

РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ ВОЗМОЖНОСТИ УРАВНИВАНИЯ ЛИНЕЙНО-УГЛОВЫХ ИЗМЕРЕНИЙ ДЛЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ GEOSLERK360

*Саранков Дмитрий Александрович, студент 4-го курса
кафедры «Геодезия и аэрокосмические геотехнологии»
Белорусский национальный технический университет, г. Минск
(Научный руководитель – Будо А.Ю., старший преподаватель)*

Современные цифровые технологии играют решающую роль в автоматизации вычислительных задач, проникая во все сферы человеческой деятельности. В эпоху цифровизации программное обеспечение становится незаменимым инструментом, позволяющим оптимизировать и упростить рабочие процессы, повысить производительность труда и сократить необходимость ручного вмешательства. Для разработки был выбран язык программирования JavaScript с использованием следующих технологий Node.js, Nest.js, TypeScript, Math.js, следующий стек был выбран исходя удобства создания веб-сервера и скорости его работы.

Общение с сервером происходит по способу REST API – «способ взаимодействия сайтов и веб-приложений с сервером», примеры запроса изображены на (Рис.1).

Тестирование происходит на локальном сервере.

```

X Заголовки Полезная нагрузка Предварительный просмотр Ответ Инициатор Время Файлы cookie
[{"element": "M15", "direction": 0, "angle": true, "stdDeviation": 2.121, "from": 1, "to": 5},...]
directions: [{element: "M15", direction: 0, angle: true, stdDeviation: 2.121, from: 1, to: 5},...]
  0: {element: "M15", direction: 0, angle: true, stdDeviation: 2.121, from: 1, to: 5}
  1: {element: "M12", direction: 1.6275461238784443, angle: true, stdDeviation: 2.121, from: 1, to: 2}
  2: {element: "M21", direction: 0, angle: true, stdDeviation: 2.121, from: 2, to: 1}
  3: {element: "M25", direction: 0.845751776922224, angle: true, stdDeviation: 2.121, from: 2, to: 5}
  4: {element: "M24", direction: 1.656026394395183, angle: true, stdDeviation: 2.121, from: 2, to: 4}
  5: {element: "M23", direction: 2.1674827422785006, angle: true, stdDeviation: 2.121, from: 2, to: 3}
points: [{point: 1, N: 246.0513, E: 50.2115, original: true},...]
  0: {point: 1, N: 246.0513, E: 50.2115, original: true}
  1: {point: 2, N: 113.4213, E: 79.9919, original: true}
  2: {point: 3, N: 40.8892, E: 263.8717, original: true}
  3: {point: 4, N: 139, E: 267, original: false}
  4: {point: 5, N: 291, E: 208, original: false}
sides: [{element: "1-5", length: 164.19294, linear: true, stdDeviation: null, from: 1, to: 5},...]
  0: {element: "1-5", length: 164.19294, linear: true, stdDeviation: null, from: 1, to: 5}
  1: {element: "4-5", length: 163.10721, linear: true, stdDeviation: null, from: 4, to: 5}
  2: {element: "3-4", length: 98.1249, linear: true, stdDeviation: null, from: 3, to: 4}
  3: {element: "2-4", length: 188.77869, linear: true, stdDeviation: null, from: 2, to: 4}
  4: {element: "2-5", length: 219.01293, linear: true, stdDeviation: null, from: 2, to: 5}
stdDeviation: {linear: {a: 2, b: 3}, angles: {mBeta: 3}}
  angles: {mBeta: 3}
  linear: {a: 2, b: 3}

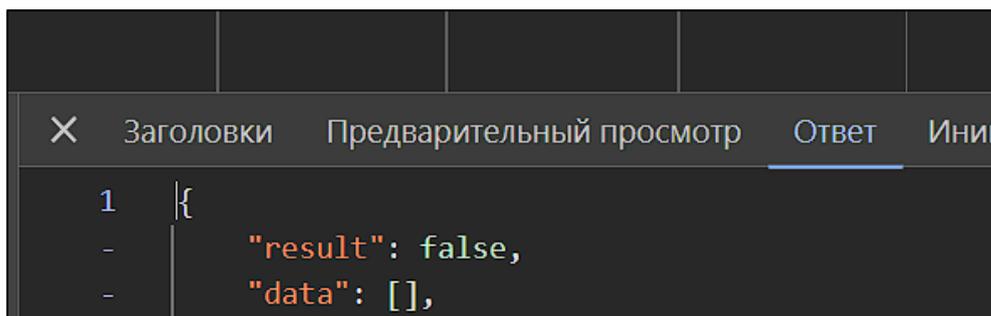
```

Рисунок 1 – Тело запроса, отправленное методом POST

Тело запроса представлено в формате JSON-объекта – беспорядочное множество пар «ключ: значение». На (Рис.1) можно увидеть JSON-объект, который представляет из себя входные данные для уравнивания линейно-угловой сети. Для реализации уравнивания был выбран параметрический способ.

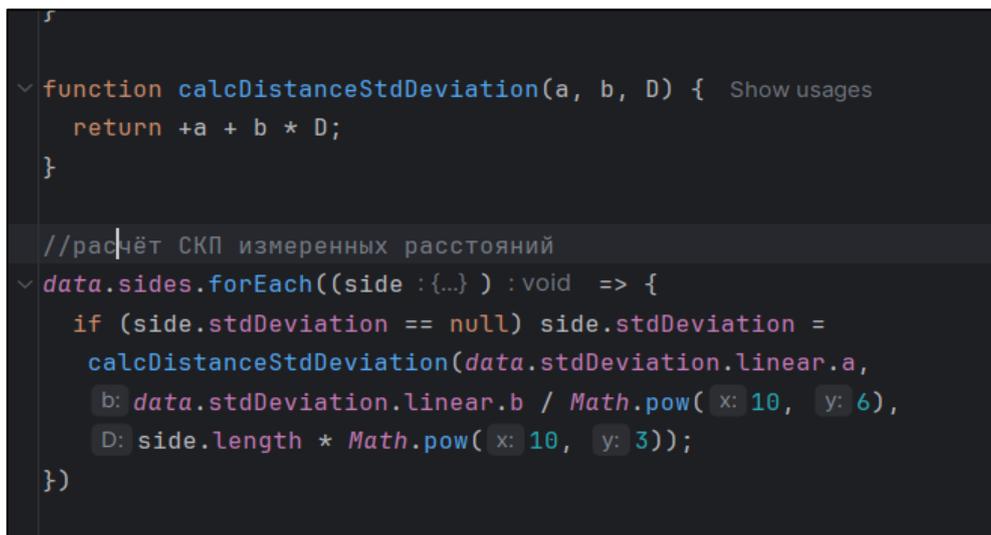
Алгоритм работы программы:

1. Обработка входных данных, в случае отсутствия обязательных параметров (координаты исходных пунктов, измеренные направления, расстояния и т.д.) сервер отправит ответ с параметром «result» равным «false», пример предоставлен на (Рис.2).
2. Обработка необязательных параметров к примеру, если не были указаны СКП измеренных расстояний, то программа выполнит их расчёт, пример предоставлен на (Рис.3).
3. Уравнивание сети, к примеру, во время которого вычисляется весовая матрица, коэффициенты параметрических уравнений, поправки в координаты, к конечному итогу уравненные значения измерений.
4. Проводиться оценка точности, расчёт эллипсов ошибок.
5. Отправка ответа клиенту, пример предоставлен на (Рис.4).



```
{
  "result": false,
  "data": []
}
```

Рисунок 2 – Ответ сервера в случае ошибок, некорректно введённых данных



```
function calcDistanceStdDeviation(a, b, D) {
  return +a + b * D;
}

//расчёт СКП измеренных расстояний
data.sides.forEach((side :{...}) :void => {
  if (side.stdDeviation == null) side.stdDeviation =
    calcDistanceStdDeviation(data.stdDeviation.linear.a,
      b: data.stdDeviation.linear.b / Math.pow(x: 10, y: 6),
      D: side.length * Math.pow(x: 10, y: 3));
})
```

Рисунок 3 – Пример обработки необязательных параметров

```

▼ accuracyAssessment: {...}
  ▶ adjustedMeasurementsStdDev: {angles: [{element: "M15", stdDev: 6.1623485}, {element: "M12", stdDev: 6.1623485},...],...}
  ▶ adjustedParametersStdDev: {point4: {N: 0.0051634, E: 0.0084645}, point5: {N: 0.0054837, E: 0.0061814}}
▼ adjustedMeasurements: {...}
  ▶ angles: [{element: "M15", measuredValue: "0°00'00.00000", correction: -1.6064, adjustedValue: 359.9995538},...]
  ▶ distances: [{element: "S1-5", measuredValue: 164.19294, correction: -0.0052, adjustedValue: 164.187697},...]
▼ corrections: {iterations: [{}, {...}],...}
  ▼ adjustedCoordinates: {point4: {N: 138.9602, E: 267.0355}, point5: {N: 291.0529, E: 208.1117}}
    ▶ point4: {N: 138.9602, E: 267.0355}
    ▶ point5: {N: 291.0529, E: 208.1117}
    ▶ iterations: [{}, {...}]
▼ errorEllipses: {...}
  ▼ point4: {W: 0.000391496, phi: 84.402952872, a: {dm: 0.0849025, m: 0.0084903}, b: {dm: 0.05121, m: 0.005121}}
    W: 0.000391496
    ▶ a: {dm: 0.0849025, m: 0.0084903}
    ▶ b: {dm: 0.05121, m: 0.005121}
    phi: 84.402952872
  ▼ point5: {W: 0.0000801862, phi: 75.023659617, a: {dm: 0.0623189, m: 0.0062319},...}
    W: 0.0000801862
    ▶ a: {dm: 0.0623189, m: 0.0062319}
    ▶ b: {dm: 0.0542618, m: 0.0054262}
    phi: 75.023659617

```

Рисунок 4 – Ответ сервера, в случае успешного уравнивания

Описанная реализация была проведена для автоматизации обработки линейно-угловых сетей, которая применима, к примеру, для проектирования сети.