

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа «Программная инженерия»

СОГЛАСОВАНО
Доцент департамента
программной инженерии
факультета компьютерных наук
канд. техн. наук

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук

И.Ю. Самоненко
«__» _____ 2022 г.

В.В.Шилов
«__» _____ 2022 г.

ПРОГРАММА ВИЗУАЛИЗАЦИИ АЛГОРИТМОВ И СТРУКТУР ДАННЫХ

Текст программы

**ЛИСТ УТВЕРЖДЕНИЯ
RU.17701729.10.03-01 12 01-1-ЛУ**

Исполнитель
студент группы БПИ203
_____/ П.Н.Ломакин/
«__» _____ 2022 г.

УТВЕРЖДЕН
RU.17701729.10.03-01 12 01-1-ЛУ

ПРОГРАММА ВИЗУАЛИЗАЦИИ АЛГОРИТМОВ И СТРУКТУР ДАННЫХ

Текст программы

RU.17701729.10.03-01 12 01-1-ЛУ

Листов 53

1. ТЕКСТ ПРОГРАММЫ

Данное приложение написано на языке JavaScript, использована среда разработки Visual Studio Code.

Программа состоит из 6 файлов расширения .js, 1 файла расширения .html и 4 файлов расширения .css. В данном документе описано содержимое данных файлов.

1.1 Файлы расширения .js

1.1.1 Файл bloom.js

```
2. let filterSize, numberOfHash;
3. let visualizationInProcess = false;
4.
5. function addEventWhenButtonBuildFilterWasClicked(buttonBuildFilter, inputFilterSize, inputNumberOfHash) {
6.     buttonBuildFilter.addEventListener('click', () => {
7.         if (visualizationInProcess) {
8.             return;
9.         }
10.        inputFilterSize.addEventListener('click', () => {
11.            if (inputFilterSize.style.borderColor == "red") {
12.                inputFilterSize.value = "";
13.                inputFilterSize.style.borderColor = "";
14.            }
15.        });
16.        inputNumberOfHash.addEventListener('click', () => {
17.            if (inputNumberOfHash.style.borderColor == "red") {
18.                inputNumberOfHash.value = "";
19.                inputNumberOfHash.style.borderColor = "";
20.            }
21.        });
22.        let dontBuildFilter = false;
23.        if (isFloat(parseFloat(inputFilterSize.value)) || !(Number.isInteger(parseInt(inputFilterSize.value)) &&
        parseInt(inputFilterSize.value) <= 13 && parseInt(inputFilterSize.value) >= 1)) {
24.            inputFilterSize.style.borderColor = "red";
25.            dontBuildFilter = true;
26.        }
27.        if (isFloat(parseFloat(inputNumberOfHash.value)) || !(Number.isInteger(parseInt(inputNumberOfHash.value))
        && parseInt(inputNumberOfHash.value) <= 13 && parseInt(inputNumberOfHash.value) >= 1)) {
28.            inputNumberOfHash.style.borderColor = "red";
29.            dontBuildFilter = true;
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

30.     }
31.
32.     if (dontBuildFilter) {
33.         // Reset the main playground.
34.         document.getElementById('playground-main').innerHTML = "";
35.         let playgroundMainTitle = document.createElement('div');
36.         playgroundMainTitle.className = 'playground-main-title';
37.         playgroundMainTitle.id = 'playground-main-title';
38.         playgroundMainTitle.innerHTML = "playground";
39.         document.getElementById('playground-main').appendChild(playgroundMainTitle);
40.         return;
41.     }
42.     if (Number.isInteger(parseInt(inputFilterSize.value)) && parseInt(inputFilterSize.value) <= 13 &&
parseInt(inputFilterSize.value) >= 1 &&
43.         Number.isInteger(parseInt(inputNumberOfHash.value)) && parseInt(inputNumberOfHash.value) <= 13 &&
parseInt(inputNumberOfHash.value) >= 1) {
44.
45.         filterSize = parseInt(inputFilterSize.value);
46.         numberOfHash = parseInt(inputNumberOfHash.value);
47.
48.         document.getElementById('pseudocode-window').innerHTML = "";
49.         let pseudocodeTitle = document.createElement('div');
50.         pseudocodeTitle.className = 'pseudocode-title';
51.         pseudocodeTitle.id = 'pseudocode-title';
52.         pseudocodeTitle.innerHTML = 'CODE';
53.         document.getElementById('pseudocode-window').appendChild(pseudocodeTitle);
54.
55.         // Reset the main playground.
56.         document.getElementById('playground-main').innerHTML = "";
57.
58.         // Creating shell for filter.
59.         createFilterArray();
60.         // Creating shell for hash functions.
61.         createHashFunctions();
62.
63.         buildBloomFilter(filterSize);
64.
65.         createZoneForBarChart();
66.
67.         // Elements to be added to the bloom filter.
68.         let addedElementsList = [];
69.
70.         // Creating universal hash functions.
71.         let hashFunctions = new UniversalHashFunctions(numberOfHash).generateFunctions();
72.
73.         createChangeFunctionParametersElements();

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
74.     document.getElementById('accept-changes-to-function-button').onclick = () => {
75.         if (!visualizationInProgress) {
76.             if (!(document.getElementById('number-of-function-to-change-input').value == "" ||
document.getElementById('a-parameter-to-change-input').value == "" ||
77.                 document.getElementById('b-parameter-to-change-input').value == "" || document.getElementById('p-
parameter-to-change-input').value == "" ||
78.                 document.getElementById('number-of-function-to-change-input').style.borderColor == "red" ||
document.getElementById('a-parameter-to-change-input').style.borderColor == "red" ||
79.                 document.getElementById('b-parameter-to-change-input').style.borderColor == "red" ||
document.getElementById('p-parameter-to-change-input').style.borderColor == "red")) {
80.
81.                 let numberOfFunction = parseInt(document.getElementById('number-of-function-to-change-
input').value);
82.                 let parameterA = parseInt(document.getElementById('a-parameter-to-change-input').value);
83.                 let parameterB = parseInt(document.getElementById('b-parameter-to-change-input').value);
84.                 let parameterP = parseInt(document.getElementById('p-parameter-to-change-input').value);
85.                 document.getElementById('number-of-function-to-change-input').value = "";
86.                 document.getElementById('a-parameter-to-change-input').value = "";
87.                 document.getElementById('b-parameter-to-change-input').value = "";
88.                 document.getElementById('p-parameter-to-change-input').value = "";
89.                 let newFunction = value => {
90.                     return [(parameterA * value + parameterB) % parameterP, parameterA, parameterB, parameterP];
91.                 }
92.                 hashFunctions[numberOfFunction - 1] = newFunction;
93.                 buildListOfHashFunctions(numberOfHash, hashFunctions, filterSize);
94.             }
95.         }
96.     }
97.     document.getElementById('change-parameters-of-function-randomly').onclick = () => {
98.         if (!visualizationInProgress) {
99.             if (document.getElementById('number-of-function-to-change-input').value != "" &&
100.                 document.getElementById('number-of-function-to-change-input').style.borderColor != "red") {
101.                 let numberOfFunction = parseInt(document.getElementById('number-of-function-to-change-
input').value);
102.                 let parameterP = Math.floor(Math.random() * (100000) + 1);
103.                 let parameterA = Math.floor(Math.random() * (parameterP - 1) + 1);
104.                 let parameterB = Math.floor(Math.random() * parameterP);
105.                 document.getElementById('number-of-function-to-change-input').value = "";
106.                 document.getElementById('a-parameter-to-change-input').value = "";
107.                 document.getElementById('b-parameter-to-change-input').value = "";
108.                 document.getElementById('p-parameter-to-change-input').value = "";
109.                 let newFunction = value => {
110.                     return [(parameterA * value + parameterB) % parameterP, parameterA, parameterB, parameterP];
111.                 }
112.                 hashFunctions[numberOfFunction - 1] = newFunction;
113.                 buildListOfHashFunctions(numberOfHash, hashFunctions, filterSize);
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

114.     }
115.     }
116. }
117. buildListOfHashFunctions(numberOfHash, hashFunctions, filterSize);
118. createButtonAddElement();
119. createInputAddElement();
120. createButtonCheckElementAvailability();
121. createShowAddedElementsButton();
122.
123. // After clicking button 'buttonClearFilter' all elements from
124. // filter would be deleted.
125. let buttonClearFilter = createButtonClearFilter();
126. buttonClearFilter.onclick = () => {
127.     if (!visualizationInProgress) {
128.         addedElementsList = [];
129.         document.getElementById('text-area-with-list-of-added-elements').innerHTML = "";
130.
131.         let filterCells = document.getElementById('filter-array-div').childNodes;
132.         for (let cell of filterCells) {
133.             cell.firstChild.innerHTML = '0';
134.             cell.classList.remove('highlighted');
135.         }
136.     }
137. }
138.
139. document.getElementById('add-element-button').onclick = () => {
140.     if (!visualizationInProgress) {
141.         if (document.getElementById('add-element-input').style.borderColor !== "red" &&
142.             document.getElementById('add-element-input').value !== "") {
143.             if (!addedElementsList.includes(parseInt(document.getElementById('add-element-input').value))) {
144.                 addedElementsList.push(parseInt(document.getElementById('add-element-input').value));
145.                 document.getElementById('text-area-with-list-of-added-elements').innerHTML +=
146.                     addedElementsList.length + ": " + parseInt(document.getElementById('add-element-
147.                     input').value) + "\n";
148.             }
149.             changeValuesInCellsAfterAddingElement(hashFunctions, filterSize);
150.         }
151.     }
152.     document.getElementById('check-element-availability-button').onclick = () => {
153.         if (!visualizationInProgress) {
154.             if (document.getElementById('add-element-input').style.borderColor !== "red" &&
155.                 document.getElementById('add-element-input').value !== "") {
156.                 checkElementAvailability(hashFunctions, addedElementsList);
157.             }
158.         }
159.     }
160. }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

157.     }
158.   }
159. });
160.}
161.
162.function createButtonClearFilter() {
163.  let buttonClearFilter = document.createElement('button');
164.  buttonClearFilter.className = "button-clear-filter";
165.  buttonClearFilter.id = "button-clear-filter";
166.  buttonClearFilter.innerHTML = "Clear filter";
167.  buttonClearFilter.style.backgroundColor = "background-color: rgb(98, 80, 57);"
168.  document.getElementById('playground-main').appendChild(buttonClearFilter);
169.
170.  return buttonClearFilter;
171.}
172.
173.function isFloat(n){
174.  return Number(n) === n && n % 1 !== 0;
175.}
176.
177./**
178. * Creates a zone in which the fields for entering
179. * input data for the bloom filter will be located.
180. */
181.function createZoneForBarChart() {
182.  // Div where input elements and chart would be located.
183.  let zoneForBarChart = document.createElement('div');
184.  zoneForBarChart.id = 'zone-for-bar-chart';
185.  zoneForBarChart.className = 'zone-for-bar-chart';
186.  document.getElementById('playground-main').appendChild(zoneForBarChart);
187.
188.  // Button that shows all input elements for bar chart and chart.
189.  let showZoneForBatChartButton = document.createElement('button');
190.  showZoneForBatChartButton.className = "show-zone-bar-chart-button";
191.  showZoneForBatChartButton.id = "show-zone-bar-chart-button";
192.  showZoneForBatChartButton.innerHTML = "Input data analysis";
193.  zoneForBarChart.appendChild(showZoneForBatChartButton);
194.
195.  let labelForFilterSizeBarChart = document.createElement('label');
196.  labelForFilterSizeBarChart.style.position = 'absolute';
197.  labelForFilterSizeBarChart.style.top = '34px';
198.  labelForFilterSizeBarChart.innerHTML = "Filter Bloom size:";
199.  labelForFilterSizeBarChart.classList.add('hide-element');
200.  labelForFilterSizeBarChart.style.fontStyle = "italic";
201.  zoneForBarChart.appendChild(labelForFilterSizeBarChart);
202.

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

203. // Input for filter size for bar chart.
204. let inputFilterSizeForBarChart = document.createElement('input');
205. inputFilterSizeForBarChart.className = 'input-filter-size-for-bar-chart';
206. inputFilterSizeForBarChart.id = 'input-filter-size-for-bar-chart';
207. inputFilterSizeForBarChart.classList.add('hide-element');
208. inputFilterSizeForBarChart.placeholder = "your number";
209. zoneForBarChart.appendChild(inputFilterSizeForBarChart);
210.
211. inputFilterSizeForBarChart.onblur = () => {
212.   if (inputFilterSizeForBarChart.value !== "" && (parseFloat(inputFilterSizeForBarChart.value)) ||
213.     !Number.isInteger(parseInt(inputFilterSizeForBarChart.value)) ||
214.     String(parseInt(inputFilterSizeForBarChart.value)).length > 16 || parseInt(inputFilterSizeForBarChart.value) <
215.     1) && inputFilterSizeForBarChart.style.borderColor !== "red") {
216.     inputFilterSizeForBarChart.style.borderColor = "red";
217.   }
218. }
219. inputFilterSizeForBarChart.addEventListener('click', () => {
220.   if (inputFilterSizeForBarChart.style.borderColor === "red") {
221.     inputFilterSizeForBarChart.style.borderColor = "";
222.     inputFilterSizeForBarChart.value = "";
223.   }
224. });
225.
226. let labelNumberOfHashFunctionsForBarChart = document.createElement('label');
227. labelNumberOfHashFunctionsForBarChart.style.position = 'absolute';
228. labelNumberOfHashFunctionsForBarChart.style.top = '78px';
229. labelNumberOfHashFunctionsForBarChart.innerHTML = "Number of hash functions:";
230. labelNumberOfHashFunctionsForBarChart.classList.add('hide-element');
231. labelNumberOfHashFunctionsForBarChart.style.fontStyle = "italic";
232. zoneForBarChart.appendChild(labelNumberOfHashFunctionsForBarChart);
233.
234. // Input for number of hash functions for bar chart.
235. let inputNumberOfHashForBarChart = document.createElement('input');
236. inputNumberOfHashForBarChart.className = 'input-number-of-hash-functions-for-bar-chart';
237. inputNumberOfHashForBarChart.id = 'input-number-of-hash-functions-for-bar-chart';
238. inputNumberOfHashForBarChart.classList.add('hide-element');
239. inputNumberOfHashForBarChart.placeholder = "your number";
240. zoneForBarChart.appendChild(inputNumberOfHashForBarChart);
241.
242. inputNumberOfHashForBarChart.onblur = () => {
243.   if (inputNumberOfHashForBarChart.value !== "" && (parseFloat(inputNumberOfHashForBarChart.value)) ||
244.     !Number.isInteger(parseInt(inputNumberOfHashForBarChart.value)) ||

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата


```

245.     String(parseInt(inputNumberOfHashForBarChart.value)).length > 16 ||
        parseInt(inputNumberOfHashForBarChart.value) < 1) && inputNumberOfHashForBarChart.style.borderColor !=
        "red") {
246.         inputNumberOfHashForBarChart.style.borderColor = "red";
247.     }
248. }
249.
250. inputNumberOfHashForBarChart.addEventListener('click', () => {
251.     if (inputNumberOfHashForBarChart.style.borderColor == "red") {
252.         inputNumberOfHashForBarChart.style.borderColor = "";
253.         inputNumberOfHashForBarChart.value = "";
254.     }
255. });
256.
257. let labelNumberOfElementsToAdd = document.createElement('label');
258. labelNumberOfElementsToAdd.style.position = 'absolute';
259. labelNumberOfElementsToAdd.style.top = '123px';
260. labelNumberOfElementsToAdd.innerHTML = "Number of elements to add:";
261. labelNumberOfElementsToAdd.classList.add('hide-element');
262. labelNumberOfElementsToAdd.style.fontStyle = "italic";
263. zoneForBarChart.appendChild(labelNumberOfElementsToAdd);
264.
265. // Input number of elements to add to filter for bar chart.
266. let inputNumberOfElementsToAddForBarChart = document.createElement('input');
267. inputNumberOfElementsToAddForBarChart.className = 'input-number-of-elements-to-add-for-bar-chart';
268. inputNumberOfElementsToAddForBarChart.id = 'input-number-of-elements-to-add-for-bar-chart';
269. inputNumberOfElementsToAddForBarChart.classList.add('hide-element');
270. inputNumberOfElementsToAddForBarChart.placeholder = "your number";
271. zoneForBarChart.appendChild(inputNumberOfElementsToAddForBarChart);
272.
273. inputNumberOfElementsToAddForBarChart.onblur = () => {
274.     if (inputNumberOfElementsToAddForBarChart.value != "" &&
        (parseFloat(parseFloat(inputNumberOfElementsToAddForBarChart.value)) ||
275.         !Number.isInteger(parseInt(inputNumberOfElementsToAddForBarChart.value)) ||
276.         String(parseInt(inputNumberOfElementsToAddForBarChart.value)).length > 16 ||
        parseInt(inputNumberOfElementsToAddForBarChart.value) < 1) &&
277.         inputNumberOfElementsToAddForBarChart.style.borderColor != "red") {
278.         inputNumberOfElementsToAddForBarChart.style.borderColor = "red";
279.     }
280. }
281.
282. inputNumberOfElementsToAddForBarChart.addEventListener('click', () => {
283.     if (inputNumberOfElementsToAddForBarChart.style.borderColor == "red") {
284.         inputNumberOfElementsToAddForBarChart.style.borderColor = "";
285.         inputNumberOfElementsToAddForBarChart.value = "";
286.     }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

287. });
288.
289. let labelNumberOfElementsToCheck = document.createElement('label');
290. labelNumberOfElementsToCheck.style.position = 'absolute';
291. labelNumberOfElementsToCheck.style.top = '167px';
292. labelNumberOfElementsToCheck.innerHTML = "Number of elements to check:";
293. labelNumberOfElementsToCheck.classList.add('hide-element');
294. labelNumberOfElementsToCheck.style.fontStyle = "italic";
295. zoneForBarChart.appendChild(labelNumberOfElementsToCheck);
296.
297. // Input number of elements to check availability in filter for bar chart.
298. let inputNumberOfElementsToCheckForBarChart = document.createElement('input');
299. inputNumberOfElementsToCheckForBarChart.className = 'input-number-of-elements-to-check-for-bar-chart';
300. inputNumberOfElementsToCheckForBarChart.id = 'input-number-of-elements-to-check-for-bar-chart';
301. inputNumberOfElementsToCheckForBarChart.classList.add('hide-element');
302. inputNumberOfElementsToCheckForBarChart.placeholder = "your number";
303. zoneForBarChart.appendChild(inputNumberOfElementsToCheckForBarChart);
304.
305. inputNumberOfElementsToCheckForBarChart.onblur = () => {
306.   if (inputNumberOfElementsToCheckForBarChart.value != "" &&
307.       (parseFloat(inputNumberOfElementsToCheckForBarChart.value)) ||
308.       !Number.isInteger(parseInt(inputNumberOfElementsToCheckForBarChart.value)) ||
309.       String(parseInt(inputNumberOfElementsToCheckForBarChart.value)).length > 16 ||
310.       parseInt(inputNumberOfElementsToCheckForBarChart.value) < 1) &&
311.       inputNumberOfElementsToCheckForBarChart.style.borderColor != "red") {
312.     inputNumberOfElementsToCheckForBarChart.style.borderColor = "red";
313.   }
314. }
315.
316. inputNumberOfElementsToCheckForBarChart.addEventListener('click', () => {
317.   if (inputNumberOfElementsToCheckForBarChart.style.borderColor == "red") {
318.     inputNumberOfElementsToCheckForBarChart.style.borderColor = "";
319.     inputNumberOfElementsToCheckForBarChart.value = "";
320.   }
321. });
322.
323. let labelNumbersRange = document.createElement('label');
324. labelNumbersRange.style.position = 'absolute';
325. labelNumbersRange.style.top = '210px';
326. labelNumbersRange.innerHTML = "Range for elements:";
327. labelNumbersRange.classList.add('hide-element');
328. labelNumbersRange.style.fontStyle = "italic";
329. zoneForBarChart.appendChild(labelNumbersRange);
330.
331. // Input left border for number for bar chart
332. let inputLeftBorderForNumbers = document.createElement('input');

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

331. inputLeftBorderForNumbers.className = 'input-left-border-number-for-bar-chart';
332. inputLeftBorderForNumbers.id = 'input-left-border-number-for-bar-chart';
333. inputLeftBorderForNumbers.classList.add('hide-element');
334. inputLeftBorderForNumbers.placeholder = "left border";
335. zoneForBarChart.appendChild(inputLeftBorderForNumbers);
336.
337. inputLeftBorderForNumbers.onblur = () => {
338.   if (inputLeftBorderForNumbers.value != "" && (isFloat(parseFloat(inputLeftBorderForNumbers.value)) ||
339.     !Number.isInteger(parseInt(inputLeftBorderForNumbers.value)) ||
340.     String(parseInt(inputLeftBorderForNumbers.value)).length > 16 ||
341.     parseInt(inputLeftBorderForNumbers.value) < 0) &&
342.     inputLeftBorderForNumbers.style.borderColor != "red") {
343.     inputLeftBorderForNumbers.style.borderColor = "red";
344.   }
345.
346. inputLeftBorderForNumbers.addEventListener('click', () => {
347.   if (inputLeftBorderForNumbers.style.borderColor == "red") {
348.     inputLeftBorderForNumbers.style.borderColor = "";
349.     inputLeftBorderForNumbers.value = "";
350.   }
351. });
352.
353. // Input right border for number for bar chart
354. let inputRightBorderForNumbers = document.createElement('input');
355. inputRightBorderForNumbers.className = 'input-right-border-number-for-bar-chart';
356. inputRightBorderForNumbers.id = 'input-right-border-number-for-bar-chart';
357. inputRightBorderForNumbers.classList.add('hide-element');
358. inputRightBorderForNumbers.placeholder = "right border";
359. zoneForBarChart.appendChild(inputRightBorderForNumbers);
360.
361. inputRightBorderForNumbers.onblur = () => {
362.   if (inputRightBorderForNumbers.value != "" && (isFloat(parseFloat(inputRightBorderForNumbers.value)) ||
363.     !Number.isInteger(parseInt(inputRightBorderForNumbers.value)) ||
364.     String(parseInt(inputRightBorderForNumbers.value)).length > 16) &&
365.     inputRightBorderForNumbers.style.borderColor != "red") {
366.     inputRightBorderForNumbers.style.borderColor = "red";
367.   }
368. }
369.
370. inputRightBorderForNumbers.addEventListener('click', () => {
371.   if (inputRightBorderForNumbers.style.borderColor == "red") {
372.     inputRightBorderForNumbers.style.borderColor = "";
373.     inputRightBorderForNumbers.value = "";
374.   }
375. });

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
376.
377. // After clicking this button bar chart would be redrawing.
378. let generateBloomFilterForBarChartButton = document.createElement('button');
379. generateBloomFilterForBarChartButton.className = 'button-to-build-filter-bar-chart';
380. generateBloomFilterForBarChartButton.id = 'button-to-build-filter-bar-chart';
381. generateBloomFilterForBarChartButton.classList.add('hide-element');
382. generateBloomFilterForBarChartButton.innerHTML = 'generate';
383. zoneForBarChart.appendChild(generateBloomFilterForBarChartButton);
384.
385. let barChart = createBarChart(0, 0, 0);
386.
387. generateBloomFilterForBarChartButton.onclick = () => {
388.   if (!(inputFilterSizeForBarChart.value == "" || inputNumberOfHashForBarChart.value == "" ||
389.     inputNumberOfElementsToAddForBarChart.value == "" ||
390.     inputNumberOfElementsToCheckForBarChart.value == "" ||
391.     inputLeftBorderForNumbers.value == "" || inputRightBorderForNumbers.value == "" ||
392.     inputFilterSizeForBarChart.style.borderColor == "red" || inputNumberOfHashForBarChart.style.borderColor
393.     == "red" ||
394.     inputNumberOfElementsToAddForBarChart.style.borderColor == "red" ||
395.     inputNumberOfElementsToCheckForBarChart.style.borderColor == "red" ||
396.     inputLeftBorderForNumbers.style.borderColor == "red" || inputRightBorderForNumbers.style.borderColor
397.     == "red" ||
398.     parseInt(inputLeftBorderForNumbers.value) > parseInt(inputRightBorderForNumbers.value))) {
399.     let numberOfFalsePositive, numberOfNegative, numberOfPositive;
400.     [numberOfFalsePositive, numberOfNegative, numberOfPositive] =
401.       getResultsFromBloomFilterForBarChart(inputFilterSizeForBarChart.value, inputNumberOfHashForBarChart.value,
402.         inputNumberOfElementsToAddForBarChart.value, inputNumberOfElementsToCheckForBarChart.value);
403.     barChart.destroy();
404.     barChart = createBarChart(numberOfFalsePositive, numberOfNegative, numberOfPositive);
405.     if (document.getElementById('barChart').style.display == "none") {
406.       document.getElementById('barChart').style.display = "block";
407.     }
408.   }
409. }
410.
411. let labelExplainingBarChart = document.createElement('label');
412. labelExplainingBarChart.style.position = 'absolute';
413. labelExplainingBarChart.style.top = '300px';
414. labelExplainingBarChart.innerHTML = "First, a filter is created, after which randomly generated numbers are
415. added to it, the number of which is equal to the value in the third field.<br><br>After that, new numbers are
416. randomly generated (their number is equal to the value in the fourth field), which are checked for belonging to the
417. filter.";
418. labelExplainingBarChart.style.fontStyle = "italic";
419. labelExplainingBarChart.classList.add('hide-element');
420. zoneForBarChart.appendChild(labelExplainingBarChart);
421.
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

414. showZoneForBatChartButton.onclick = () => {
415.     inputRightBorderForNumbers.classList.toggle('hide-element');
416.     inputLeftBorderForNumbers.classList.toggle('hide-element');
417.     labelNumbersRange.classList.toggle('hide-element');
418.     labelForFilterSizeBarChart.classList.toggle('hide-element');
419.     inputFilterSizeForBarChart.classList.toggle('hide-element');
420.     labelNumberOfHashFunctionsForBarChart.classList.toggle('hide-element');
421.     inputNumberOfHashForBarChart.classList.toggle('hide-element');
422.     labelNumberOfElementsToAdd.classList.toggle('hide-element');
423.     inputNumberOfElementsToAddForBarChart.classList.toggle('hide-element');
424.     labelNumberOfElementsToCheck.classList.toggle('hide-element');
425.     inputNumberOfElementsToCheckForBarChart.classList.toggle('hide-element');
426.     generateBloomFilterForBarChartButton.classList.toggle('hide-element');
427.     labelExplainingBarChart.classList.toggle('hide-element');
428.
429.     if (document.getElementById('barChart').style.display == "none") {
430.         document.getElementById('barChart').style.display = "block";
431.     } else {
432.         document.getElementById('barChart').style.display = "none";
433.     }
434. }
435. }
436.
437. /**
438.  * Generating Bloom Filter.
439.  * Adding elements to it, checking availability.
440.  * @param {number} filterSize
441.  * @param {number} numberOfFunctions
442.  * @param {number} numberOfElementsToAdd
443.  * @param {number} numberOfElementsToCheck
444.  */
445. function getresultsFromBloomFilterForBarChart(filterSizeBarChart, numberOfFunctions, numberOfElementsToAdd,
    numberOfElementsToCheck) {
446.     let hashFunctions = (new UniversalHashFunctions(numberOfFunctions)).generateFunctions();
447.     let minRangeForNumber = document.getElementById('input-left-border-number-for-bar-chart').value;
448.     let maxRangeForNumber = document.getElementById('input-right-border-number-for-bar-chart').value;
449.     let addedElements = [];
450.     let filter = [];
451.     for (let index = 0; index < filterSizeBarChart; index++) {
452.         filter[index] = 0;
453.     }
454.     for (let indexNewElement = 0; indexNewElement < numberOfElementsToAdd; ++indexNewElement) {
455.         let newElement = Math.floor(Math.random() * (maxRangeForNumber - minRangeForNumber) +
            minRangeForNumber);
456.         if (!addedElements.includes(newElement)) {
457.             addedElements.push(newElement);

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
458.     }
459.     for (let indexHashFunction = 0; indexHashFunction < hashFunctions.length; ++indexHashFunction) {
460.         let currentFunction = hashFunctions[indexHashFunction];
461.         let indexForFilter = currentFunction(newElement)[0] % filterSizeBarChart;
462.         filter[indexForFilter] = 1;
463.     }
464. }
465. let countFalsePositive = 0, countNegative = 0, countPositive = 0;
466. let definatelyNotInFilter = false;
467. for (let indexCheckelement = 0; indexCheckelement < numberOfElementsToCheck; indexCheckelement++) {
468.     let newElement = Math.floor(Math.random() * (maxRangeForNumber - minRangeForNumber) +
        minRangeForNumber);
469.     definatelyNotInFilter = false;
470.     for (let indexHashFunction = 0; indexHashFunction < hashFunctions.length; ++indexHashFunction) {
471.         let currentFunction = hashFunctions[indexHashFunction];
472.         let indexForFilter = currentFunction(newElement)[0] % filterSizeBarChart;
473.         if (filter[indexForFilter] == 0) {
474.             definatelyNotInFilter = true;
475.             ++countNegative;
476.             break;
477.         }
478.     }
479.     if (!definatelyNotInFilter) {
480.         if (addedElements.includes(newElement)) {
481.             ++countPositive;
482.         } else {
483.             ++countFalsePositive;
484.         }
485.     }
486. }
487. return [countFalsePositive, countNegative, countPositive];
488. }
489.
490. /**
491.  * Creating bar chart for showing results of generation.
492.  */
493. function createBarChart(numberOfFalsePositive, numberOfNegative, numberOfPositive) {
494.     let canvas = document.createElement('canvas');
495.     canvas.style.position = "absolute";
496.     canvas.id = "barChart";
497.     canvas.style.display = "none";
498.
499.     document.getElementById('zone-for-bar-chart').appendChild(canvas);
500.     canvas.parentNode.style.bottom = '10px';
501.     canvas.parentNode.style.width = '380px';
502.     canvas.parentNode.style.top = '10px';
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
503. canvas.style.right = '10px';
504. canvas.style.bottom = '10px';
505.
506. const data = {
507.   labels: ['false positive', 'negative', 'positive'],
508.   datasets: [{
509.     label: "results",
510.     data: [numberOfFalsePositive, numberOfNegative, numberOfPositive],
511.     backgroundColor: [
512.       'rgba(255, 99, 132, 0.2)',
513.       'rgba(54, 162, 235, 0.2)',
514.       'rgba(255, 206, 86, 0.2)',
515.     ],
516.     borderColor: [
517.       'rgba(255, 99, 132, 1)',
518.       'rgba(54, 162, 235, 1)',
519.       'rgba(255, 206, 86, 1)',
520.     ],
521.     borderWidth: 1
522.   }]
523. };
524.
525. const config = {
526.   type: 'bar',
527.   data,
528.   options: {
529.     plugins: {
530.       legend: {
531.         display: false
532.       }
533.     },
534.     scales: {
535.       y: {
536.         beginAtZero: true
537.       }
538.     }
539.   }
540. };
541.
542. const barChart = new Chart(
543.   document.getElementById('barChart'),
544.   config
545. );
546. return barChart;
547. }
548.
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
549. /**
550.  * Creates the elements needed to change the parameters of the function.
551.  */
552. function createChangeFunctionParametersElements() {
553.     // Shows/hides fields for changing function parameters.
554.     let changeFunctionParametersButton = document.createElement('button');
555.     changeFunctionParametersButton.className = 'change-function-parameters-button';
556.     changeFunctionParametersButton.id = 'change-function-parameters-button';
557.     changeFunctionParametersButton.innerHTML = "Change function parameters";
558.
559.     // The number of the function for which you want to change the parameters.
560.     let numberOfFunctionToChangeInput = document.createElement('input');
561.     numberOfFunctionToChangeInput.className = 'number-of-function-to-change-input';
562.     numberOfFunctionToChangeInput.id = 'number-of-function-to-change-input';
563.     numberOfFunctionToChangeInput.placeholder = "number of function";
564.     numberOfFunctionToChangeInput.classList.toggle('hide-element');
565.
566.     numberOfFunctionToChangeInput.onblur = () => {
567.         if (numberOfFunctionToChangeInput.value !== "" &&
568.             (isFloat(parseFloat(numberOfFunctionToChangeInput.value)) ||
569.              !Number.isInteger(parseInt(numberOfFunctionToChangeInput.value)) ||
570.              parseInt(numberOfFunctionToChangeInput.value) > numberOfHash ||
571.              parseInt(numberOfFunctionToChangeInput.value) < 1) && numberOfFunctionToChangeInput.style.borderColor !==
572.              "red") {
573.             numberOfFunctionToChangeInput.style.borderColor = "red";
574.         }
575.     };
576.
577.     numberOfFunctionToChangeInput.addEventListener('click', () => {
578.         if (numberOfFunctionToChangeInput.style.borderColor === "red") {
579.             numberOfFunctionToChangeInput.style.borderColor = "";
580.             numberOfFunctionToChangeInput.value = "";
581.         }
582.     });
583.
584.     // New parameter 'a' for function.
585.     let parameterAToChangeInput = document.createElement('input');
586.     parameterAToChangeInput.className = 'a-parameter-to-change-input';
587.     parameterAToChangeInput.id = 'a-parameter-to-change-input';
588.     parameterAToChangeInput.placeholder = "'a' parameter";
589.     parameterAToChangeInput.classList.toggle('hide-element');
590.
591.     parameterAToChangeInput.onblur = () => {
592.         if (parameterAToChangeInput.value !== "" && (isFloat(parseFloat(parameterAToChangeInput.value)) ||
593.              !Number.isInteger(parseInt(parameterAToChangeInput.value)) ||
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата


```
589.     String(parseInt(parameterAtoChangelInput.value)).length > 16) &&
    parameterAtoChangelInput.style.borderColor != "red") {
590.     parameterAtoChangelInput.style.borderColor = "red";
591. }
592. }
593.
594. parameterAtoChangelInput.addEventListener('click', () => {
595.     if (parameterAtoChangelInput.style.borderColor == "red") {
596.         parameterAtoChangelInput.style.borderColor = "";
597.         parameterAtoChangelInput.value = "";
598.     }
599. });
600.
601. // New parameter 'b' for function.
602. let parameterBtoChangelInput = document.createElement('input');
603. parameterBtoChangelInput.className = 'b-parameter-to-change-input';
604. parameterBtoChangelInput.id = 'b-parameter-to-change-input';
605. parameterBtoChangelInput.placeholder = "'b' parameter";
606. parameterBtoChangelInput.classList.toggle('hide-element');
607.
608. parameterBtoChangelInput.onblur = () => {
609.     if (parameterBtoChangelInput.value != "" && (isFloat(parseFloat(parameterBtoChangelInput.value)) ||
    !Number.isInteger(parseInt(parameterBtoChangelInput.value))) ||
610.     String(parseInt(parameterBtoChangelInput.value)).length > 16) &&
    parameterBtoChangelInput.style.borderColor != "red") {
611.         parameterBtoChangelInput.style.borderColor = "red";
612.     }
613. }
614.
615. parameterBtoChangelInput.addEventListener('click', () => {
616.     if (parameterBtoChangelInput.style.borderColor == "red") {
617.         parameterBtoChangelInput.style.borderColor = "";
618.         parameterBtoChangelInput.value = "";
619.     }
620. });
621.
622. // New parameter 'p' for function.
623. let parameterPtoChangelInput = document.createElement('input');
624. parameterPtoChangelInput.className = 'p-parameter-to-change-input';
625. parameterPtoChangelInput.id = 'p-parameter-to-change-input';
626. parameterPtoChangelInput.placeholder = "'p' parameter";
627. parameterPtoChangelInput.classList.toggle('hide-element');
628.
629. parameterPtoChangelInput.onblur = () => {
630.     if (parameterPtoChangelInput.value != "" && (isFloat(parseFloat(parameterPtoChangelInput.value)) ||
    !Number.isInteger(parseInt(parameterPtoChangelInput.value))) ||
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
631.     String(parseInt(parameterPToChangelInput.value)).length > 16) &&
        parameterPToChangelInput.style.borderColor != "red") {
632.     parameterPToChangelInput.style.borderColor = "red";
633.     }
634. }
635.
636. parameterPToChangelInput.addEventListener('click', () => {
637.     if (parameterPToChangelInput.style.borderColor == "red") {
638.         parameterPToChangelInput.style.borderColor = "";
639.         parameterPToChangelInput.value = "";
640.     }
641. });
642.
643. // When you click this button all parameters of function will change.
644. let acceptChangesToFunctionButton = document.createElement('button');
645. acceptChangesToFunctionButton.className = 'accept-changes-to-function-button';
646. acceptChangesToFunctionButton.id = 'accept-changes-to-function-button';
647. acceptChangesToFunctionButton.innerHTML = "Accept changes";
648. acceptChangesToFunctionButton.classList.toggle('hide-element');
649.
650. // When you click this button all parameters of function will change randomly.
651. let changeParametersOfFunctionRandomly = document.createElement('button');
652. changeParametersOfFunctionRandomly.className = 'change-parameters-of-function-randomly';
653. changeParametersOfFunctionRandomly.id = 'change-parameters-of-function-randomly';
654. changeParametersOfFunctionRandomly.innerHTML = "Change randomly";
655. changeParametersOfFunctionRandomly.classList.toggle('hide-element');
656.
657. changeFunctionParametersButton.onclick = () => {
658.     numberOfFunctionToChangelInput.classList.toggle('hide-element');
659.     parameterAToChangelInput.classList.toggle('hide-element');
660.     parameterBToChangelInput.classList.toggle('hide-element');
661.     parameterPToChangelInput.classList.toggle('hide-element');
662.     acceptChangesToFunctionButton.classList.toggle('hide-element');
663.     changeParametersOfFunctionRandomly.classList.toggle('hide-element');
664. }
665.
666. document.getElementById('playground-main').appendChild(changeFunctionParametersButton);
667. document.getElementById('playground-main').appendChild(numberOfFunctionToChangelInput);
668. document.getElementById('playground-main').appendChild(parameterAToChangelInput);
669. document.getElementById('playground-main').appendChild(parameterBToChangelInput);
670. document.getElementById('playground-main').appendChild(parameterPToChangelInput);
671. document.getElementById('playground-main').appendChild(acceptChangesToFunctionButton);
672. document.getElementById('playground-main').appendChild(changeParametersOfFunctionRandomly);
673. }
674.
675. /**
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
676. * Creating a button to show items added to a filter
677. */
678. function createShowAddedElementsButton() {
679.     let showAddedElementsButton = document.createElement('button');
680.     showAddedElementsButton.className = 'show-added-elements-button';
681.     showAddedElementsButton.id = 'show-added-elements-button';
682.     showAddedElementsButton.innerHTML = "Added elements";
683.
684.     let textAreaWithListOfAddedElements = document.createElement('textarea');
685.     textAreaWithListOfAddedElements.className = 'text-area-with-list-of-added-elements';
686.     textAreaWithListOfAddedElements.id = 'text-area-with-list-of-added-elements';
687.     textAreaWithListOfAddedElements.classList.add('hide-element');
688.
689.     showAddedElementsButton.onclick = () => textAreaWithListOfAddedElements.classList.toggle('hide-element');
690.     document.getElementById('playground-main').appendChild(showAddedElementsButton);
691.     document.getElementById('playground-main').appendChild(textAreaWithListOfAddedElements);
692. }
693.
694. function changeTextInPseudocodeAfterCheckingElementAvailability() {
695.     document.getElementById('pseudocode-window').innerHTML =
696.     "<span class='type-pseudocode'>for</span> <span class='bracket-pseudocode'>(</span><span class='type-
        pseudocode'>int</span> <span class='variable-pseudocode'>index</span> <span class='bracket-
        pseudocode'>=</span> <span class='numbers-pseudocode'>0</span><span class='bracket-
        pseudocode'>;</span> <span class='variable-pseudocode'>index</span> <span class='type-
        pseudocode'>></span> <span class='variable-pseudocode'>numberOfFunctions</span><span class='bracket-
        pseudocode'>;</span> <span class='bracket-pseudocode'>++</span><span class='variable-
        pseudocode'>index</span><span class='bracket-pseudocode'>)</span> <span class='bracket-
        pseudocode'>{</span><span><br>" +
697.     "<span class='pseudocode-check-element-get-filter-index' id='pseudocode-check-element-get-filter-
        index'><span class='underscore-pseudocode' id='underscore-pseudocode'>---></span><span class='type-
        pseudocode'>int</span> <span class='variable-pseudocode'>indexInFilter</span> <span class='bracket-
        pseudocode'>=</span> <span class='variable-pseudocode'>listOfFunctions</span><span class='bracket-
        pseudocode'>[</span><span class='variable-pseudocode'>index</span><span class='bracket-
        pseudocode'>]</span><span class='bracket-pseudocode'>(</span><span class='variable-
        pseudocode'>element</span><span class='bracket-pseudocode'>)</span><span class='bracket-
        pseudocode'>;</span><span><br>" +
698.     "<span class='pseudocode-check-element-check-cell-is-zero' id='pseudocode-check-element-check-cell-is-
        zero'><span class='underscore-pseudocode' id='underscore-pseudocode'>---></span><span class='type-
        pseudocode'>if</span> <span class='bracket-pseudocode'>(</span><span class='variable-
        pseudocode'>filter</span><span class='bracket-pseudocode'>[</span><span class='variable-
        pseudocode'>indexInFilter</span><span class='bracket-pseudocode'>]</span> <span class='bracket-
        pseudocode'>=</span><span class='numbers-pseudocode'>0</span><span class='bracket-
        pseudocode'>)</span> <span class='bracket-pseudocode'>{</span><span><br>" +
699.     "<span class='pseudocode-check-element-cell-is-zero' id='pseudocode-check-element-cell-is-zero'><span
        class='underscore-pseudocode' id='underscore-pseudocode'>-----></span><span class='variable-
        pseudocode'>gotZero</span> <span class='bracket-pseudocode'>=</span> <span class='type-
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

pseudocode'>true</span><span class='bracket-pseudocode'>;</span><span class='bracket-
pseudocode'>></span></span><br>" +
700. " <span class='pseudocode-check-element-check-gotZero-is-true' id='pseudocode-check-element-check-
gotZero-is-true'><span class='type-pseudocode'>if</span><span class='bracket-pseudocode'>(</span><span
class='variable-pseudocode'>gotZero</span><span class='bracket-pseudocode'>==</span><span class='type-
pseudocode'>true</span><span class='bracket-pseudocode'>)</span></span></span><span class='pseudocode-check-
element-print-not-in-filter' id='pseudocode-check-element-print-not-in-filter'><span class='numbers-
pseudocode'>print</span><span class='bracket-pseudocode'>(</span><span class='pseudocode-check-
element-string-element'>'definitely not in filter'</span><span class='bracket-
pseudocode'>)</span></span><br>" +
701. " <span class='type-pseudocode'>else</span><span class='bracket-pseudocode'>{</span><br>" +
702. " <span id='pseudocode-check-element-check-if-list-includes-element'><span class='underscore-pseudocode'
id='underscore-pseudocode'>---></span><span class='type-pseudocode'>if</span><span class='bracket-
pseudocode'>(!</span><span class='variable-pseudocode'>listOfAddedElements</span><span class='bracket-
pseudocode'>.</span><span class='numbers-pseudocode'>includes</span><span class='bracket-
pseudocode'>(</span><span class='variable-pseudocode'>element</span><span class='bracket-
pseudocode'>))</span></span><br>" +
703. " <span id='pseudocode-check-element-print-false'><span class='underscore-pseudocode' id='underscore-
pseudocode'>-----></span><span class='numbers-pseudocode'>print</span><span class='pseudocode-check-
element-string-element'>'false positive result'</span><span class='bracket-
pseudocode'>)</span></span><br>" +
704. " <span id='pseudocode-check-element-print-cant-say'><span class='underscore-pseudocode' id='underscore-
pseudocode'>---></span><span class='type-pseudocode'>else</span><span class='numbers-
pseudocode'>print</span><span class='bracket-pseudocode'>(</span><span class='pseudocode-check-
element-string-element'>'can't say for sure'</span><span class='bracket-pseudocode'>)</span></span></span>";
705.}
706.
707. /**
708. * Check if there is an element in the filter.
709. * @param {Array} hashFunctions list of hash functions
710. * @param {number} filterSize number of cells in filter
711. * @param {Array} addedElementsList list of elements that were added to filter
712. */
713. function checkElementAvailability(hashFunctions, addedElementsList) {
714.     visualizationInProgress = true;
715.
716.     let speedVisualization = document.getElementById('input-range-speed').value;
717.
718.     changeTextInPseudocodeAfterCheckingElementAvailability();
719.     ctx = document.getElementById("canvasArrows").getContext("2d");
720.
721.     let inputAddElement = document.getElementById('add-element-input');
722.     let value = parseInt(inputAddElement.value);
723.     inputAddElement.value = "";
724.     let filterCells = document.getElementById('filter-array-div').childNodes;
725.     // True if all result values were in filter.

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
726. let wasInFilter = true;
727. let indexForFunctionDivs = 0;
728. let timerIdGlobalInterval = setInterval(() => {
729.     if (indexForFunctionDivs == hashFunctions.length) {
730.         document.getElementById('pseudocode-check-element-check-gotZero-is-true').classList.add('highlighted-
pseudocode');
731.         if (wasInFilter) {
732.             document.getElementById('pseudocode-check-element-check-if-list-includes-
element').classList.add('highlighted-pseudocode');
733.             // If the element was never added but the result is positive.
734.             if (!addedElementsList.includes(value)) {
735.                 document.getElementById('pseudocode-check-element-print-false').classList.add('highlighted-
pseudocode');
736.                 document.getElementById('checking-element-result').innerHTML = "<b>False positive</b> result";
737.             } else {
738.                 document.getElementById('pseudocode-check-element-print-cant-say').classList.add('highlighted-
pseudocode');
739.                 document.getElementById('checking-element-result').innerHTML = "Can't say for sure";
740.             }
741.         } else {
742.             document.getElementById('pseudocode-check-element-print-not-in-filter').classList.add('highlighted-
pseudocode');
743.             document.getElementById('checking-element-result').innerHTML = "Definitely <b>not</b> in the filter";
744.         }
745.         setTimeout(() => {
746.             document.getElementById('checking-element-result').innerHTML = "";
747.             document.getElementById('pseudocode-check-element-check-gotZero-is-
true').classList.remove('highlighted-pseudocode');
748.             document.getElementById('pseudocode-check-element-print-not-in-filter').classList.remove('highlighted-
pseudocode');
749.             document.getElementById('pseudocode-check-element-check-if-list-includes-
element').classList.remove('highlighted-pseudocode');
750.             document.getElementById('pseudocode-check-element-print-false').classList.remove('highlighted-
pseudocode');
751.             document.getElementById('pseudocode-check-element-print-cant-say').classList.remove('highlighted-
pseudocode');
752.         }, 2000);
753.         setTimeout(() => {
754.             visualizationInProgress = false;
755.             speedVisualization / 2);
756.             clearInterval(timerIdGlobalInterval);
757.         }
758. let func = hashFunctions[indexForFunctionDivs];
759. let filterIndex = func(value)[0] % filterSize;
760.
761. if (filterIndex < 0) {
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

762.     filterIndex *= -1;
763.     filterIndex = filterSize - filterIndex;
764. }
765.
766. // Coordinates for the value input field.
767. let rectangleInputCoordinates = inputAddElement.getBoundingClientRect();
768. let fromInputStartX = rectangleInputCoordinates.left + inputAddElement.clientWidth + 10;
769. let fromInputStartY = rectangleInputCoordinates.top + inputAddElement.height / 2;
770.
771. let currentHashFunctionDiv = document.getElementById('hash-functions-list-
div').childNodes[indexForFunctionDivs];
772. // Coordinates for the div that the arrow goes to.
773. let rectangleFunctionDivCoordinates = currentHashFunctionDiv.getBoundingClientRect();
774. let fromInputEndX = rectangleFunctionDivCoordinates.left - 20;
775. let fromInputEndY = rectangleFunctionDivCoordinates.top - 5 + currentHashFunctionDiv.clientHeight / 2;
776.
777. ctx.beginPath();
778. drawAnArrow(ctx, fromInputStartX, fromInputStartY, fromInputEndX, fromInputEndY);
779. ctx.stroke();
780.
781. let innerTextOfFunctionDiv = currentHashFunctionDiv.firstChild.innerHTML;
782. currentHashFunctionDiv.firstChild.innerHTML = `${filterIndex}`;
783. currentHashFunctionDiv.classList.add('highlighted');
784.
785. document.getElementById('pseudocode-check-element-get-filter-index').classList.add('highlighted-
pseudocode');
786.
787. let indexForFilterCells = 0;
788. // Run on each function in array of has functions.
789. for (let cell of filterCells) {
790.     if (indexForFilterCells == filterIndex) {
791.         let fromFunctionDivStartX = rectangleFunctionDivCoordinates.left + currentHashFunctionDiv.clientWidth
+ 5;
792.         let fromFunctionDivStartY = rectangleFunctionDivCoordinates.top + currentHashFunctionDiv.clientHeight
/ 2;
793.
794.         let rectangleFilterCellCoordinates = cell.getBoundingClientRect();
795.         let fromFunctionDivEndX = rectangleFilterCellCoordinates.left - 20;
796.         let fromFunctionDivEndY = rectangleFilterCellCoordinates.top - 5 + cell.clientHeight / 2;
797.
798.         setTimeout(() => {
799.             ctx.beginPath();
800.             drawAnArrow(ctx, fromFunctionDivStartX, fromFunctionDivStartY, fromFunctionDivEndX,
fromFunctionDivEndY);
801.             ctx.stroke();
802.             currentHashFunctionDiv.firstChild.innerHTML = innerTextOfFunctionDiv;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

803.         currentHashFunctionDiv.classList.remove('highlighted');
804.
805.         document.getElementById('pseudocode-check-element-get-filter-index').classList.remove('highlighted-
pseudocode');
806.         document.getElementById('pseudocode-check-element-check-cell-is-zero').classList.add('highlighted-
pseudocode');
807.
808.         if (cell.firstChild.innerHTML == '0') {
809.             cell.classList.add('red-background');
810.             document.getElementById('pseudocode-check-element-cell-is-zero').classList.add('highlighted-
pseudocode');
811.             setTimeout(() => {
812.                 cell.classList.remove('red-background');
813.                 ctx.clearRect(0, 0, 1650, 942);
814.                 document.getElementById('pseudocode-check-element-cell-is-
zero').classList.remove('highlighted-pseudocode');
815.                 document.getElementById('pseudocode-check-element-check-cell-is-
zero').classList.remove('highlighted-pseudocode');
816.             }, speedVisualization / 3);
817.             wasInFilter = false;
818.         } else {
819.             cell.classList.add('green-background');
820.             setTimeout(() => {
821.                 cell.classList.remove('green-background');
822.                 ctx.clearRect(0, 0, 1650, 942);
823.                 document.getElementById('pseudocode-check-element-check-cell-is-
zero').classList.remove('highlighted-pseudocode');
824.             }, speedVisualization / 3);
825.         }
826.     }, speedVisualization / 2);
827.     break;
828. }
829. ++indexForFilterCells;
830. }
831. ++indexForFunctionDivs;
832. }, speedVisualization);
833. }
834.
835. /**
836.  * Creat button that checks if element is in filter.
837.  */
838. function createButtonCheckElementAvailability() {
839.     let buttonCheckAvailability = document.createElement('button');
840.     buttonCheckAvailability.className = 'check-element-availability-button';
841.     buttonCheckAvailability.id = 'check-element-availability-button';
842.     buttonCheckAvailability.innerHTML = 'Check availability';

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

843. document.getElementById('playground-main').appendChild(buttonCheckAvailability);
844.
845. let resultOfChecking = document.createElement('span');
846. resultOfChecking.className = 'checking-element-result';
847. resultOfChecking.id = 'checking-element-result';
848. document.getElementById('playground-main').appendChild(resultOfChecking);
849. }
850.
851. function changeTextInPseudocodeAfterAddingElement() {
852.   document.getElementById('pseudocode-window').innerHTML =
853.     "<span class='type-pseudocode'>int</span> <span class='variable-pseudocode'>element</span> <span
      class='type-pseudocode'>=</span> <span class='numbers-pseudocode'>read</span> <span class='bracket-
      pseudocode'>)</span> <span class='bracket-pseudocode'>;</span> <br> <span class='type-
      pseudocode'>for</span> <span class='bracket-pseudocode'>(</span> <span class='type-
      pseudocode'>int</span> <span class='variable-pseudocode'>index</span> <span class='type-
      pseudocode'>=</span> <span class='numbers-pseudocode'>0</span> <span class='bracket-
      pseudocode'>;</span> <span class='variable-pseudocode'>index</span> <span class='type-
      pseudocode'><</span> <span class='variable-pseudocode'>numberOfFunctions</span> <span class='bracket-
      pseudocode'>;</span> <span class='bracket-pseudocode'>++</span> <span class='variable-
      pseudocode'>index</span> <span class='bracket-pseudocode'>)</span> <span class='bracket-
      pseudocode'>{</span> <br>" +
854.     "<span class='pseudocode-add-element-get-index' id='pseudocode-add-element-get-index'> <span
      class='underscore-pseudocode' id='underscore-pseudocode'>---> </span> <span class='type-
      pseudocode'>int</span> <span class='variable-pseudocode'>indexInFilter</span> <span class='type-
      pseudocode'>=</span> <span class='variable-pseudocode'>listOfFunctions</span> <span class='bracket-
      pseudocode'>[</span> <span class='variable-pseudocode'>index</span> <span class='bracket-
      pseudocode'>]</span> <span class='bracket-pseudocode'>(</span> <span class='variable-
      pseudocode'>element</span> <span class='bracket-pseudocode'>)</span> <span class='bracket-
      pseudocode'>;</span> </span> </span>" +
855.     "<br> <span class='pseudocode-add-element-assign-filter' id='pseudocode-add-element-assign-filter'> <span
      class='underscore-pseudocode' id='underscore-pseudocode'>---> </span> <span class='variable-
      pseudocode'>filter</span> <span class='bracket-pseudocode'>[</span> <span class='variable-
      pseudocode'>indexInFilter</span> <span class='bracket-pseudocode'>]</span> <span class='type-
      pseudocode'>=</span> <span class='numbers-pseudocode'>1</span> <span class='bracket-
      pseudocode'>;</span> </span> </span> <br> <span class='bracket-pseudocode'>}</span> </span>"
856. }
857.
858. /**
859.  * When adding an item, change the values in the cells.
860.  * If the element was added to the filter, change the value
861.  * of the corresponding cells by one.
862.  * @param {Array} hashFunctions list of hash functions
863.  */
864. function changeValuesInCellsAfterAddingElement(hashFunctions) {
865.   visualizationInProgress = true;
866.

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата


```

867. let speedVisualization = document.getElementById('input-range-speed').value;
868.
869. changeTextInPseudocodeAfterAddingElement();
870. ctx = document.getElementById("canvasArrows").getContext("2d");
871.
872. let inputAddElement = document.getElementById('add-element-input');
873. let value = parseInt(inputAddElement.value);
874. inputAddElement.value = "";
875. let filterCells = document.getElementById('filter-array-div').childNodes;
876.
877. let indexForFunctionDivs = 0;
878. let timerId = setInterval(() => {
879.     let func = hashFunctions[indexForFunctionDivs];
880.
881.     let filterIndex = func(value)[0] % filterSize;
882.
883.     if (filterIndex < 0) {
884.         filterIndex *= -1;
885.         filterIndex = filterSize - filterIndex;
886.     }
887.
888.     // Coordinates for the value input field.
889.     let rectangleInputCoordinates = inputAddElement.getBoundingClientRect();
890.     let fromInputStartX = rectangleInputCoordinates.left + inputAddElement.clientWidth + 10;
891.     let fromInputStartY = rectangleInputCoordinates.top + inputAddElement.height / 2;
892.
893.     let currentHashFunctionDiv = document.getElementById('hash-functions-list-
div').childNodes[indexForFunctionDivs];
894.     // Coordinates for the div that the arrow goes to.
895.     let rectangleFunctionDivCoordinates = currentHashFunctionDiv.getBoundingClientRect();
896.     let fromInputEndX = rectangleFunctionDivCoordinates.left - 20;
897.     let fromInputEndY = rectangleFunctionDivCoordinates.top - 5 + currentHashFunctionDiv.clientHeight / 2;
898.
899.     ctx.beginPath();
900.     drawAnArrow(ctx, fromInputStartX, fromInputStartY, fromInputEndX, fromInputEndY);
901.     ctx.stroke();
902.
903.     let innerTextOfFunctionDiv = currentHashFunctionDiv.firstChild.innerHTML;
904.     currentHashFunctionDiv.firstChild.innerHTML = `${filterIndex}`;
905.     currentHashFunctionDiv.classList.add('highlighted');
906.
907.     // When finding index in filter highlight the code.
908.     document.getElementById('pseudocode-add-element-get-index').classList.add('highlighted-pseudocode');
909.
910.     let indexForCell = 0;
911.     // Run on each cell in filter.

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

912.     for (let cell of filterCells) {
913.         // Change cell value to '1' in cell, which index is 'filterIndex'.
914.         if (indexForCell == filterIndex) {
915.             let fromFunctionDivStartX = rectangleFunctionDivCoordinates.left +
currentHashFunctionDiv.clientWidth + 5;
916.             let fromFunctionDivStartY = rectangleFunctionDivCoordinates.top +
currentHashFunctionDiv.clientHeight / 2;
917.
918.             let rectangleFilterCellCoordinates = cell.getBoundingClientRect();
919.             let fromFunctionDivEndX = rectangleFilterCellCoordinates.left - 20;
920.             let fromFunctionDivEndY = rectangleFilterCellCoordinates.top - 5 + cell.clientHeight / 2;
921.
922.             setTimeout(() => {
923.                 ctx.beginPath();
924.                 drawAnArrow(ctx, fromFunctionDivStartX, fromFunctionDivStartY, fromFunctionDivEndX,
fromFunctionDivEndY);
925.                 ctx.stroke();
926.                 currentHashFunctionDiv.firstChild.innerHTML = innerTextOfFunctionDiv;
927.                 currentHashFunctionDiv.classList.remove('highlighted');
928.
929.                 document.getElementById('pseudocode-add-element-get-index').classList.remove('highlighted-
pseudocode');
930.                 document.getElementById('pseudocode-add-element-assign-filter').classList.add('highlighted-
pseudocode');
931.
932.                 cell.firstChild.innerHTML = '1';
933.                 cell.classList.add('highlighted');
934.                 setTimeout(() => {
935.                     ctx.clearRect(0, 0, 1650, 942);
936.                     document.getElementById('pseudocode-add-element-assign-filter').classList.remove('highlighted-
pseudocode');
937.                     }, speedVisualization / 3);
938.                     }, speedVisualization / 2);
939.                     break;
940.                 }
941.                 ++indexForCell;
942.             }
943.             ++indexForFunctionDivs;
944.             if (indexForFunctionDivs == hashFunctions.length) {
945.                 setTimeout(() => {
946.                     visualizationInProgress = false;
947.                     }, speedVisualization / 2);
948.                     clearInterval(timerId);
949.                 }
950.             }, speedVisualization);
951. }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
952.
953. /**
954.  * Adding a field to the playground for entering an element
955.  * that will be added to the filter.
956.  */
957. function createInputAddElement() {
958.   let inputAddElement = document.createElement('input');
959.   inputAddElement.className = 'add-element-input';
960.   inputAddElement.id = 'add-element-input';
961.   inputAddElement.placeholder = 'your number';
962.
963.   inputAddElement.onblur = () => {
964.     if (inputAddElement.value != "" && (isFloat(parseFloat(inputAddElement.value)) ||
965.       !Number.isInteger(parseInt(inputAddElement.value)) ||
966.       String(parseInt(inputAddElement.value)).length > 16) && inputAddElement.style.borderColor != "red") {
967.       inputAddElement.style.borderColor = "red";
968.     }
969.   }
970.   inputAddElement.addEventListener('click', () => {
971.     if (inputAddElement.style.borderColor == "red") {
972.       inputAddElement.style.borderColor = "";
973.       inputAddElement.value = "";
974.     }
975.   });
976.
977.   document.getElementById('playground-main').appendChild(inputAddElement);
978. }
979.
980. /**
981.  * Adding a button to the playground, after clicking on
982.  * which the element will be added to the filter.
983.  */
984. function createButtonAddElement() {
985.   let buttonAddElement = document.createElement('button');
986.   buttonAddElement.className = 'add-element-button';
987.   buttonAddElement.id = 'add-element-button';
988.   buttonAddElement.innerHTML = 'Add element';
989.   document.getElementById('playground-main').appendChild(buttonAddElement);
990. }
991.
992. /**
993.  * Creating a wrapper that will store a list of divs,
994.  * each of which is associated with a specific hash function.
995.  */
996. function createHashFunctions() {
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

997. let hashFunctionsList = document.createElement('div');
998. hashFunctionsList.classList.add('hash-functions-list-div');
999. hashFunctionsList.id = 'hash-functions-list-div';
1000. document.getElementById('playground-main').appendChild(hashFunctionsList);
1001. }
1002.
1003. /**
1004.  * Creating a wrapper that will store a list of divs,
1005.  * each of which is associated with a specific bloom filter cell.
1006.  */
1007. function createFilterArray() {
1008.     let filterArrayDiv = document.createElement('div');
1009.     filterArrayDiv.classList.add('filter-array-div');
1010.     filterArrayDiv.id = 'filter-array-div';
1011.     document.getElementById('playground-main').appendChild(filterArrayDiv);
1012. }
1013.
1014. /**
1015.  * Draws divs that match bloom filter cells.
1016.  */
1017. function buildBloomFilter() {
1018.     // Div that contains cells for Bloom filter
1019.     let filterArrayDiv = document.getElementById('filter-array-div');
1020.     // Every new cell would be next this position.
1021.     let cellShift = (filterArrayDiv.clientHeight - 50 * filterSize) / 2;
1022.     for (let index = 0; index < filterSize; ++index) {
1023.         // Creating new cell for filter.
1024.         let newDivCell = document.createElement('div');
1025.         newDivCell.classList.add('filter-cell-div');
1026.         newDivCell.style.top = cellShift + 'px';
1027.         // Text in the cell.
1028.         let innerTextDivCell = document.createElement('div');
1029.         innerTextDivCell.className = 'filter-cell-div-inner-text';
1030.         innerTextDivCell.innerHTML = '0';
1031.         newDivCell.appendChild(innerTextDivCell);
1032.         // Adding cell into filter.
1033.         filterArrayDiv.appendChild(newDivCell);
1034.         cellShift += 50;
1035.     }
1036. }
1037.
1038. /**
1039.  * A sequence of div blocks is drawn, each of which is responsible for a hash function.
1040.  * @param {number} numberOfHash amount of hash functions
1041.  * @param {Array} hashFunctions array of hash functions
1042.  */

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

1043.     function buildListOfHashFunctions(numberOfHash, hashFunctions) {
1044.         let hashFunctionsList = document.getElementById('hash-functions-list-div');
1045.         hashFunctionsList.innerHTML = "";
1046.         // Every new hash function would be next this position.
1047.         let hashFunctionShift = ((hashFunctionsList.clientHeight - 50 * numberOfHash) / 2);
1048.         for (let index = 0; index < numberOfHash; ++index) {
1049.             // Creating new hash function.
1050.             let newHashFunction = document.createElement('div');
1051.             newHashFunction.classList.add('hash-function-div');
1052.             newHashFunction.style.top = hashFunctionShift + 'px';
1053.             newHashFunction.dataToggle = "tooltip";
1054.             let [a, b, p] = [hashFunctions[index](1)[1], hashFunctions[index](1)[2], hashFunctions[index](1)[3]];
1055.             newHashFunction.title = `(${a}x + ${b} mod (${p})) mod (${filterSize})`;
1056.             // Text in the cell.
1057.             let innerTextHashFunction = document.createElement('div');
1058.             innerTextHashFunction.className = 'hash-function-div-inner-text';
1059.             innerTextHashFunction.innerHTML = 'k' + '<sub>' + (index + 1);
1060.             newHashFunction.appendChild(innerTextHashFunction);
1061.             hashFunctionsList.appendChild(newHashFunction);
1062.             hashFunctionShift += 50;
1063.         }
1064.     }

```

1.1.2 Файл canvas.js

```

function drawAnArrow(context, fromx, fromy, tox, toy) {
    context.lineWidth = 2;
    context.strokeStyle = 'khaki';
    var headlen = 7; // length of head in pixels
    var dx = tox - fromx;
    var dy = toy - fromy;
    var angle = Math.atan2(dy, dx);
    context.moveTo(fromx, fromy);
    context.lineTo(tox, toy);
    context.lineTo(tox - headlen * Math.cos(angle - Math.PI / 6), toy - headlen * Math.sin(angle - Math.PI / 6));
    context.moveTo(tox, toy);
    context.lineTo(tox - headlen * Math.cos(angle + Math.PI / 6), toy - headlen * Math.sin(angle + Math.PI / 6));
}

```

1.1.3 Файл footer.js

```

/**
 * Fill footer with elements.
 */
function fillTheFooterBloomFilter() {

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

// Element in which some settings of footer are situated.
let elementsPlaygroundFooter = document.getElementById('elements-playground-footer');

document.getElementById('playground-footer-title').style.display = "none";

addInputRangeForSpeedVisualization(elementsPlaygroundFooter);
createLabelHintAboutHashFunctions(elementsPlaygroundFooter);
}

function createLabelHintAboutHashFunctions(elementsPlaygroundFooter) {
  let labelFooterHint = document.createElement('label');
  labelFooterHint.innerHTML = "When hovering over a hash function, you can see how the final value is calculated!";
  labelFooterHint.style.position = "absolute";
  labelFooterHint.style.left = "10px";
  labelFooterHint.style.top = "100px";
  labelFooterHint.style.color = "white";
  elementsPlaygroundFooter.appendChild(labelFooterHint);
}

/**
 * Adding the input range that can regulate speed of visualization.
 * @param {HTMLDivElement} elementsPlaygroundFooter element in which some settings of footer are situated
 */
function addInputRangeForSpeedVisualization(elementsPlaygroundFooter) {
  let shellForInputRangeSpeed = document.createElement('div');
  shellForInputRangeSpeed.className = 'shell-input-range-speed';
  shellForInputRangeSpeed.id = 'shell-input-range-speed';

  let inputRangeSpeedVisual = document.createElement('input');
  inputRangeSpeedVisual.className = 'input-range-speed';
  inputRangeSpeedVisual.id = 'input-range-speed';
  inputRangeSpeedVisual.type = 'range';
  inputRangeSpeedVisual.min = "500";
  inputRangeSpeedVisual.max = "3000";
  inputRangeSpeedVisual.step = "any";

  shellForInputRangeSpeed.innerHTML = '<i>Speed visualization:</i>';

  let subIndeciesForSpeedRange = document.createElement('div');
  subIndeciesForSpeedRange.className = 'indecies-for-speed-range';
  subIndeciesForSpeedRange.id = 'indecies-for-speed-range';
  subIndeciesForSpeedRange.innerHTML = "faster<span style='color:grey'>-----</span>slower";

  shellForInputRangeSpeed.appendChild(inputRangeSpeedVisual);
  elementsPlaygroundFooter.appendChild(subIndeciesForSpeedRange);
  elementsPlaygroundFooter.appendChild(shellForInputRangeSpeed);
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1.1.4 Файл header.js

```
/**
 * Building an interface for the Bloom filter header.
 */
function fillTheHeaderBloomFilter() {
    // DIV containing all header elements.
    let elementsPlaygroundCap = document.getElementById('elements-playground-cap');

    elementsPlaygroundCap.innerHTML = "";

    // Creating settings for input filter size.
    let inputFilterSize = createInputFilterSize(elementsPlaygroundCap);

    // Creating settings for input number of hesh.
    let inputNumberOfHash = createInputNumberOfHash(elementsPlaygroundCap);

    // Add button for building the filter.
    let buttonBuildFilter = createButtonBuldingTheFilter(elementsPlaygroundCap);

    labelRestrictionsForFilterAndFunctions(elementsPlaygroundCap);

    addEventWhenButtonBuildFilterWasClicked(buttonBuildFilter, inputFilterSize, inputNumberOfHash);
}

/**
 * Label that shows restriction for filter size.
 * @param {object} elementsPlaygroundCap div containing header elements
 */
function labelRestrictionsForFilterAndFunctions(elementsPlaygroundCap) {
    let divLabelRestrictionsForFilterSize = document.createElement('div');
    divLabelRestrictionsForFilterSize.innerHTML = "<i>restrictions: 1 ≤ value ≤ 13</i><br><br><i>restrictions: 1 ≤ value ≤ 13</i>";
    divLabelRestrictionsForFilterSize.style.position = "absolute";
    divLabelRestrictionsForFilterSize.style.left = "300px";
    divLabelRestrictionsForFilterSize.style.top = "9px";
    divLabelRestrictionsForFilterSize.style.color = "rgb(48, 184, 246, 0.95)";
    elementsPlaygroundCap.appendChild(divLabelRestrictionsForFilterSize);
}

/**
 * Creating an element that serves as an input field
 * for the element to be added to the bloom filter.
 */
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

* @param {object} elementsPlaygroundCap div containing header elements
* @returns field for input nwe value
*/
function createInputFilterSize(elementsPlaygroundCap) {
  let divFilterSize = document.createElement('div');
  divFilterSize.innerHTML = '<i>Filter size: </i>';
  divFilterSize.className = 'div-filter-size';
  let inputFilterSize = document.createElement('input');
  inputFilterSize.className = "input-filter-size-header";
  inputFilterSize.style = "background: white";
  inputFilterSize.style.border = "1.5px solid black";
  inputFilterSize.placeholder = 'your number';

  divFilterSize.appendChild(inputFilterSize);
  elementsPlaygroundCap.appendChild(divFilterSize);
  return inputFilterSize;
}

/**
* Creating an element that serves as an input field
* for number of hash functions.
* @param {object} elementsPlaygroundCap div containing header elements
* @returns field for input new number of hash functions
*/
function createInputNumberOfHash(elementsPlaygroundCap) {
  let divNumberOfHesh = document.createElement('div');
  divNumberOfHesh.innerHTML = '<i>Number of hash: </i>';
  divNumberOfHesh.className = 'div-hesh-number';
  let inputNumberOfHesh = document.createElement('input');
  inputNumberOfHesh.style = "background-color: white";
  inputNumberOfHesh.style.border = "1.5px solid black";
  inputNumberOfHesh.placeholder = 'your number';
  divNumberOfHesh.appendChild(inputNumberOfHesh);
  elementsPlaygroundCap.appendChild(divNumberOfHesh);
  return inputNumberOfHesh;
}

/**
* Creating an element that serves as a button
* that builds interface for filter.
* @param {object} elementsPlaygroundCap div containing header elements
* @returns Button that builds filter.
*/
function createButtonBuldingTheFilter(elementsPlaygroundCap) {
  let buttonBuildFilter = document.createElement('button');
  buttonBuildFilter.innerHTML = "BUILD FILTER";

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата


```
buttonBuildFilter.className = 'button-build-filter';
buttonBuildFilter.id = 'button-build-filter';
elementsPlaygroundCap.appendChild(buttonBuildFilter);
return buttonBuildFilter;
}
```

1.1.5 Файл js.js

```
/**
 * As the data structure Bloom filter was chosen.
 */
document.getElementById("bloom-filter-carousel-element").onclick = function() {
    document.getElementById('bloom-filter-description').style.display = "block";
    document.getElementById("bloom-filter-carousel-element").style.border = "1px solid white";
    let elementInCarouselBloomFilter = document.getElementById("bloom-filter-carousel-element");
    if (elementInCarouselBloomFilter.hasAttribute("in-use")) {
        return;
    }
    elementInCarouselBloomFilter.setAttribute('in-use', 'bloom-filter-carousel-element');
    drawInterface();
}

/**
 * Building an interface in the cap of playground for the user.
 */
function drawInterface() {
    switch(document.getElementById("bloom-filter-carousel-element").getAttribute('in-use')) {
        case 'bloom-filter-carousel-element':
            fillTheHeaderBloomFilter();
            fillTheFooterBloomFilter();
            break;
    }
}
```

1.1.6 Файл universal_hash_functions.js

```
class UniversalHashFunctions {
    constructor(numberOfFunctions) {
        this.numberOfFunctions = numberOfFunctions;
    }
}
/**
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
* Generates array of universal hash functions.
* @returns Array of hash functions.
*/
generateFunctions() {
  let p = 115249;
  let functions = [];
  for (let index = 0; index < this.numberOfFunctions; ++index) {
    let a = Math.floor(Math.random() * (p - 1) + 1);
    let b = Math.floor(Math.random() * p);
    let newFunction = value => {
      return [(a * value + b) % p, a, b, p];
    }
    functions[index] = newFunction;
  }
  return functions;
}
```

1.2 Файлы расширения .html

1.2.1 Файл vads.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>VADS</title>
  <link rel="stylesheet" href="styles/main.css">
  <link rel="stylesheet" href="styles/bloom.css">
  <link rel="stylesheet" href="styles/footer.css">
  <link rel="stylesheet" href="styles/header.css">
</head>
<body>
  <canvas id="canvasArrows" width="1650" height="942"></canvas>
  <div class="main-menu">
    <div class="title-block" id="title-block">
      VADS
    </div>
    <div class="elements-carousel">
      <div class="bloom-filter-carousel-element" id="bloom-filter-carousel-element" data-toggle="tooltip">
        <div class="bloom-filter-picture" id="bloom-filter-picture">
          
        </div>
        <div class="bloom-filter-header-carousel" id="bloom-filter-header-carousel">
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

<b>BLOOM FILTER</b><br>
Probabilistic data structure
</div>
</div>
<div class="bloom-filter-description" id="bloom-filter-description" style="display: none;">
  <h3>Description</h3>
  The Bloom filter is a bitmap of <b>m</b> bits. At the very beginning, the user defines <b>k</b> independent hash
  functions, each of which is mapped to a filter, that is, to a bit array.
  If the function mapped to the i-th cell, the i-th bit becomes equal to <b>one</b>. <br>
  Optimal number of hash functions:  $(m/n)\ln 2 \approx 0.6931(m/n)$ .
  <h3>Use cases</h3>
  Bloom filter allows you to determine that the element is <b>not</b> included in the set. For example, if a database is
  accessed, then the filter can say with <b>100%</b> probability that the element is <b>not</b> there.
  <h3>Output cases</h3>
  There are <b>three</b> possible outputs: the element is <b>definitely not</b> in the filter, the element <b>may
  be</b> present, or a <b>>false positive</b> (occurs when the filter says that the element may be present, but it was not
  added).
  <h3>Probabilities</h3>
  False positive probability:  $(1 - e^{-kn/m})^k$ 
  False alarm probability:  $2^{-k} \approx 0.6185^{m/n}$ , where k - number of functions, m - filter size, n
  - number of added elements.
</div>
</div>
</div>
<div class="playground">
  <div class="elements-playground-cap" id="elements-playground-cap">
    <div class="playground-cap-title" id="playground-cap-title">input data</div>
  </div>
  <div class="playground-main" id="playground-main">
    <div class="playground-main-title" id="playground-main-title">playground</div>
  </div>
  <div class="elements-playground-footer" id="elements-playground-footer">
    <div class="playground-footer-title" id="playground-footer-title">settings</div>
    <div class="pseudocode-window" id="pseudocode-window">
      <div class="pseudocode-title" id="pseudocode-title">CODE</div>
    </div>
  </div>
</div>
</div>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script src="scripts/canvas.js"></script>
<script src="scripts/js.js"></script>
<script src="scripts/bloom.js"></script>
<script src="scripts/universal_hash_functions.js"></script>
<script src="scripts/header.js"></script>
<script src="scripts/footer.js"></script>
</body>

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

</html>

1.3 Файлы расширения .css

1.3.1 Файл bloom.css

```
.bloom-filter-carousel-element {  
  border: 1px solid black;  
  position: absolute;  
  left: 5px;  
  right: 5px;  
  top: 5px;  
  height: 100px;  
  background-color: rgb(129, 107, 81);  
}  
  
.bloom-filter-carousel-element:hover {  
  background-color: rgb(170, 165, 158);  
}  
  
.bloom-filter-carousel-element:hover {  
  cursor: pointer;  
}  
  
.div-filter-size {  
  position: absolute;  
  left: 10px;  
  top: 10px;  
}  
  
.div-hesh-number {  
  position: absolute;  
  left: 10px;  
  bottom: 10px;  
}  
  
.filter-array-div {  
  position: absolute;  
  left: 50%;  
  width: 50px;  
  top: 0px;  
  bottom: 0px;  
}  
  
.hash-functions-list-div {  
  position: absolute;
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
left: 30%;
width: 50px;
top: 0px;
bottom: 0px;
}

.filter-cell-div {
border: 1px solid black;
width: 50px;
height: 50px;
position: absolute;
text-align: center;
background-color: rgb(134, 121, 103);
}

.hash-function-div {
border: 1px solid black;
width: 50px;
height: 50px;
position: absolute;
text-align: center;
background-color: rgb(134, 121, 103);
}

.filter-cell-div-inner-text {
position: absolute;
font-size: 25px;
left: 50%;
margin-left: -5px;
top: 50%;
margin-top: -16px;
}

.hash-function-div-inner-text {
position: absolute;
font-size: 23px;
left: 50%;
margin-left: -7px;
top: 50%;
margin-top: -16px;
}

.add-element-button {
position: absolute;
left: 10px;
top: 50%;
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
width: 105.5px;
height: 35px;
margin-top: -52px;
background-color: rgb(117, 90, 54);
font-size: 12px;
}

.add-element-button:hover {
  cursor: pointer;
}

.check-element-availability-button {
  position: absolute;
  left: 10px;
  bottom: 50%;
  width: 105.5px;
  height: 35px;
  margin-bottom: -46px;
  background-color: rgb(117, 90, 54);
  font-size: 12px;
}

.check-element-availability-button:hover {
  cursor: pointer;
}

.checking-element-result {
  position: absolute;
  left: 10px;
  bottom: 50%;
  height: 30px;
  margin-bottom: -90px;
  color: rgb(29, 221, 29);
}

.add-element-input {
  position: absolute;
  left: 10px;
  top: 50%;
  width: 100px;
  margin-top: -13px;
  background-color: white;
  border: 1.5px solid black;
}

.highlighted {
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
background-color: rgb(190, 178, 160);
}

.red-background {
background-color: red;
}

.green-background {
background-color: rgb(3, 245, 3);
}

.bloom-filter-picture {
border: 1px solid black;
position: absolute;
top: 5px;
left: 5px;
bottom: 5px;
right: 70%;
float: left;
}

.bloom-filter-header-carousel {
border: 1px solid black;
position: absolute;
top: 5px;
float: right;
right: 5px;
bottom: 5px;
left: 31%;
text-align: center;
line-height: 40px;
}

.show-added-elements-button {
position: absolute;
float: left;
left: 10px;
top: 208px;
width: 200px;
height: 30px;
font-size: 13px;
background-color: rgb(117, 90, 54);
}

.show-added-elements-button:hover {
cursor: pointer;
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
}

.change-function-parameters-button {
  position: absolute;
  float: left;
  left: 10px;
  bottom: 215px;
  width: 200px;
  height: 30px;
  font-size: 13px;
  background-color: rgb(117, 90, 54);
}

.change-function-parameters-button:hover {
  cursor: pointer;
}

.text-area-with-list-of-added-elements {
  position: absolute;
  float: left;
  left: 10px;
  top: 10px;
  width: 194px;
  height: 175px;
  font-size: 17px;
  border: 1px solid #000;
  bottom: 10px;
  background-color: rgb(93, 80, 62, 0.3);
  resize: none;
  pointer-events: none;
}

.hide-element {
  display: none;
}

.number-of-function-to-change-input {
  position: absolute;
  float: left;
  left: 10px;
  bottom: 183px;
  width: 194px;
  height: 20px;
  font-size: 13px;
  background-color: white;
  border: 1.5px solid black;
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата


```
}  
  
.a-parameter-to-change-input {  
  position: absolute;  
  float: left;  
  left: 10px;  
  bottom: 150px;  
  width: 194px;  
  height: 20px;  
  font-size: 13px;  
  background-color: white;  
  border: 1.5px solid black;  
}  
  
.b-parameter-to-change-input {  
  position: absolute;  
  float: right;  
  left: 10px;  
  bottom: 118px;  
  width: 194px;  
  height: 20px;  
  font-size: 13px;  
  background-color: white;  
  border: 1.5px solid black;  
}  
  
.p-parameter-to-change-input {  
  position: absolute;  
  float: left;  
  left: 10px;  
  bottom: 85px;  
  width: 194px;  
  height: 20px;  
  font-size: 13px;  
  background-color: white;  
  border: 1.5px solid black;  
}  
  
.accept-changes-to-function-button {  
  position: absolute;  
  float: left;  
  left: 10px;  
  bottom: 45px;  
  width: 200px;  
  height: 30px;  
  font-size: 13px;
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

background-color: rgb(98, 80, 57);
}

.accespt-changes-to-function-button:hover {
  cursor: pointer;
}

.change-parameters-of-function-randomly {
  position: absolute;
  float: left;
  left: 10px;
  bottom: 10px;
  width: 200px;
  height: 30px;
  font-size: 13px;
  background-color: rgb(98, 80, 57);
}

.change-parameters-of-function-randomly:hover {
  cursor: pointer;
}

.zone-for-bar-chart {
  float: right;
  position: absolute;
  right: 0px;
  top: 0px;
  bottom: 0px;
  left: 70%;
}

.show-zone-bar-chart-button {
  position: absolute;
  left: 0px;
  top: 0px;
  right: 0px;
  height: 30px;
  background-color: rgb(117, 90, 54);
}

.show-zone-bar-chart-button:hover {
  cursor: pointer;
}

.input-filter-size-for-bar-chart {
  position: absolute;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
top: 53px;
width: 373px;
background-color: white;
border: 1.5px solid black;
}

.input-number-of-hash-functions-for-bar-chart {
  position: absolute;
  top: 98px;
  width: 373px;
  background-color: white;
  border: 1.5px solid black;
}

.input-number-of-elements-to-add-for-bar-chart {
  position: absolute;
  top: 142px;
  width: 373px;
  background-color: white;
  border: 1.5px solid black;
}

.input-number-of-elements-to-check-for-bar-chart {
  position: absolute;
  top: 186px;
  width: 373px;
  background-color: white;
  border: 1.5px solid black;
}

.input-left-border-number-for-bar-chart {
  position: absolute;
  top: 230px;
  width: 180px;
  background-color: white;
  border: 1.5px solid black;
}

.input-right-border-number-for-bar-chart {
  position: absolute;
  right: 0px;
  top: 230px;
  width: 180px;
  background-color: white;
  border: 1.5px solid black;
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
.button-to-build-filter-bar-chart {
  position: absolute;
  top: 260px;
  height: 30px;
  width: 379px;
  background-color: rgb(98, 80, 57);
}

.button-to-build-filter-bar-chart:hover {
  cursor: pointer;
}

.underscore-pseudocode {
  color: rgb(29, 28, 28);
}

.type-pseudocode {
  color: rgb(55, 107, 248);
}

.bracket-pseudocode {
  color: white;
}

.variable-pseudocode {
  color: rgb(0, 174, 255);
}

.numbers-pseudocode {
  color: rgba(156, 156, 45, 0.972);
}

.highlighted-pseudocode {
  background-color: wheat;
}

.pseudocode-check-element-string-element {
  color: rgb(247, 188, 80);
}

.playground-main-title {
  position: absolute;
  font-size: 30px;
  left: 50%;
  top: 50%;
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

margin-top: -30px;
margin-left: -55px;
font-weight: 600;
}

.button-clear-filter {
  position: absolute;
  top: 50%;
  right: 470px;
  height: 30px;
  width: 100px;
  margin-top: -14px;
  background-color: rgb(98, 80, 57);
}

.button-clear-filter:hover {
  cursor: pointer;
}

```

1.3.2 Файл footer.css

```

.shell-input-range-speed {
  position: absolute;
  left: 5px;
  top: 5px;
}

.input-range-speed {
  margin: 10px;
}

.input-range-speed:hover {
  cursor: pointer;
}

.pseudocode-title {
  position: absolute;
  left: 50%;
  top: 50%;
  font-size: 30px;
  margin-top: -20px;
  margin-left: -35px;
  color: darkgoldenrod;
}

.playground-footer-title {

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

position: absolute;
top: 50%;
left: 33%;
font-size: 30px;
margin-top: -15px;
font-weight: 600;
}

.indicies-for-speed-range {
position: absolute;
bottom: 140px;
left: 143px;
font-size: 13px;
}

```

1.3.3 Файл header.css

```

.button-build-filter {
position: absolute;
font-size: 20px;
top: 20px;
right: 10px;
bottom: 20px;
width: 200px;
background-color: rgb(13, 119, 3);
}

.button-build-filter:hover {
cursor: pointer;
}

.playground-cap-title {
position: absolute;
left: 50%;
top: 50%;
font-size: 30px;
font-weight: 600;
margin-left: -46px;
margin-top: -20px;
}

.input-filter-size-header {
position: absolute;
top: 0px;
left: 111px;
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
.button-show-description-filter {
  position: absolute;
  width: 200px;
  height: 35px;
  right: 250px;
  font-size: 12px;
  top: 28px;
  background-color: rgb(122, 120, 120);
}

.button-show-description-filter:hover {
  cursor: pointer;
}
```

1.3.4 Файл main.css

```
body {
  background: linear-gradient(to bottom right, rgb(138, 110, 47), rgb(155, 115, 122));
}

.main-menu {
  border: 1px solid black;
  position: absolute;
  left: 10px;
  top: 10px;
  bottom: 10px;
  right: 80%;
  float: left;
}

.playground {
  border: 1px solid black;
  position: absolute;
  right: 10px;
  top: 10px;
  bottom: 10px;
  left: 21%;
  float: right;
}

.title-block {
  border: 1px solid black;
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
position: absolute;
left: 0px;
right: 0px;
top: 0px;
bottom: 90%;
font-size: 50px;
text-align: center;
line-height: 90px;
background-color: rgb(134, 108, 76);
}

.elements-carousel {
border: 1px solid black;
position: absolute;
left: 0px;
right: 0px;
top: 15%;
bottom: 0px;
}

.elements-playground-cap {
border: 1px solid black;
position: absolute;
float: left;
left: 0px;
right: 0px;
top: 0px;
bottom: 90%;
}

.elements-playground-footer {
border: 1px solid black;
position: absolute;
left: 0px;
right: 0px;
top: 80%;
bottom: 0px;
}

.pseudocode-window {
border: 1px solid white;
position: absolute;
left: 70%;
right: 0px;
top: 0px;
bottom: 0px;
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата


```

background-color: rgb(29, 28, 28);
padding-left: 10px;
padding-top: 10px;
color: white;
}

.playground-main {
border: 1px solid black;
position: absolute;
left: 0px;
right: 0px;
top: 10%;
bottom: 20%;
}

.bloom-filter-description {
position: absolute;
bottom: 10px;
left: 10px;
right: 10px;
top: 15%;
}

```

ПРИЛОЖЕНИЕ 1

Список использованной литературы

- 1) ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 2) ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 3) ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 4) ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 5) ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 7) ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 8) ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 9) ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.V
- 10) VisuAlgo. [Электронный ресурс]//URL: <https://visualgo.net/ru> (Дата обращения: 03.02.2022, режим доступа: свободный).
- 11) Vamonos. [Электронный ресурс]//URL: <https://rosulek.github.io/vamonos/> (Дата обращения: 03.02.2022, режим доступа: свободный).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата