

Лабораторная работа «Построение графиков и диаграмм, библиотека D3»

Задание. Создать html-страницу, которая включает (рисунок 1):

- раздел настройки построения графика;
- область **svg** для вывода графика;
- таблица, по которой строится график.

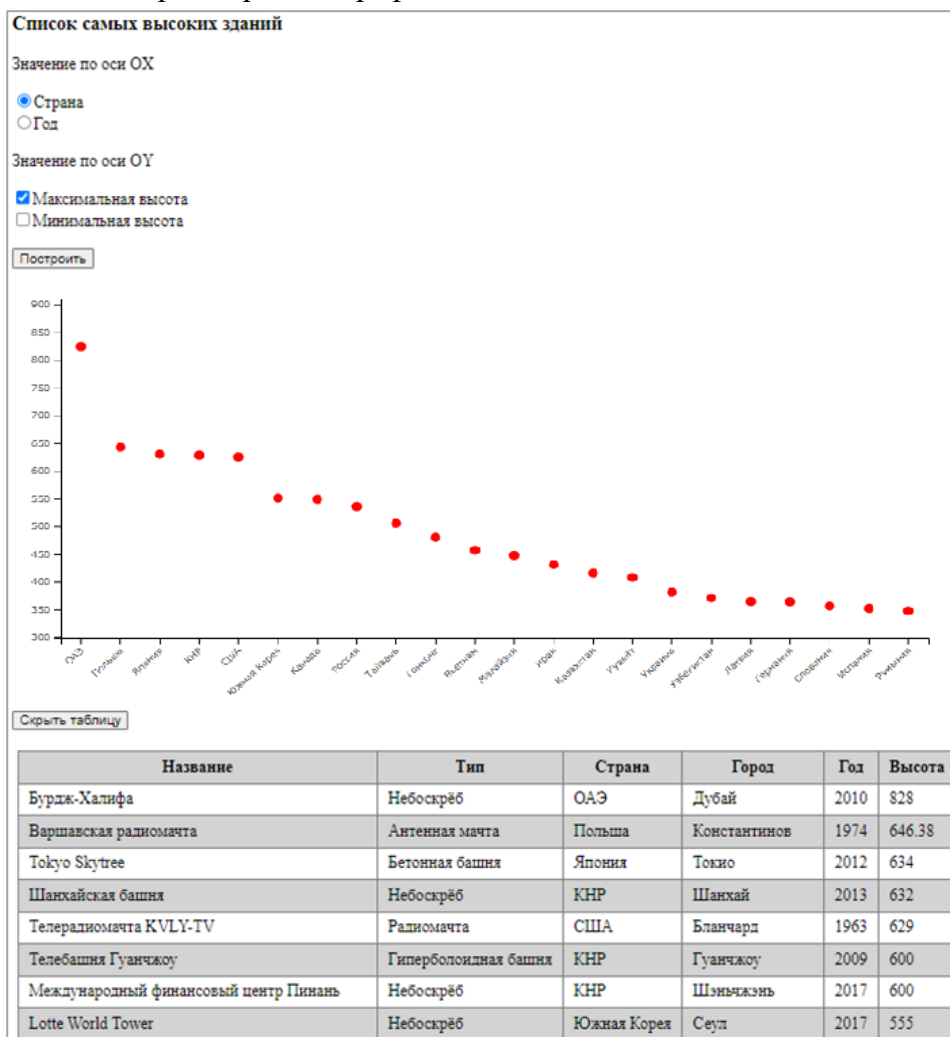


Рисунок 1. Страница HTML

Таблица формируется на основе данных, размещенных в файле **data.js**. Фрагмент данных:

```
buildings= [
  {
    "Название": "Бурдж-Халифа",
    "Тип": "Небоскрёб",
    "Страна": "ОАЭ",
    "Город": "Дубай",
    "Год": 2010,
    "Высота": 828
  },
  {
    "Название": "Варшавская радиомачта",
    ...
  },
  ...
]
```

1

При клике на кнопку **Построить** – должна выводиться диаграмма по указанным пользователем параметрам.

При клике на кнопку **Скрыть таблицу** – таблица должна скрываться, подпись кнопки меняться на **Показать таблицу**.

При загрузке страницы ни диаграмма, ни таблица не должны выводиться (рисунок 2).

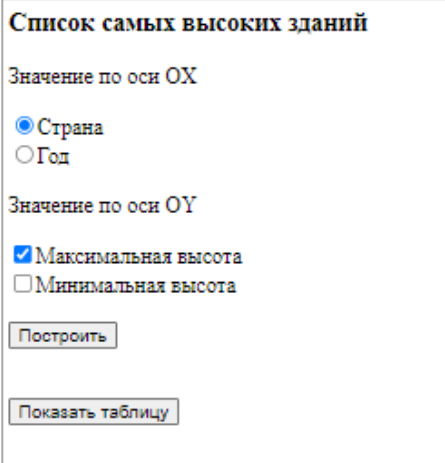


Рисунок 2. Вид страницы при загрузке

Порядок выполнения работы

1. Создать HTML-файл следующей структуры:

```
<html>
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="style.css">
  <script src="http://d3js.org/d3.v7.min.js"></script>
</head>
<body>
  <h3>Список самых высоких зданий</h3>

  <div>
    <form>
      <p>Значение по оси OX</p>
      <input type="radio" name="ox" value="Страна" checked>Страна<br>
      <input type="radio" name="ox" value="Год">Год<br>
      <p>Значение по оси OY</p>
      <input type="checkbox" name="oy" checked>Максимальная высота<br>
      <input type="checkbox" name="oy">Минимальная высота<br>
      <p>
        <input type="button" value="Построить" onclick="drawGraph(this.form)" >
      </p>
    </form>
    <svg height=10> </svg>
  </div>

  <div class="table">
    <p>
      <input type="button" id="showTable" value="Показать таблицу" >
    </p>
    <table> </table>
  </div>
</body>
<script src="program.js"></script>
<script src="data.js"></script>
</html>
```

2. Создать файл со стилями (**style.css**), в котором описать произвольный стиль для вывода таблицы.

3. Создать файл **program.js**, в котором с событием *click* по кнопке **showTable** связать функцию для вывода/удаления таблицы, построенной на основе массива **buildings**:

```
d3.select("#showTable")
.on('click', function() {
  let buttonValue = d3.select(this);

  if (buttonValue.property("value") === "Показать таблицу") {

    buttonValue.attr("value", "Скрыть таблицу");

    // создание таблицы
    let table = d3.select("div.table")
      .select("table")

    // сформировать таблицу на основе массива buildings
    // стр. 11-12 теоретического материала
    ...

  } else {
    buttonValue.attr("value", "Показать таблицу");

    d3.select("div.table")
      .select("table")
      // удалить все строки таблицы
      ...
  }
});
```

4. В результате на странице должна выводиться таблица или удаляться ее строки при клике по кнопке **showTable**.

5. Реализовать функцию **createArrGraph()**, которая создает новый массив, необходимый для построения диаграммы (выполняет группировку по столбцу). Например, если в качестве значений по оси ОХ выбрана **Страна**, то должен получиться следующий массив:

```
[{
  "labelX": "ОАЭ",
  "values": [354.6, 828]
},
{
  "labelX": "Польша",
  "values": [646.38, 646.38]
},
...
]
// Входные данные:
// data - исходный массив (например, buildings)
// key - поле, по которому осуществляется группировка

function createArrGraph(data, key) {

  groupObj = d3.group(data, d => d[key]);

  let arrGraph =[];
```

```

for(let entry of groupObj) {
    let minMax = d3.extent(entry[1].map(d => d['Высота']));
    arrGraph.push({labelX : entry[0], values : minMax});
}

return arrGraph;
}

```

В результате в консоли можно посмотреть полученный массив, если вызвать функцию **createArrGraph()** с параметрами **buildings** и "Страна".

6. В файл **style.css** добавить описание стилей для текста и осей:

```

svg text {
    font: 8px Verdana;
}
svg path, line {
    fill: none;
    stroke: #333333;
    width: 1px;
}

```

7. В **program.js** включить настройку параметров и инициализацию svg-элемента:

```

const marginX = 50;
const marginY = 50;
const height = 400;
const width = 800;

let svg = d3.select("svg")
    .attr("height", height)
    .attr("width", width);

```

8. Реализовать функцию **drawGraph()** для построения диаграммы, в качестве параметра функция получает данные формы.

```

function drawGraph(data) {
    // значения по оси OX
    const keyX = data.ox.value;

    // значения по оси OY
    const isMin = data.oy[1].checked;
    const isMax = data.oy[0].checked;

    // создаем массив для построения графика
    const arrGraph = createArrGraph(buildings, keyX);

    svg.selectAll('*').remove();

    // создаем шкалы преобразования и выводим оси
    const [scX, scY] = createAxis(arrGraph, isMin, isMax);

    // рисуем графики
    if (isMin) {
        createChart(arrGraph, scX, scY, 0, "blue")
    }
    if (isMax) {
        createChart(arrGraph, scX, scY, 1, "red")
    }
}

```

9. Реализовать функцию **createAxis()** для создания шкал преобразования и рисования осей. Функция в качестве параметров получает массив для построения диаграммы, а также признаки, какие значения по оси ОУ будут выводиться (**true** – если линия должна быть построена, **false** – в противном случае).

```
function createAxis(data, isFirst, isSecond) {

    // в зависимости от выбранных пользователем данных по ОУ
    // находим интервал значений по оси ОУ
    let firstRange = d3.extent(data.map(d => d.values[0]));
    let secondRange = d3.extent(data.map(d => d.values[1]));

    let min = firstRange[0];
    let max = secondRange[1];

    // функция интерполяции значений на оси
    let scaleX = d3.scaleBand()
        .domain(data.map(d => d.labelX))
        .range([0, width - 2 * marginX]);

    let scaleY = d3.scaleLinear()
        .domain([min * 0.85, max * 1.1 ])
        .range([height - 2 * marginY, 0]);

    // создание осей
    let axisX = d3.axisBottom(scaleX); // горизонтальная
    let axisY = d3.axisLeft(scaleY); // вертикальная

    // отрисовка осей в SVG-элементе
    svg.append("g")
        .attr("transform", `translate(${marginX}, ${height - marginY})`)
        .call(axisX)
        .selectAll("text") // подписи на оси - наклонные
        .style("text-anchor", "end")
        .attr("dx", "-.8em")
        .attr("dy", ".15em")
        .attr("transform", d => "rotate(-45)");

    svg.append("g")
        .attr("transform", `translate(${marginX}, ${marginY})`)
        .call(axisY);

    return [scaleX, scaleY]
}
```

10. Реализовать функцию **createChart()** для построения точечной диаграммы. В качестве параметров функция получает массив для построения, шкалы преобразования координат **x** и **y**, индекс диаграммы (0 – выводятся минимальные высоты, 1 – максимальные) и цвет отображения точек.

```
function createChart(data, scaleX, scaleY, index, color) {
    const r = 4
    // чтобы точки не накладывались, сдвинем их по вертикали
    let ident = (index == 0)? -r / 2 : r / 2;

    svg.selectAll(".dot")
        .data(data)
        .enter()
        .append("circle")
        .attr("r", r)
```

```

.attr("cx", d => scaleX(d.labelX) + scaleX.bandwidth() / 2)
.attr("cy", d => scaleY(d.values[index]) + ident)
.attr("transform", `translate(${marginX}, ${marginY})`)
.style("fill", color)
}

```

В результате, если выбрать построение диаграммы по **Странам**, вывод и **Максимальных**, и **Минимальных** значений высот, при клику по кнопке **Построить**, должна получиться диаграмма, как показано на рисунок 3.

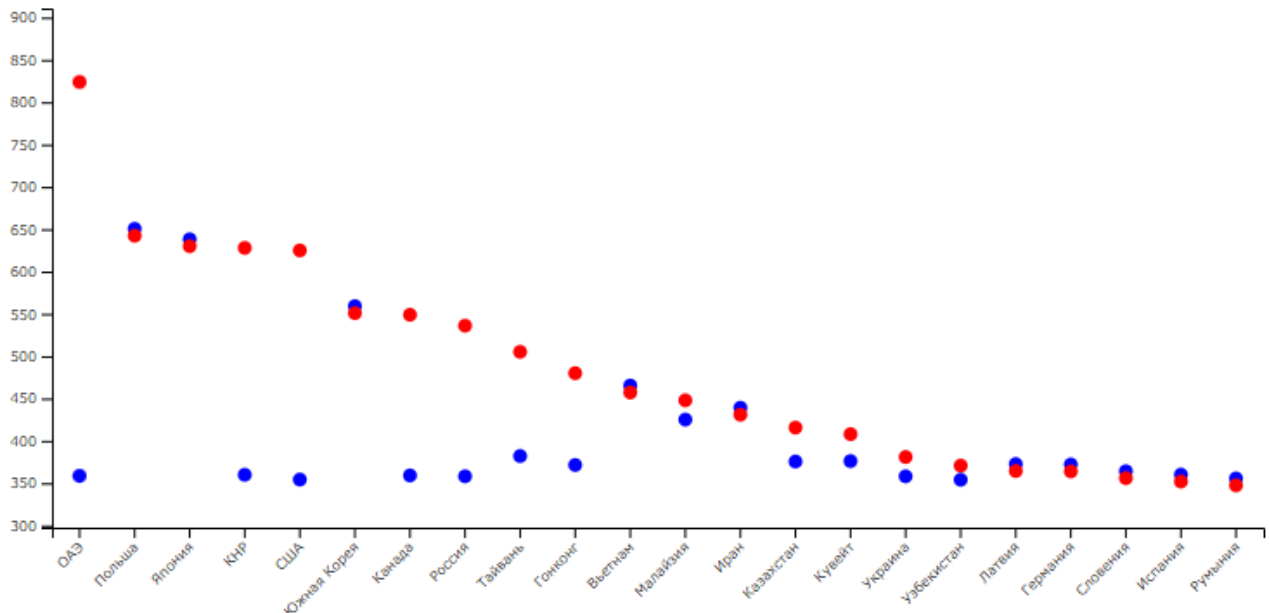


Рисунок 3. Диаграмма максимальных и минимальных высот по странам

Самостоятельные задания

1. Внести изменения в код функции так, чтобы при построении учитывались настройки диаграммы из формы. При этом:

- если не выбрана ни одна категория для построения графика по оси ОУ, вывести ошибку;
- если выбирается только минимальное значение высоты или только максимальное, изменить шкалу построения по оси ОУ.

2. При загрузке страницы вывести диаграмму, которая построена по настройкам из формы, выставленным по умолчанию.

3. Добавить в форму дополнительную опцию, которая определяет тип диаграммы: точечная диаграмма или столбчатая диаграмма. Реализовать построение диаграммы с учетом этой опции.

4*. Массив **arrGraph** в функции **createArrGraph()** получен с помощью цикла (не в стиле D3). Получить этот массив **БЕЗ** цикла, средствами D3.