институт
(Национальный исследовательский университет)**

Институт № 3

Кафедра 311

Проектирование информационных систем

Лабораторная работа № 2

**Выполнил студент
Плотников Антон Сергеевич**

Группа МЗЗ-201-БК

Дата 05.03.2021 г.

**Принял преподаватель
Смирнов Владимир Юрьевич**

Москва 2021 г.

Содержание

Цель лабораторной работы	3
Глава 1. Отображение поступающих данных	4
Глава 2. Индикация/предупреждения	6
Глава 3. Используемое ПО	7

Цель лабораторной работы

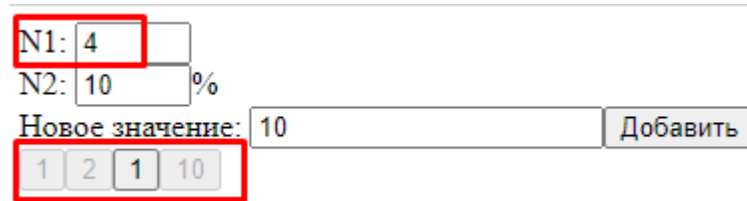
Разработать и реализовать интерфейс программы, предназначенной для использования в качестве автоматизированного рабочего места специалиста по контролю и анализу поступающей информации одного параметра.

В данном АРМ необходимо предусмотреть следующие возможности:

1. отображение значений поступающих данных: текущего значения, $N1=2..10$ предыдущих;
2. отображение полученной информации на экране в виде графика;
3. возможность выбора значений из группы текущих данных в группу выборочных значений указанным в задании способом;
4. отображения выборочных значений;
5. отображение диаграммы выборочных значений;
6. визуальная индикация предупреждения при изменении текущего значения более чем на $N2=5..40$ процентов;
7. визуальная и звуковая индикация тревоги при выходе текущего значения из заданной области допустимых значений.

Глава 1. Отображение поступающих данных

Интерфейс предоставляет возможность регулировать количество отображаемых входных значений (поле N1, Рисунок 1.1)



N1: 4
N2: 10 %
Новое значение: 10 [Добавить]
1 2 1 10

Рисунок 1.1 – Ввод лимита отображения поступающих данных

Входящие значения отображаются на графике, в количестве, ограниченном значением поля N1 (Рисунок 1.2)

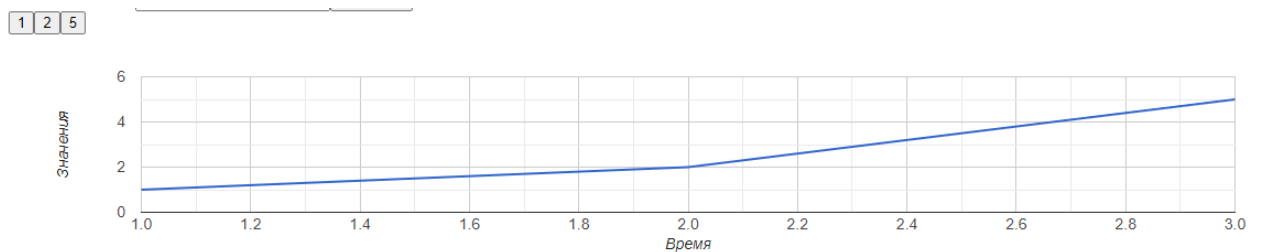


Рисунок 1.2 – График отображения входящих значений

Выбранные значения отображаются в отдельном вертикальном графике (Рисунок 1.3) и в списке над графиком (Рисунок 1.4)

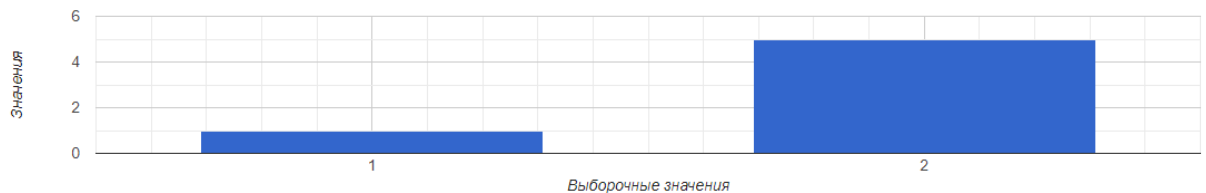


Рисунок 1.3 – График отображения выборочных значений

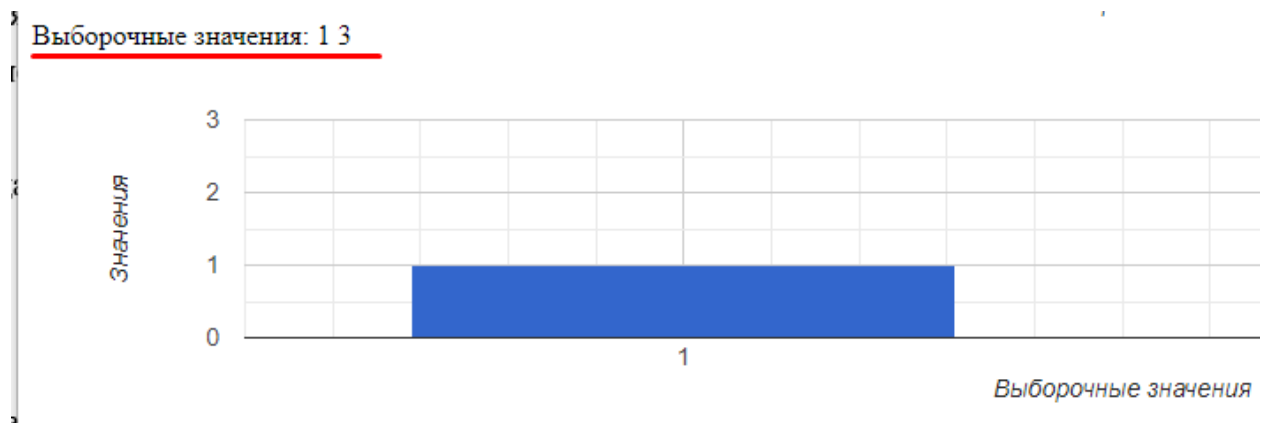


Рисунок 1.4 – Список выборочных значений

Также выборочные значения отображаются на линейном графике, где дополнительно отображается их среднее арифметическое (Рисунок 1.5)

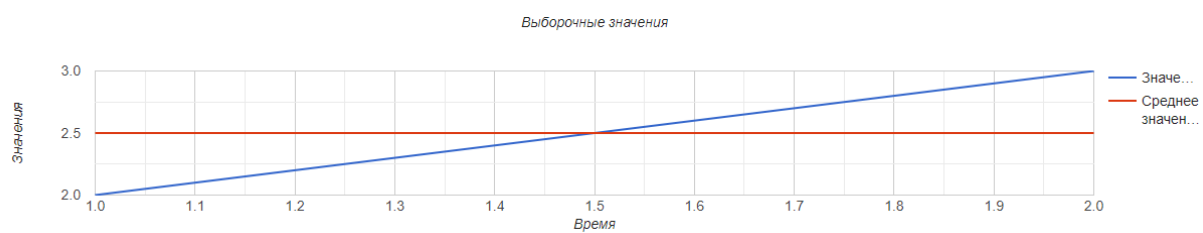


Рисунок 1.5 – Линейный график выборочных значений

Глава 2. Индикация/предупреждения

Интерфейс программы позволяет выставить количество %, при превышении которого новым значением будет выведено информационное сообщение (Рисунок 2.1)

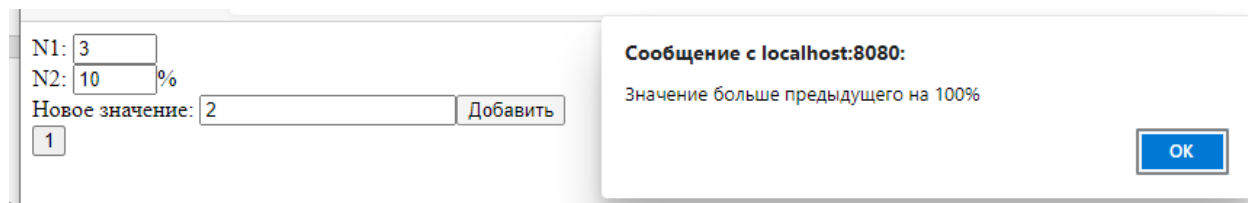


Рисунок 2.1 – Информационное сообщение о превышении N2

При вводе отрицательного значения так же будет выведено предупреждение и значение добавлено не будет (Рисунок 2.2)

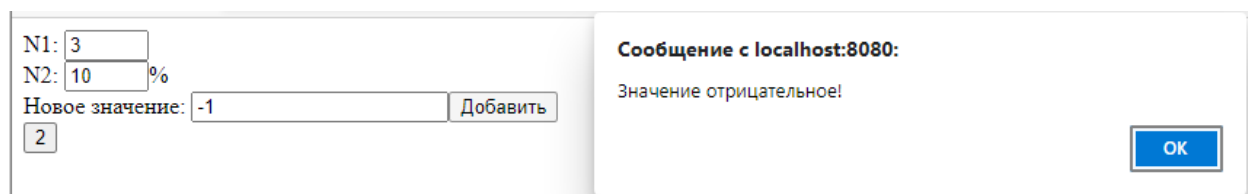


Рисунок 2.2 – Информационное сообщение об отрицательном значении

Глава 3. Используемое ПО

Интерфейс реализован на HTML5 с использованием jQuery и фреймворка Freemarker.

Серверная часть реализована на языке Java 1.8 с использованием фреймворка Spring Boot для работы веб-приложения

Глава 4. Текст программы

- HTML + jQuery

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-gb" lang="en-gb" dir="ltr">
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
  <script src="http://code.jquery.com/jquery-3.4.1.min.js"></script>
  <title>main</title>
</head>
<body>
<script language=JavaScript>
  let zn_col = 0, zn_last = 0, vybor_cnt = 0, sr_znach = 0;
  data_zn = Array();
  data_vybor = Array();
</script>
<script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
<div class=main>
  <div class="table" id="const">
    <div class="client_row">
      <div class="cell">N1: <input type=text id="n1" value='3' size=3></div>
      <div class="cell">N2: <input type=text id="n2" value='10' size=3>%</div>
    </div>
  </div>
  <div class="table" id="zhach">
    <div class="client_row">
      <div class="cell">Новое значение: <input type=text id="new_zn"><input
type=button id="new_but"
value="Добавить"></div>
    </div>
  </div>
  <div class="table" id="tekush">
    <div class="client_row">
      <div class="cell" id="tek_cell"></div>
    </div>
  </div>
  <div class="table" id="chart">
    <div class="client_row">
      <div class="cell">
        <div id="chart_div"></div>
      </div>
    </div>
  </div>
  <div class="table" id="vybor">
    <div class="client_row">
      <div class="cell" id="vybor_cell">Выборочные значения: </div>
    </div>
  </div>
  <div class="table" id="diag">
    <div class="client_row">
      <div class="cell">
        <div id="chart_div2"></div>
        <div id="chart_div3"></div>
      </div>
    </div>
  </div>
<script language=JavaScript>
  $('#new_but').click(function () {
```



```

        let new_val = $('#new_zn').val();
        if (new_val < 0) {
            alert('Значение отрицательное!');
        } else {
            zn_last++;
            $('#tek_cell').append('<input type=button value=' + new_val + '
class="tek_zn" id="tek' + zn_last + '">');
            $('#tek' + zn_last).click(function () {
                $('#vybor_cell').append($(this).val() + ' ');
                $(this).prop('disabled', true);
                vybor_cnt++;
                let zn_arr2 = [Number(vybor_cnt), Number($(this).val())];
                data_vybor.push(zn_arr2);
                let sumVybor = 0;
                data_vybor.forEach(function (innerArray, index, data_vybor) {
                    let val = innerArray[1];
                    sumVybor += val;
                });
                sr_znach = sumVybor / vybor_cnt;
                drawColumnChart();
                drawLineChartForVybor();
            });
            let zn_arr = [Number(zn_last), Number(new_val)];
            data_zn.push(zn_arr);
            zn_col = $('#n1').val();
            if ($.tek_zn'.length > zn_col) {
                $('#tek' + (zn_last - zn_col)).css('display', 'none');
            }
            drawLineChart();
            if (zn_last > 1) {
                pred = $('#tek' + (zn_last - 1)).val();
                proc = $('#n2').val();
                if ((new_val - pred) / pred > (proc / 100)) {
                    alert('Значение больше предыдущего на ' + Math.round((new_val
- pred) / pred * 100) + '%');
                }
            }
        }
    });
    google.charts.load('current', {packages: ['corechart', 'line']});
    google.charts.load('current', {packages: ['corechart', 'bar']});

    function drawLineChart() {
        var data = new google.visualization.DataTable();
        data.addColumn('number', 'X');
        data.addColumn('number', 'Значения');

        data.addRows(data_zn);

        var options = {
            hAxis: {
                title: 'Время'
            },
            vAxis: {
                title: 'Значения'
            },
        };

        var chart = new
google.visualization.LineChart(document.getElementById('chart_div'));
        chart.draw(data, options);
    };

```

```

function drawLineChartForVybor() {
    var data = new google.visualization.DataTable();
    data.addColumn('number', 'X');
    data.addColumn('number', 'Значения');
    data.addColumn('number', 'Среднее значение');

    data_vybor.forEach(function (cur, index, data_vybor) {
        data.addRow([cur[0], cur[1], sr_znach]);
    });

    var options = {
        hAxis: {
            title: 'Время'
        },
        vAxis: {
            title: 'Значения'
        },
    };

    var chart = new
google.visualization.LineChart(document.getElementById('chart_div3'));
    chart.draw(data, options);
}

function drawColumnChart() {
    var data = new google.visualization.DataTable();
    data.addColumn('number', 'X');
    data.addColumn('number', 'Значения');

    data.addRows(data_vybor);

    var options = {
        hAxis: {
            title: 'Выборочные значения',
            format: '0'
        },
        vAxis: {
            title: 'Значения',
            format: '0'
        },
    };

    var chart = new
google.visualization.ColumnChart(document.getElementById('chart_div2'));
    chart.draw(data, options);
};
</script>

</div>
</body>
</html>

```

- Java + Spring Boot

```
package ru.labs.second;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class SecondApplication {

    public static void main(String[] args) {
        SpringApplication.run(SecondApplication.class, args);
    }

}
```

```
package ru.labs.second.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class MainController {

    @GetMapping("/")
    public String mainPage() {
        return "mainPage";
    }

}
```

Вывод

Мы научились создавать интерфейсы, позволяющие анализировать входящие значения.

Список литературы

- <https://freemarker.apache.org/docs/index.html> - Документация фреймворка Freemarker;
- <https://dev.w3.org/html5/html-author/> - Документация HTML5;
- <https://docs.oracle.com/javase/8/docs/> - Документация Java 8;
- <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/> - Документация Spring Boot;