УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по практическому заданию №1

по предмету «Метрология, стандартизация и сертификация (в информационных технологиях»

Тема работы: Метрики Холстеда

Выполнил:

Крутько А.А.

Заяц А. Р.

гр. 251004

Проверила:

Болтак С.В.

Минск 2024

1. **Код TypeScript:**

const gridSize = 9;  
  
document.addEventListener('DOMContentLoaded', () => {  
 const solveButton = document.getElementById('solve-btn');  
 solveButton.addEventListener('click', solveSudoku);  
  
 const clearButton = document.getElementById('clear-btn');  
 clearButton.addEventListener('click', clearGrid);  
  
 const sudokuGrid = document.getElementById('sudoku-grid');  
  
 for (let row = 0; row < gridSize; row += 1) {  
 const newRow = document.createElement('tr');  
 for (let col = 0; col < gridSize; col += 1) {  
 const cell = document.createElement('td');  
 const input = document.createElement('input');  
 input.type = 'number';  
 input.className = 'cell';  
 input.id = `cell-${row}-${col}`;  
 cell.appendChild(input);  
 newRow.appendChild(cell);  
 }  
 sudokuGrid?.appendChild(newRow);  
 }  
});  
  
async function solveSudoku() {  
 const clearButton = document.getElementById('clear-btn') as HTMLButtonElement;  
 clearButton.disabled = true;  
 const sudokuArray : number[][] = [];  
  
 for (let row = 0; row < gridSize; row += 1) {  
 sudokuArray[row] = [];  
 for (let col = 0; col < gridSize; col += 1) {  
 const cellId = `cell-${row}-${col}`;  
 const cell = document.getElementById(cellId) as HTMLInputElement;  
 const cellValue = cell.value;  
 sudokuArray[row][col] = cellValue !== '' ? parseInt(cellValue) : 0;  
  
 if (sudokuArray[row][col] !== 0) {  
 cell.classList.add('user-input');  
 }  
 }  
 }  
  
 if (solveSudokuHelper(sudokuArray, 0, 0)) {  
 alert('Solution exists for the given Sudoku.');  
 } else {  
 alert('No solution exists for the given Sudoku.');  
 }  
 clearButton.disabled = false;  
}  
  
function solveSudokuHelper(board : number[][], row : number, col : number) : boolean {  
 if (col === 9) {  
 col = 0;  
 row += 1;  
 if (row === 9) {  
 return true;  
 }  
 }  
  
 if (board[row][col] !== 0) {  
 return solveSudokuHelper(board, row, col + 1);  
 }  
  
 for (let c = 1; c <= 9; c += 1) {  
 if (isValid(board, row, col, c)) {  
 board[row][col] = c;  
  
 if (solveSudokuHelper(board, row, col + 1)) {  
 return true;  
 }  
  
 board[row][col] = 0;  
 }  
 }  
  
 return false;  
  
 function isValid(board : number[][], row : number, col : number, c : number) : boolean {  
   
 for (let i = 0; i < 9; i += 1) {  
 if (board[row][i] === c) {  
 return false;  
 }  
 if (board[i][col] === c) {  
 return false;  
 }  
 }  
  
 let startRow = Math.floor(row / 3) \* 3;  
 let startCol = Math.floor(col / 3) \* 3;  
 for (let i = startRow; i < startRow + 3; i += 1) {  
 for (let j = startCol; j < startCol + 3; j += 1) {  
 if (board[i][j] === c) {  
 return false;  
 }  
 }  
 }  
  
 return true;  
 }  
}

function clearGrid() {  
 for (let row = 0; row < gridSize; row += 1) {  
 for (let col = 0; col < gridSize; col += 1) {  
 const cellId = `cell-${row}-${col}`;  
 const cell = document.getElementById(cellId) as HTMLInputElement;  
 cell.value = '';  
 cell.classList.remove('user-input', 'solved');  
 }  
 }  
}

1. **Ручной расчёт:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***j*** | **Оператор** | ***f1j*** | ***i*** | **Операнд** | ***f2i*** |
| 1 | => | 1 | 1 | 'DOMContentLoaded' | 1 |
| 2 | <= | 1 | 2 | 'solve-btn' | 1 |
| 3 | solveSudoku | 1 | 3 | solveSudoku | 1 |
| 4 | parseInt | 1 | 4 | clearGrid | 1 |
| 5 | add | 1 | 5 | 'sudoku-grid' | 1 |
| 6 | clearGrid | 1 | 6 | 'tr' | 1 |
| 7 | remove | 1 | 7 | 'td' | 1 |
| 8 | if ... else | 1 | 8 | 'input' | 1 |
| 9 | \* | 2 | 9 | type | 1 |
| 10 | / | 2 | 10 | 'number' | 1 |
| 11 | ? | 2 | 11 | className | 1 |
| 12 | alert | 2 | 12 | 'cell' | 1 |
| 13 | isValid | 2 | 13 | id | 1 |
| 14 | floor | 2 | 14 | HTMLButtonElement | 1 |
| 15 | as | 3 | 15 | 'Solution exists for the given Sudoku.' | 1 |
| 16 | !== | 3 | 16 | 'No solution exists for the given Sudoku.' | 1 |
| 17 | addEventListener | 3 | 17 | 'solved' | 1 |
| 18 | createElement | 3 | 18 | solveButton | 2 |
| 19 | appendChild | 3 | 19 | 'click' | 2 |
| 20 | + | 4 | 20 | 'clear-btn' | 2 |
| 21 | helper | 4 | 21 | sudokuGrid | 2 |
| 22 | === | 5 | 22 | disabled | 2 |
| 23 | getElementById | 6 | 23 | HTMLInputElement | 2 |
| 24 | return | 8 | 24 | value | 2 |
| 25 | < | 9 | 25 | '' | 2 |
| 26 | if | 9 | 26 | classList | 2 |
| 27 | for | 10 | 27 | 'user-input' | 2 |
| 28 | += | 11 | 28 | boolean | 2 |
| 29 | : | 11 | 29 | Math | 2 |
| 30 | , | 18 | 30 | newRow | 3 |
| 31 | . | 28 | 31 | `cell-${row}-${col}` | 3 |
| 32 | = | 37 | 32 | cellValue | 3 |
| 33 | [] | 50 | 33 | startRow | 3 |
| 34 | {} | 52 | 34 | startCol | 3 |
| 35 | ; | 66 | 35 | true | 4 |
| 36 | () | 102 | 36 | cellId | 4 |
| 37 | \r\n | 115 | 37 | j | 4 |
| 38 | пробел | 1063 | 38 | 9 | 5 |
|  |  |  | 39 | clearButton | 5 |
|  |  |  | 40 | input | 5 |
|  |  |  | 41 | sudokuArray | 5 |
|  |  |  | 42 | false | 5 |
|  |  |  | 43 | 3 | 6 |
|  |  |  | 44 | gridSize | 7 |
|  |  |  | 45 | number | 8 |
|  |  |  | 46 | cell | 9 |
|  |  |  | 47 | c | 9 |
|  |  |  | 48 | i | 9 |
|  |  |  | 49 | document | 10 |
|  |  |  | 50 | board | 11 |
|  |  |  | 51 | 0 | 14 |
|  |  |  | 52 | 1 | 14 |
|  |  |  | 53 | col | 23 |
|  |  |  | 54 | row | 24 |
| **η1 *= 38*** |  | ***N1 = 1643*** | **η2 *= 54*** |  | ***N2 = 237*** |

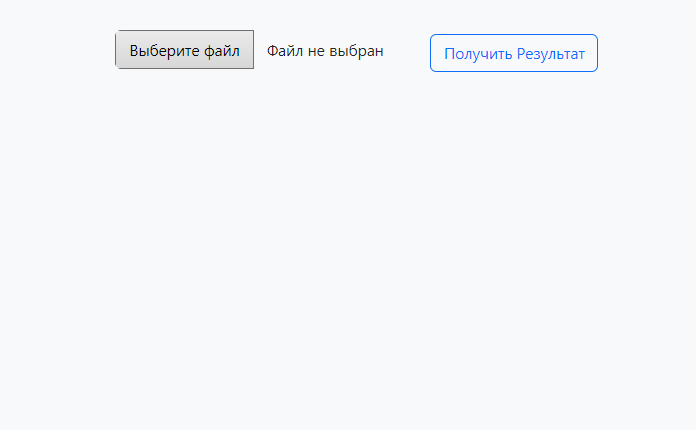
Словарь программы **η**=38 + 54 = 92.

Длина программы ***N*** = 237 + 1643 = 1880.

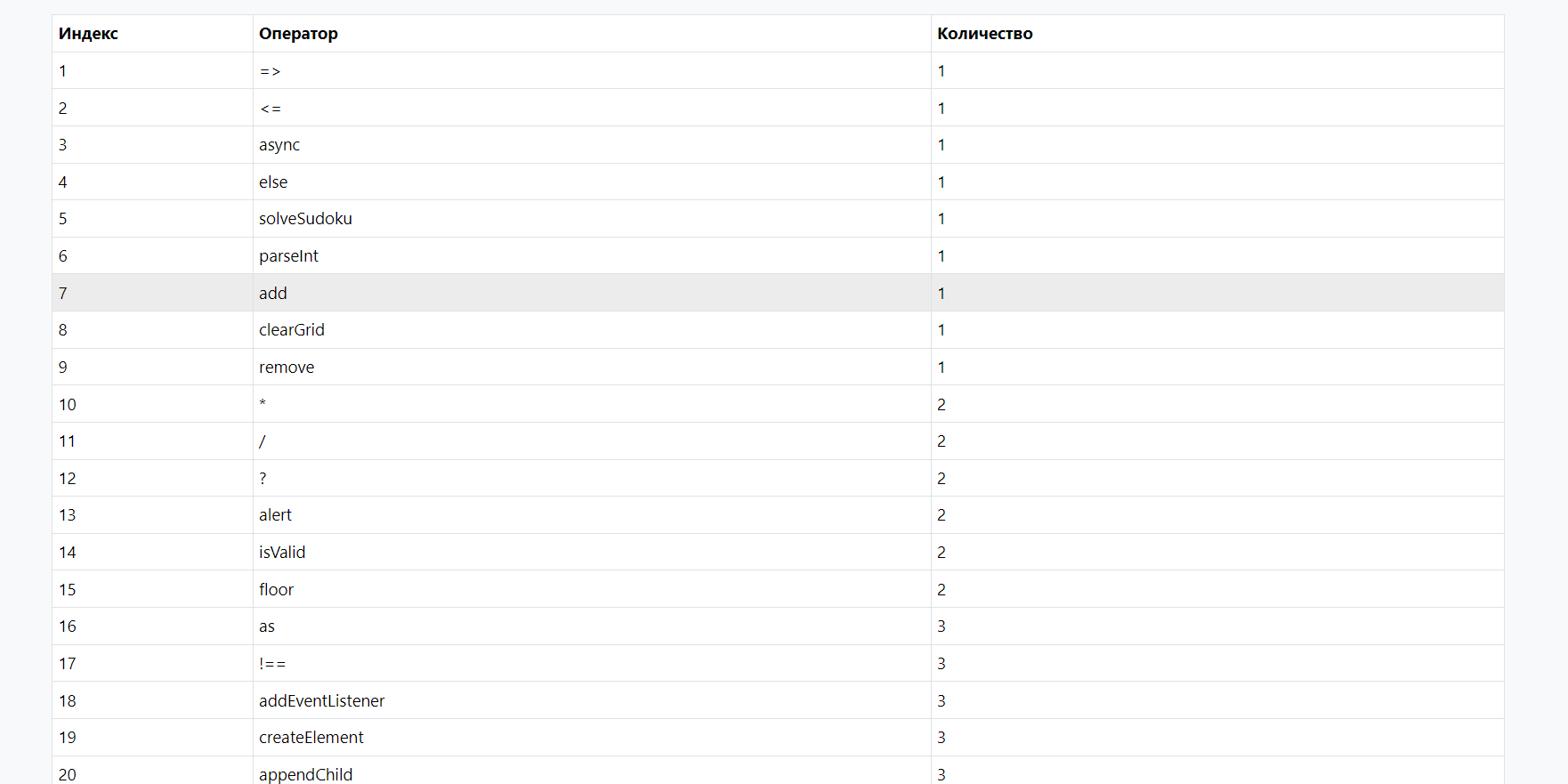
Объем программы ***V*** = 1880 \* = 12264,30;

1. **Скриншоты работы программы:**

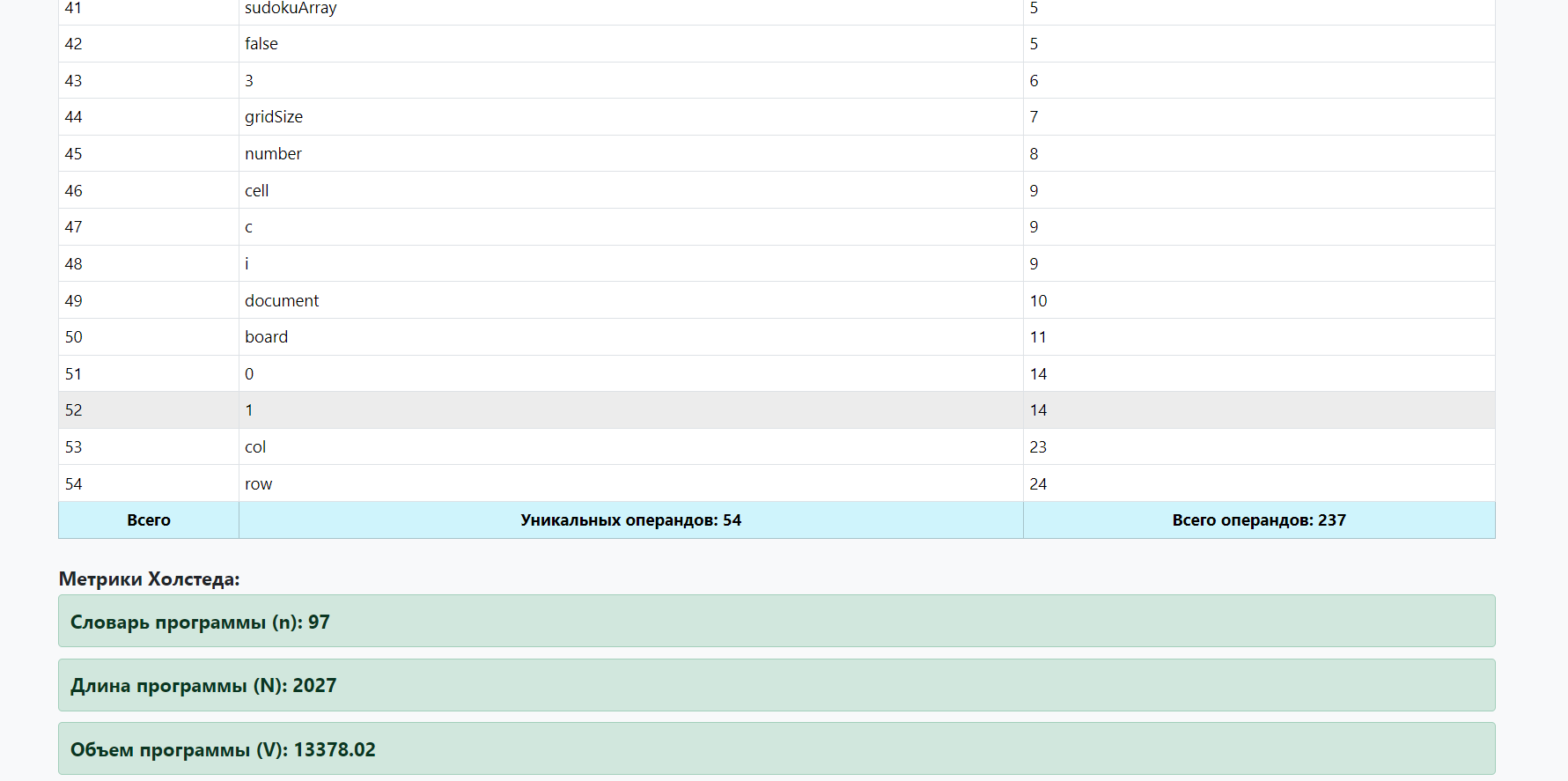
Программа выводит результаты в виде html-страницы, на которой есть кнопка, после нажатии которой генерируется результат на странице.



Изображение 1



Изображение 2



Изображение 3