**Клиент-серверное web приложение визуального построения и расчета формул**

**SP V-G-SH**

Техническая документация

Белгород 2017 г.

Оглавление

[Введение 3](#_Toc488648209)

[Клиентское web приложение 3](#_Toc488648210)

[Описание установки и развертывания 3](#_Toc488648211)

[Общий функционал 3](#_Toc488648212)

[Структура входного файла 3](#_Toc488648213)

[Подключение к серверу 5](#_Toc488648214)

[Импорт входных данных 5](#_Toc488648215)

[Построение схем для вычисления 7](#_Toc488648216)

[Редактирование и фильтрация данных 9](#_Toc488648217)

[Проверка и решение схем на сервере 12](#_Toc488648218)

[Экспорт схем в выходной файл 12](#_Toc488648219)

[Сервер приложения 13](#_Toc488648220)

[Описание установки и развертывания 13](#_Toc488648221)

[Общий функционал 13](#_Toc488648222)

[REST API сервера с описанием структуры входных и выходных данных 13](#_Toc488648223)

[Обработка ошибок 13](#_Toc488648224)

[Функции сервера 13](#_Toc488648225)

# Введение

Клиент-серверное приложение предназначено для визуального построения, проверки и расчета составленных или загруженных формул. Данное приложение позволяет производить математические операции над числовыми значениями, а также производить операции над матрицами по всем правилам математики, добавлять к табличным данным условия фильтрации, редактировать данные и сохранять произведенные расчеты и сопутствующие данные в выходной файл для последующей работы.

Сервер приложения может использоваться обособленно от клиентского приложения, в роли формульного REST API движка.

# Клиентское web приложение

Интерфейс клиентского приложения представлен на рисунке Рис.1.

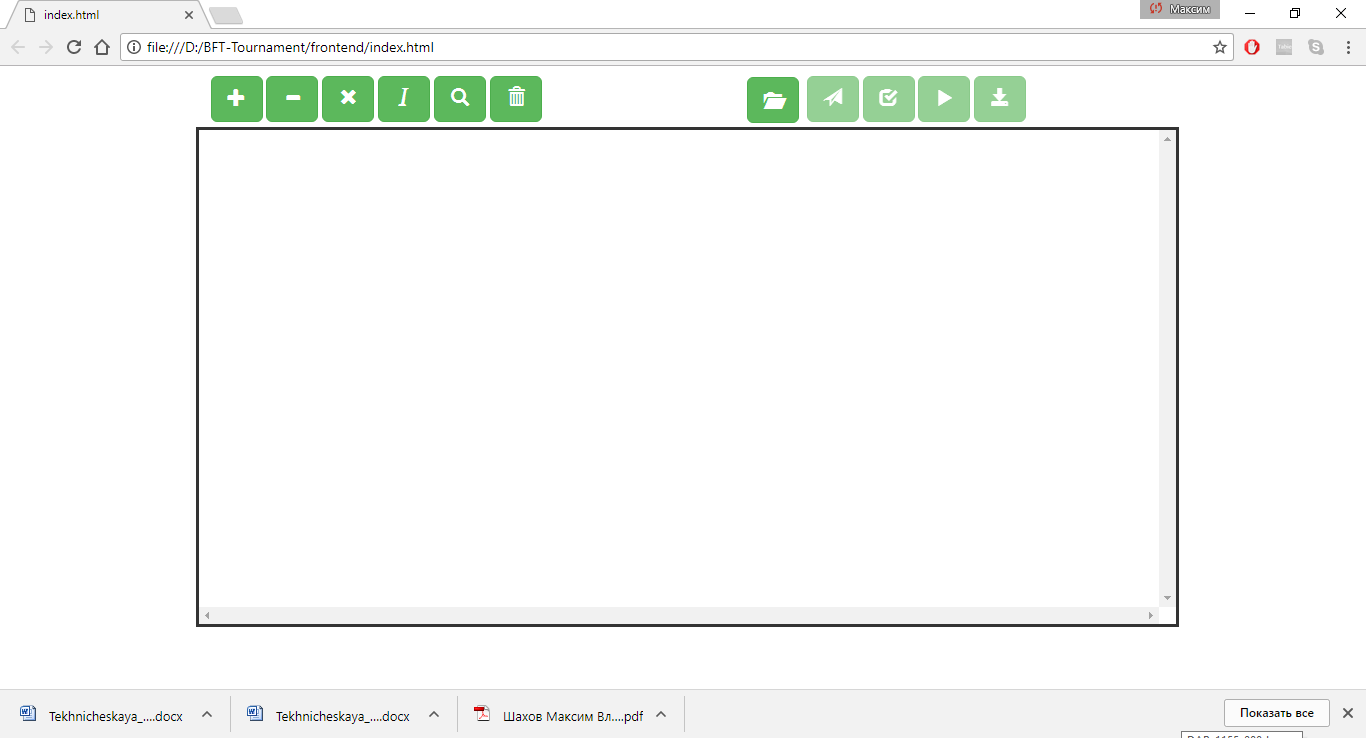


Рис.1. Интерфейс клиентского приложения

## Описание установки и развертывания

Клиентское приложение, для ознакомительных целей или в целях тестирования функциональности, в предварительной установке и развертывании не нуждается. Для запуска клиента необходимо в папке с клиентом открыть файл index.html через браузер.

## Общий функционал

Клиентское приложение позволяет загружать входные данные в формате JSON, составлять формулу типа «схема», фильтровать и редактировать входные и промежуточные данные, проверять на корректность формулу и производить расчет по формуле, заносить промежуточные вычисления и результат в переменные, в дальнейшем использовать переменные в вычислении, выгружать созданные схемы и данные в выходной файл для последующей работы. Если результат в формуле не помещается во временную переменную, то создается переменная result для просмотра результата. В дальнейшем данную переменную нельзя использовать в вычислениях.

## Структура входного файла

Входной файл представляет собой JSON файл со следующей структурой:  
корневой элемент – это массив, в котором содержатся объекты, объект описывает данные или математическую операцию.  
В каждом элементе необходимыми полями являются поля **name** и **type**, которые описывают уникальное наименование элемента и его тип. У элементов допустимыми типами являются типы – double,long,table,temp. У операций допустимые типы – mul,add,sub,div.

Если элемент является копией другого элемента, тогда необходимо, чтобы у этого элемента имелось поле source со значением наименования эталонного объекта (объекта клонирования).   
 Если элемент является таблицей, то у элемента должны присутствовать поля **columns** c описанием колонок таблицы. Если же элемент является копией другого элемента, то описание колонок необязательно, метаданные подтянутся из родительского элемента. У числовых элементов должно отсутствовать поле **columns**. Поле **columns** представляет собой массив объектов с описанием колонок. Каждый объект имеет поля **name** и **type**. **Name** – это наименование колонки , **type** – тип колонки, который принимает значения long,double,string,date. В поле **value** заносится значение элемента – при этом у числовых элементов – это числовое значение, у таблицы – это таблица. При этом у таблицы размерность описания колонок **columns** должно соответствовать размерности значения **value**, также значения должны соответствовать типу своей колонки. Табличный элемент может содержать поле фильтрации **filter**, которое представляет собой объект в котором перечислены объекты с наименованием соответствующих колонок. Каждый объект содержит поля **type, value, operator**, которые хранят информацию о типе колонки, значении фильтра и логической операции сравнения, если таковая применяется (< - меньше, > - больше, = - равно). Типы в фильтрах должны соответствовать типам соответствующих колонок.

Ниже представлен пример входного файла.

[

{

"name": "table\_1",

"type": "table",

"value": [ [ "Tiger Nixon", "System Architect", "Edinburgh", 5421, "2011-04-25", 320.800 ],

[ "Garrett Winters", "Accountant", "Tokyo", 8422, "2011-07-25", 170.750 ]],

"columns": [

{"name": "Name","type": "string"},{"name": "Position","type": "string"},

{"name": "Office","type": "string"},{"name": "Extn","type": "long"},

{"name": "StartDate","type": "date"},{"name": "Salary","type": "double"}

]

},

{

"name": "table\_2",

"type": "table",

"value": [

[ "System Architect", "Edinburgh", 5421, "2011-04-25", 320.800 ],

[ "Accountant", "Tokyo", 8422, "2011-07-25", 170.750 ],

[ "Support Lead", "Edinburgh", 9497, "2013-03-03", 342.000 ]],

"columns": [

{"name": "Position","type": "string"},

{"name": "Office","type": "string"},{"name": "Extn","type": "long"},

{"name": "StartDate","type": "date"},{"name": "Salary","type": "double"}

]

},

{

"name": "table\_3",

"type": "table",

"value": [

[ "2011-04-25", 320.800 ],

[ "2011-07-25", 170.750 ],

[ "2009-01-12", 86.000 ],

[ "2012-03-29", 433.060 ],

[ "2008-11-28", 162.700 ],

[ "2013-03-03", 342.000 ]],

"columns": [

{"name": "StartDate","type": "date"},{"name": "Salary","type": "double"}

]

},

{

"name": "table\_4",

"type": "table",

"value": [

[ 13, 26],

[ 9, 13]],

"columns": [

{"name": "a1","type": "long"},{"name": "a2","type": "long"}

]

},

{

"name": "table\_5",

"type": "table",

"value": [

[ 7, 4],

[ 2, 3]

],

"columns": [

{"name": "b1","type": "long"},{"name": "b2","type": "long"}

]

},

{

"name": "A\_1",

"type": "long",

"value": 10000

},

{

"name": "B\_1",

"type": "double",

"value": 5000.500

}

]

## Подключение к серверу

Для подключения к серверу используется адрес (url) <http://localhost:6060/servletexam/mainServlet?action=>

url меняется в файле index.html.

## Импорт входных данных

Для импорта входного файла и начала работы, необходимо выбрать на верхней панели значок папки и в диалоговом окне выбрать соответствующий файл для загрузки (Рис.2.).

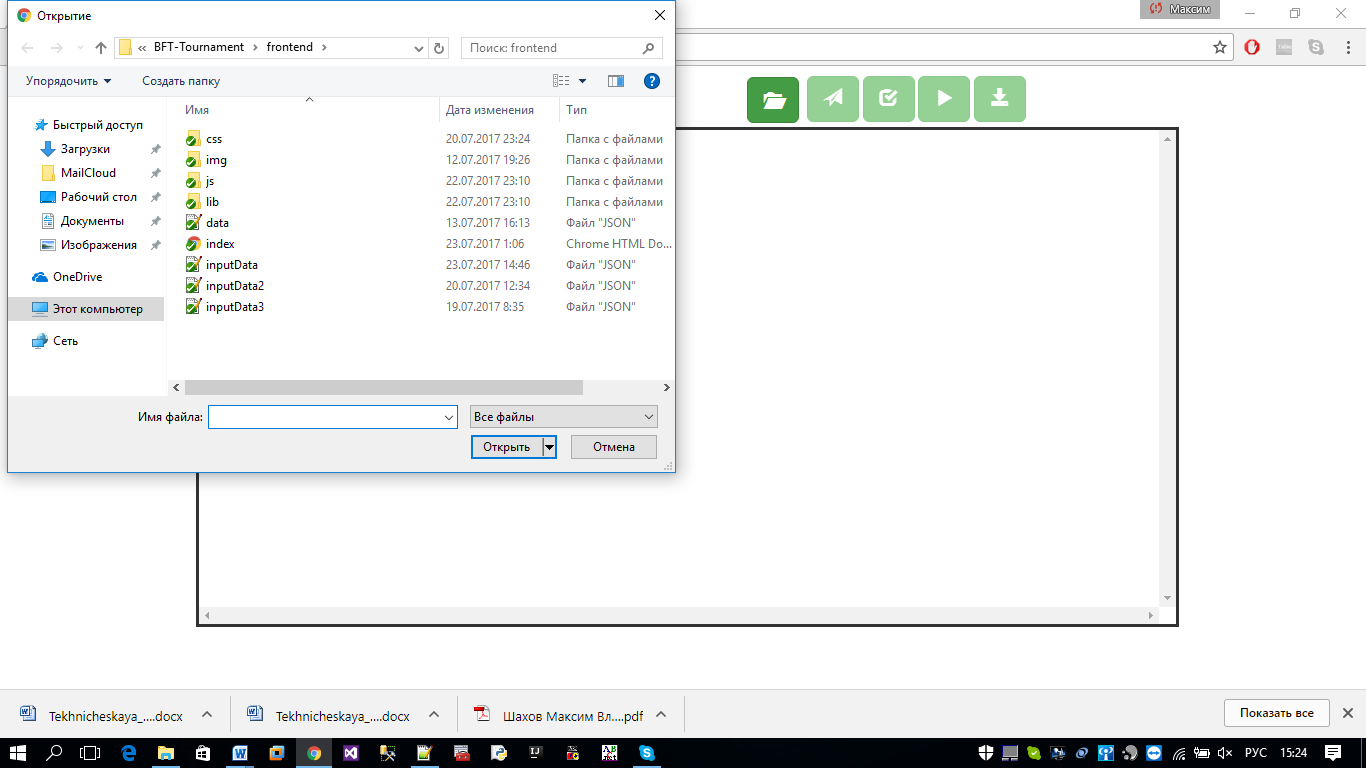


Рис.2. Импорт входных данных

После выбора файла, проводится проверка входного файла на корректность . В случае корректности файла, элементы создаются в рабочей области приложения и становятся активными кнопки сохранения схемы в файл и отправки данных на сервер (Рис.3.).

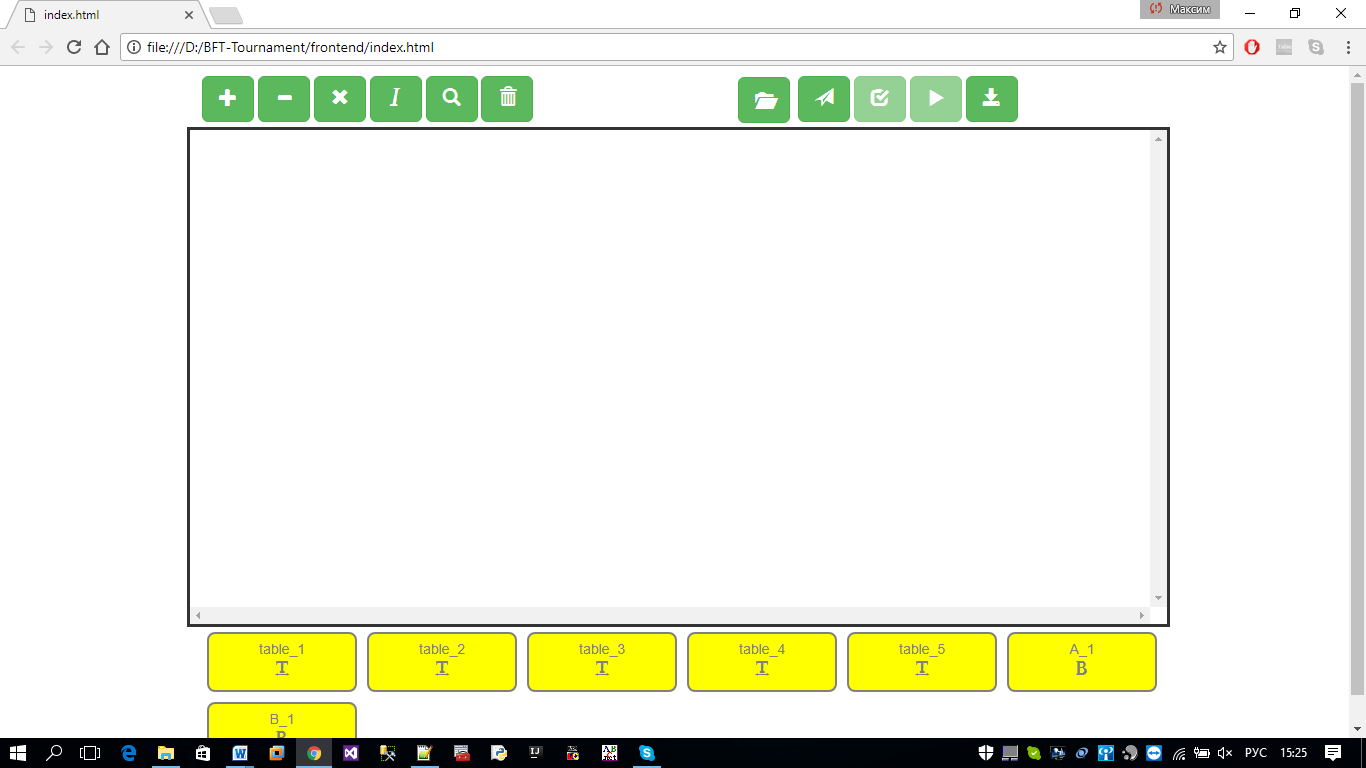


Рис.3. Интерфейс клиентского приложения с загруженными данными

Если файл не проходит проверку, то выводится сообщение с ошибкой (Рис.4.).

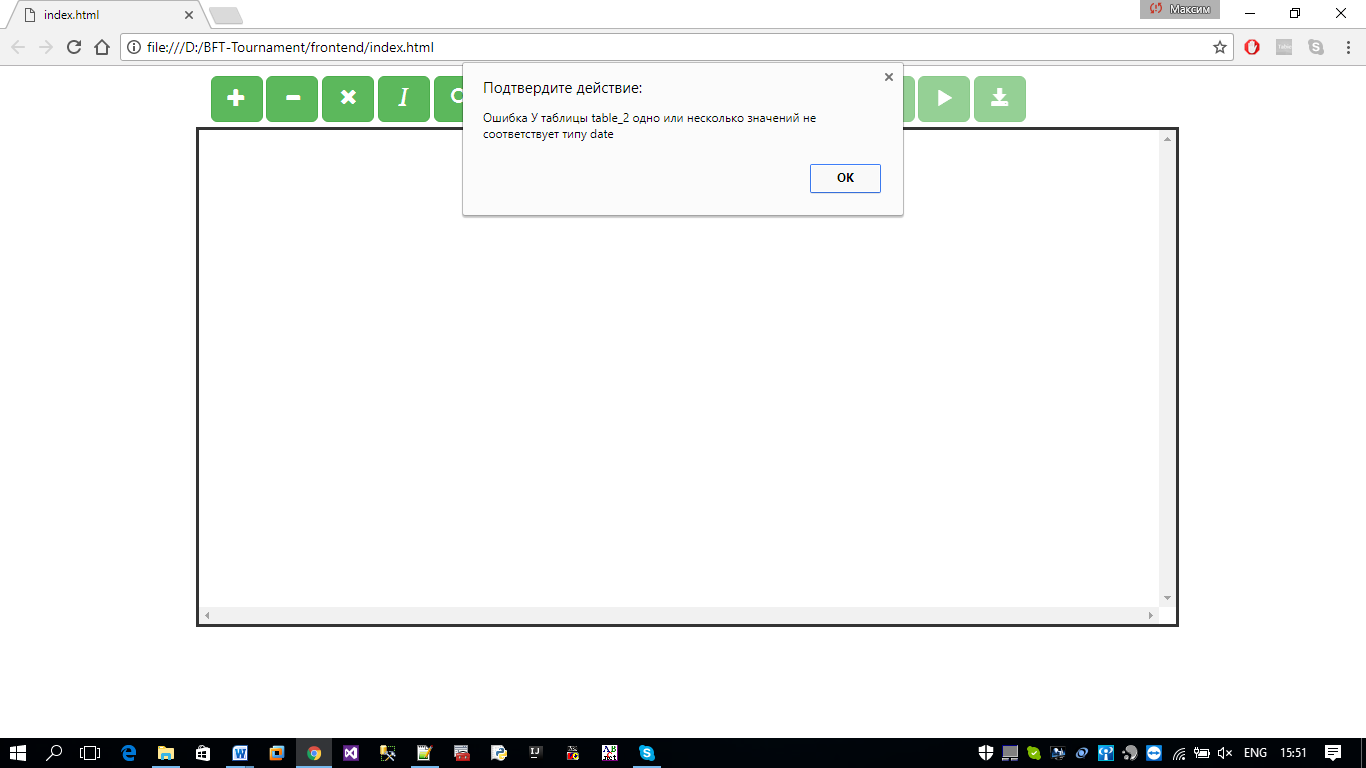


Рис.4. Ошибка при проверке входного файла

## Построение схем для вычисления

Все операционные кнопки и элементы кликабельны. При нажатии на элементы, которые расположены ниже зоны построения схем, создаются их копии и помещаются в зону построения схем. При нажатии на операции, также создаются их копии и помещаются в зону построения схем (Рис.5.)..

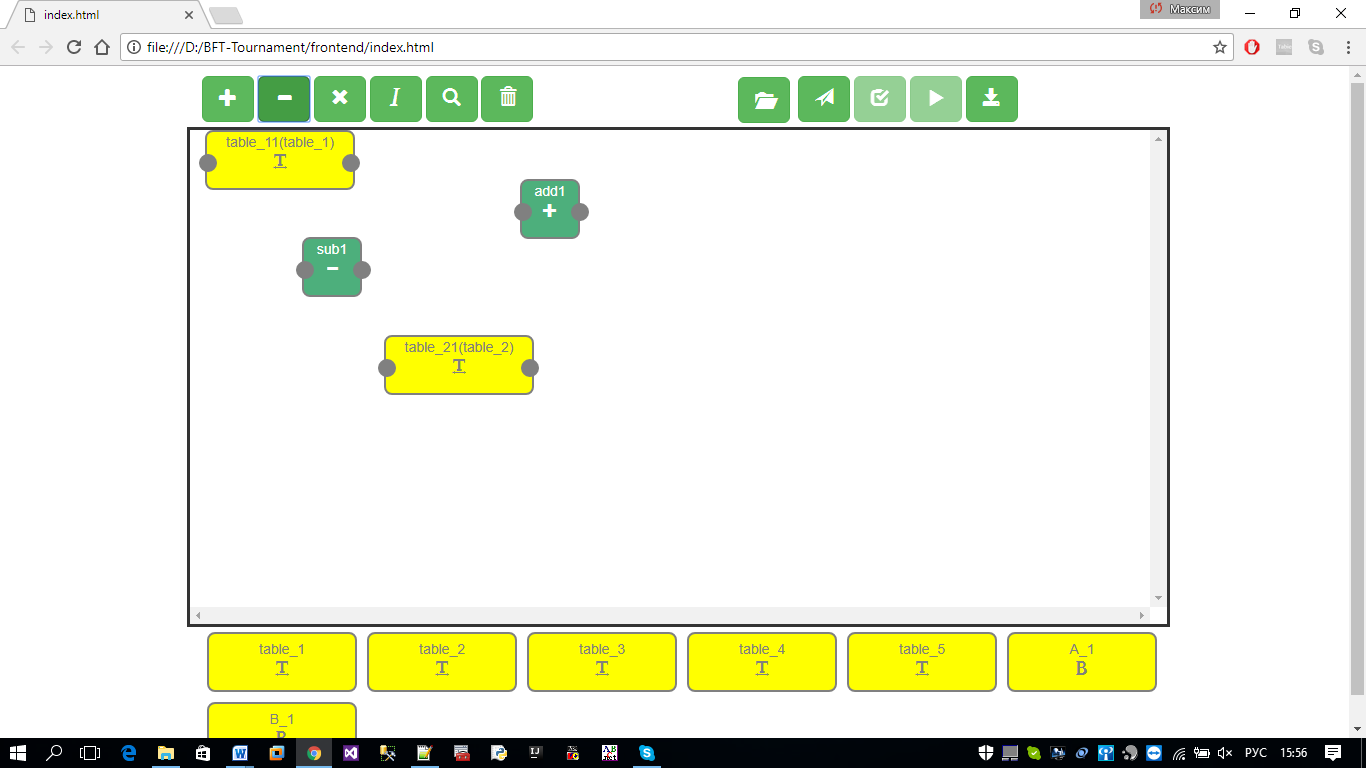


Рис.5. Создание элементов в зоне построения формул

У элементов данных имеется контекстное меню (Рис.6.), которое вызывается нажатием правой кнопки мыши на элементе, для просмотра элемента и его удаления.

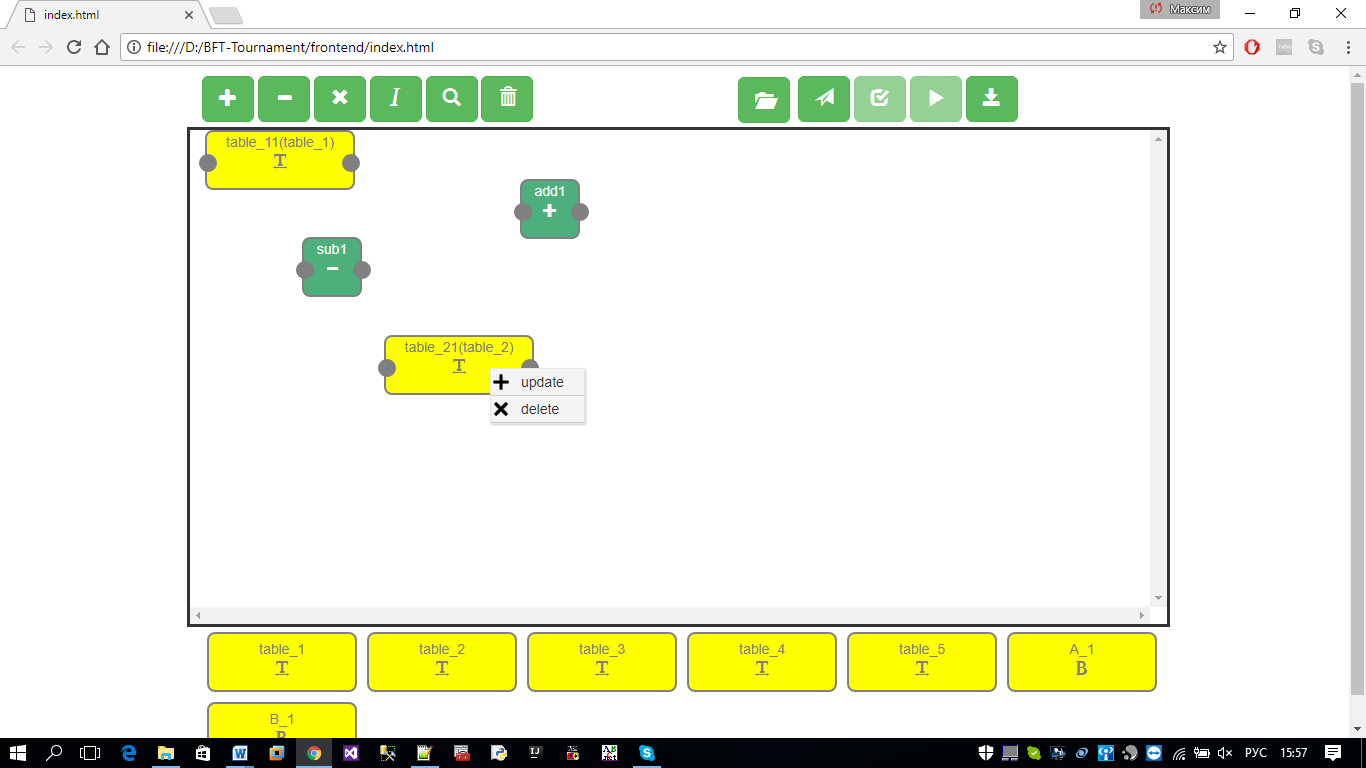


Рис.6. Контекстное меню элементов

Также, элементы, в зоне построения схем, можно выделять для последующего удаления, нажав на них однократно левой кнопкой мыши, при этом они выделяются рамкой (Рис.7.).

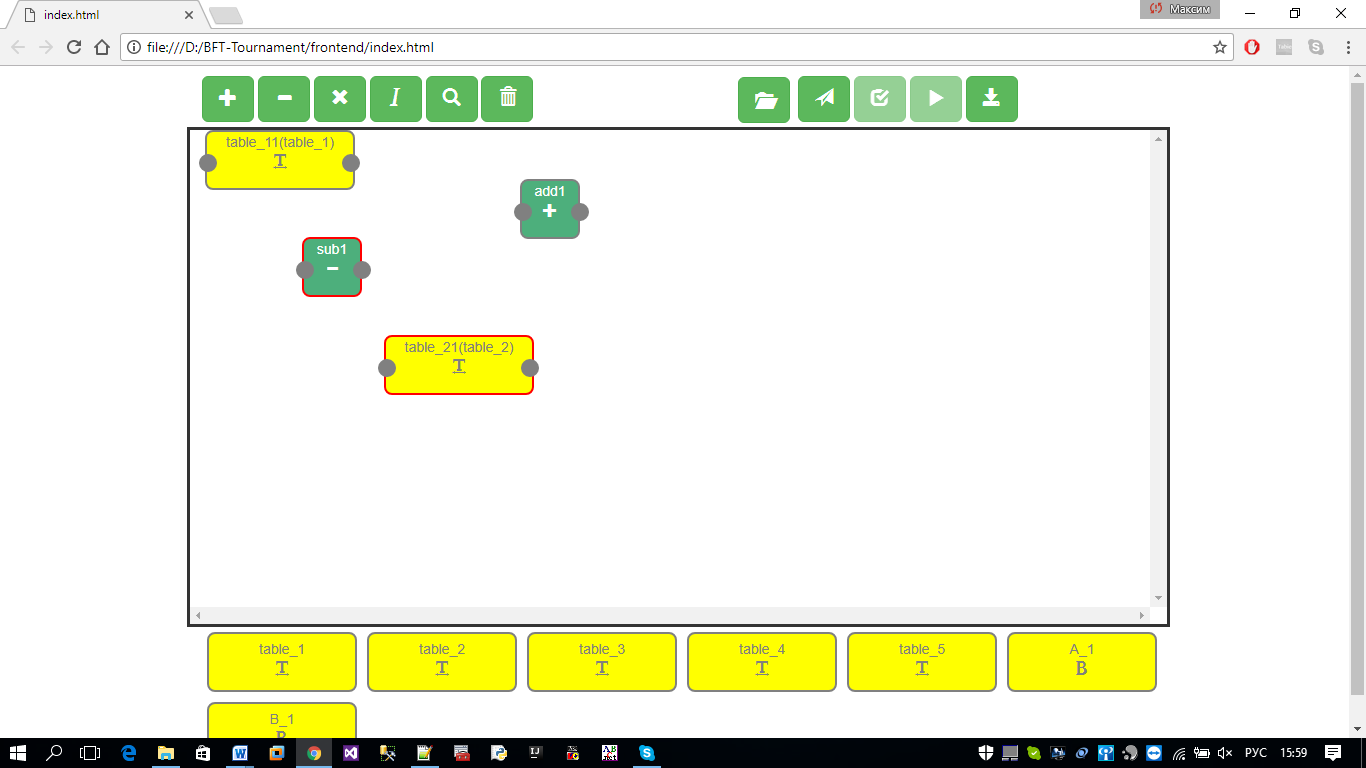


Рис.7. Множественное выделение объектов для удаления

Для удаления выделенных элементов, на панели операций имеется кнопка с значком корзины.

Элементы, находящиеся в зоне построения формул, можно соединять между собой (Рис.8.), при этом нельзя соединить операции с операциями, элементы данных между собой, зациклить формулу.

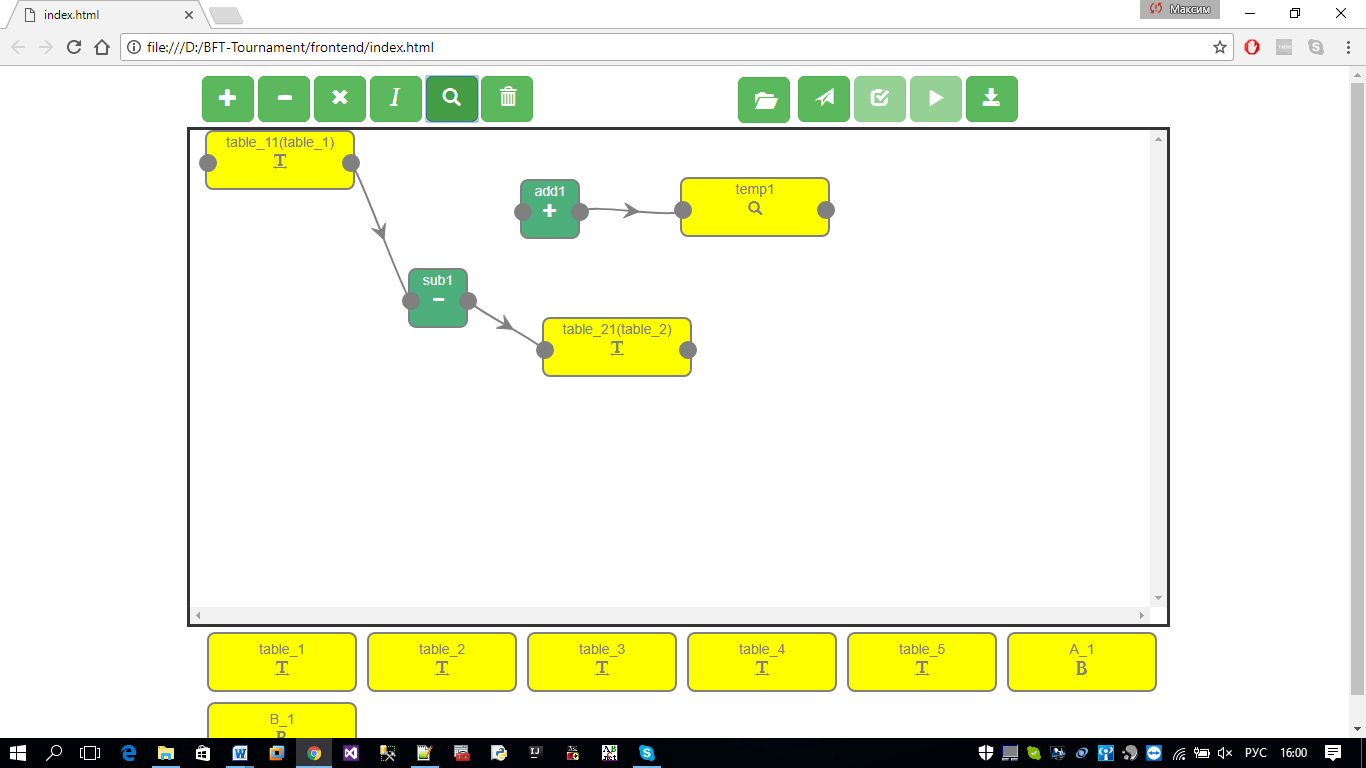


Рис.8. Связывание элементов в формулы

## Редактирование и фильтрация данных

После вызова контекстного меню у элемента, можно выбрать пункт update и просмотреть (Рис.9.) или отредактировать элемент (Рис.10.). При этом открывается форма просмотра элемента, если элемент имеет тип таблица, то открывается таблица со значениями элемента и полями задания условной фильтрации.

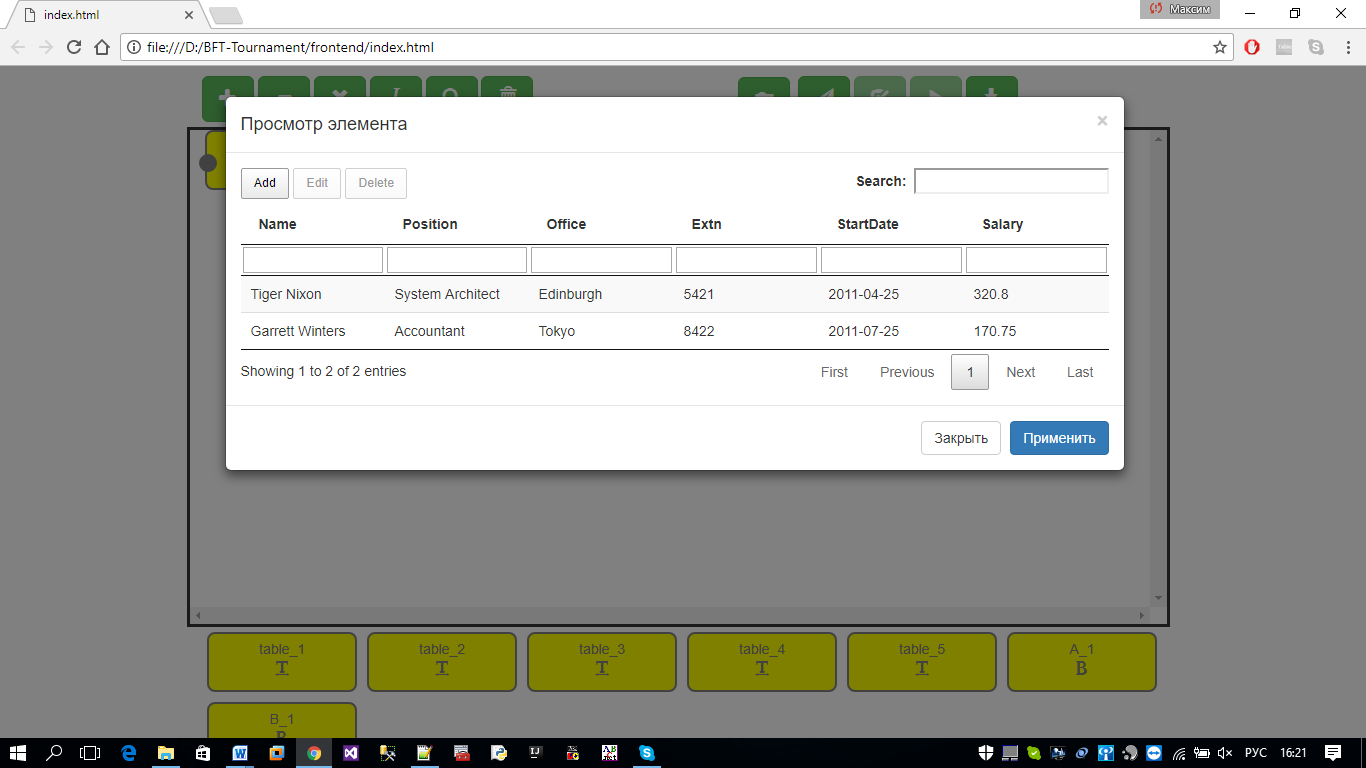


Рис.9. Просмотр табличного элемента

Если элемент имеет числовой тип, то открывается окно с полем ввода, содержащим значение элемента.

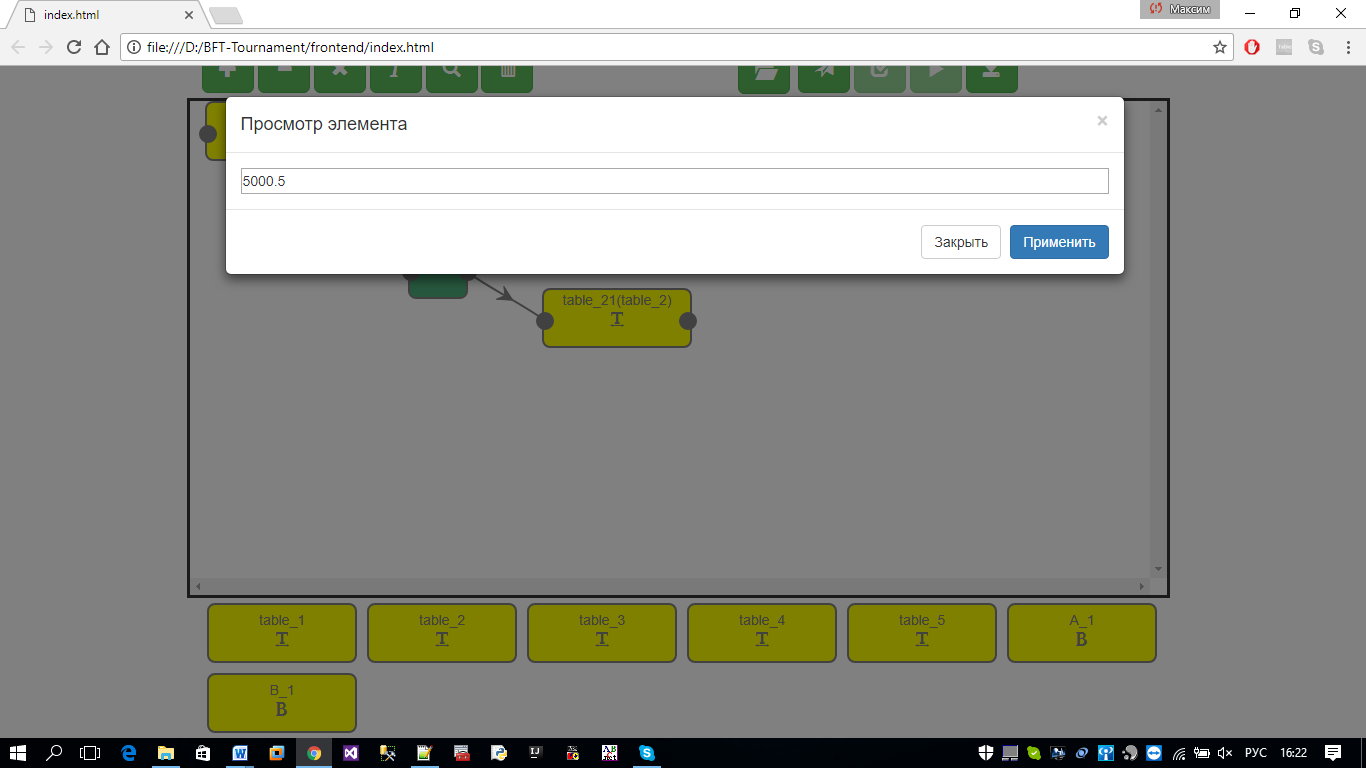


Рис.10. Просмотр скалярного элемента

К табличным элементам, к их колонкам, можно добавлять условия фильтрации (Рис.11.), используя операторы >,<,=. Если данные операторы отсутствуют, тогда идет фильтрация на частичное соответствие значениям.

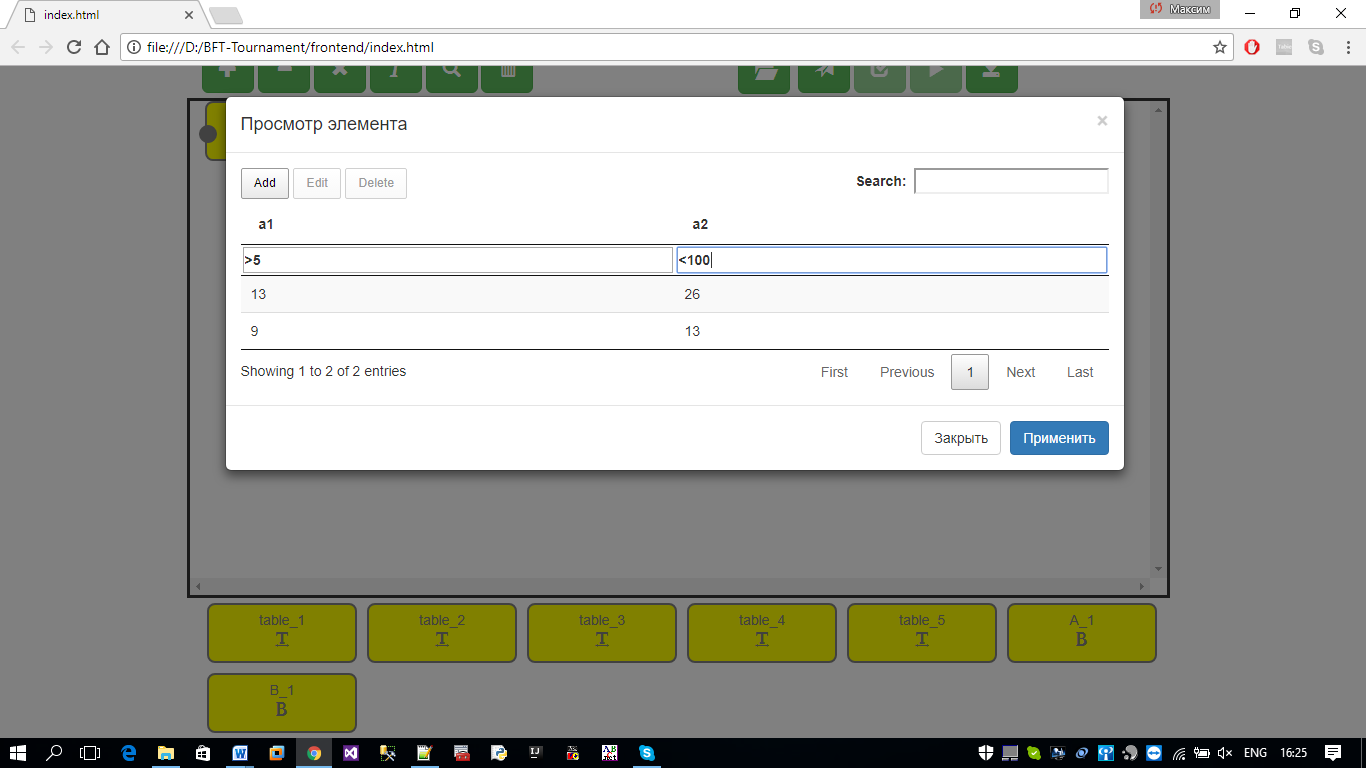


Рис.11. Фильтрация табличных данных

Запись в таблице можно отредактировать (Рис.12.), добавить новую (Рис.13.) или удалить (Рис.14.), нажав на соответствующие кнопки на форме.

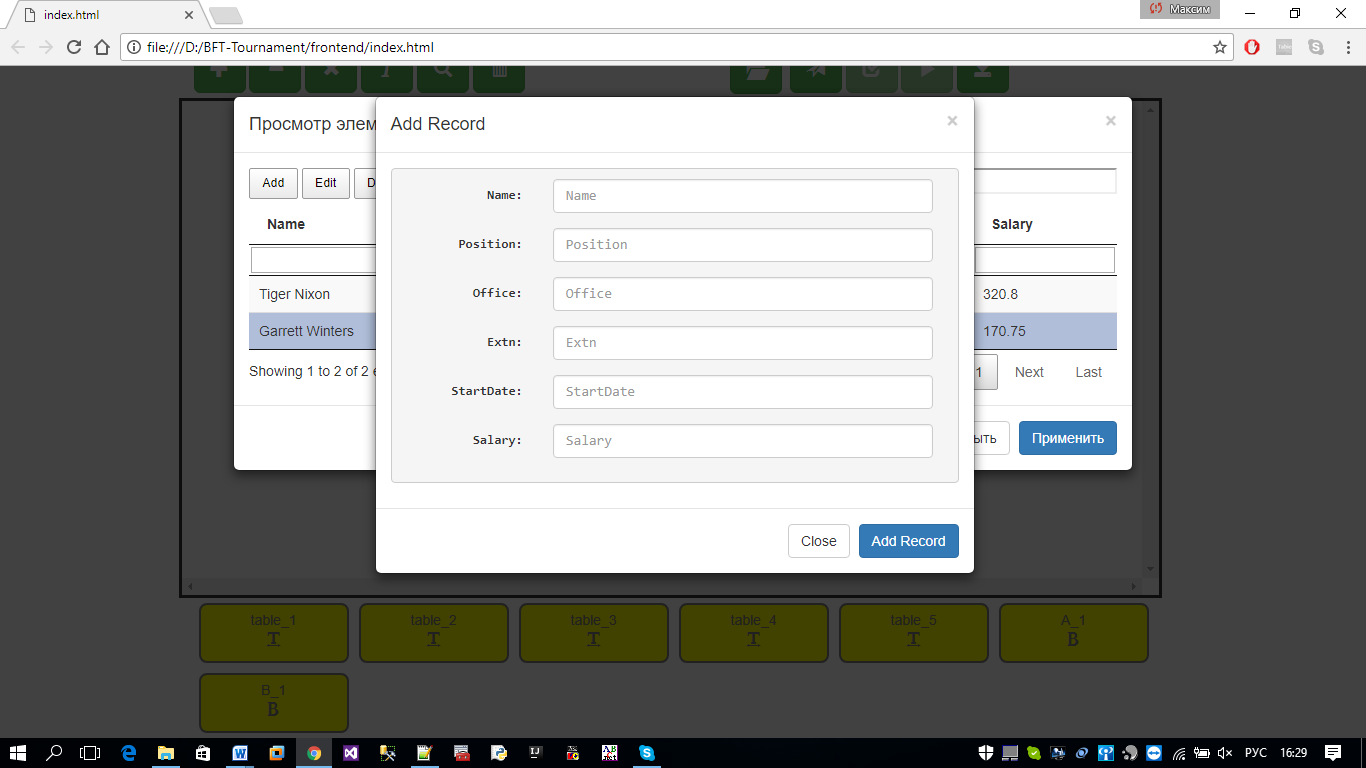


Рис.12. Добавление записи в табличный элемент

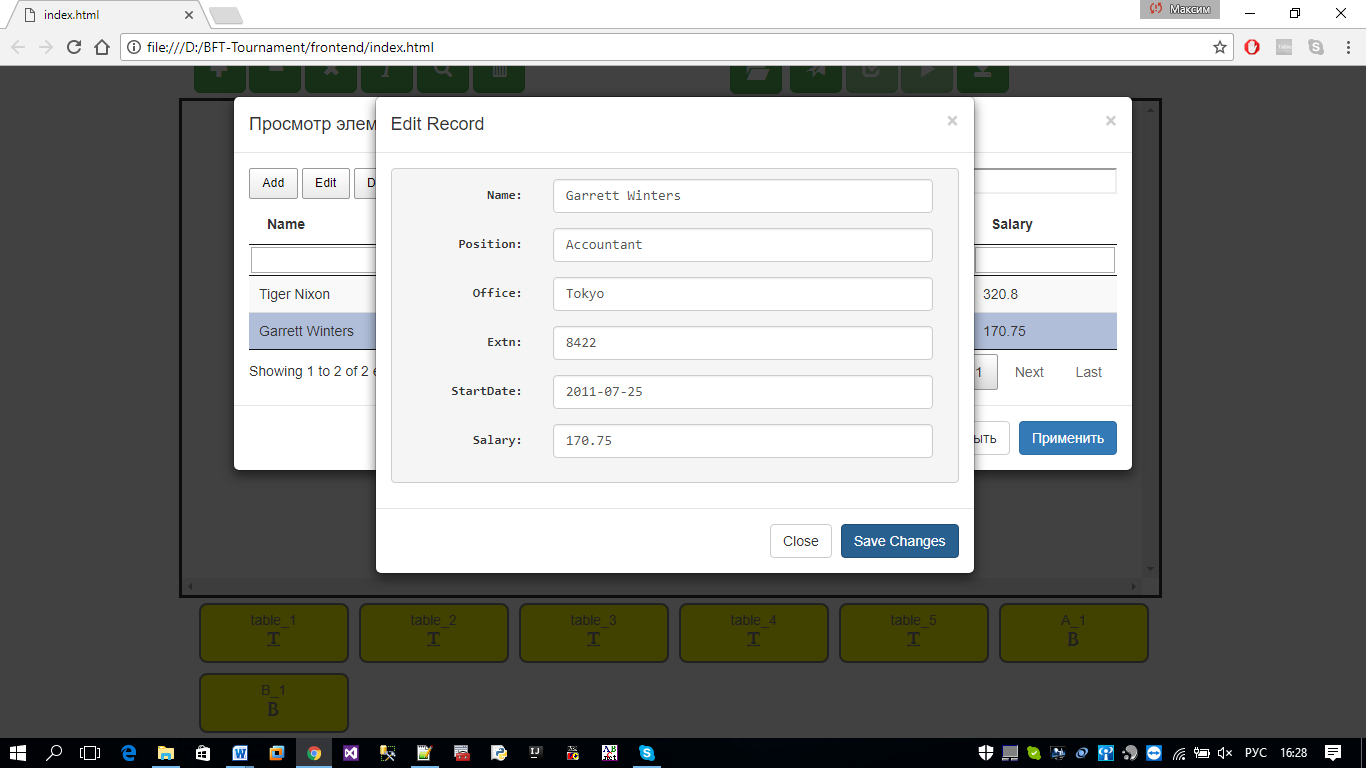


Рис.13. Редактирование табличной записи

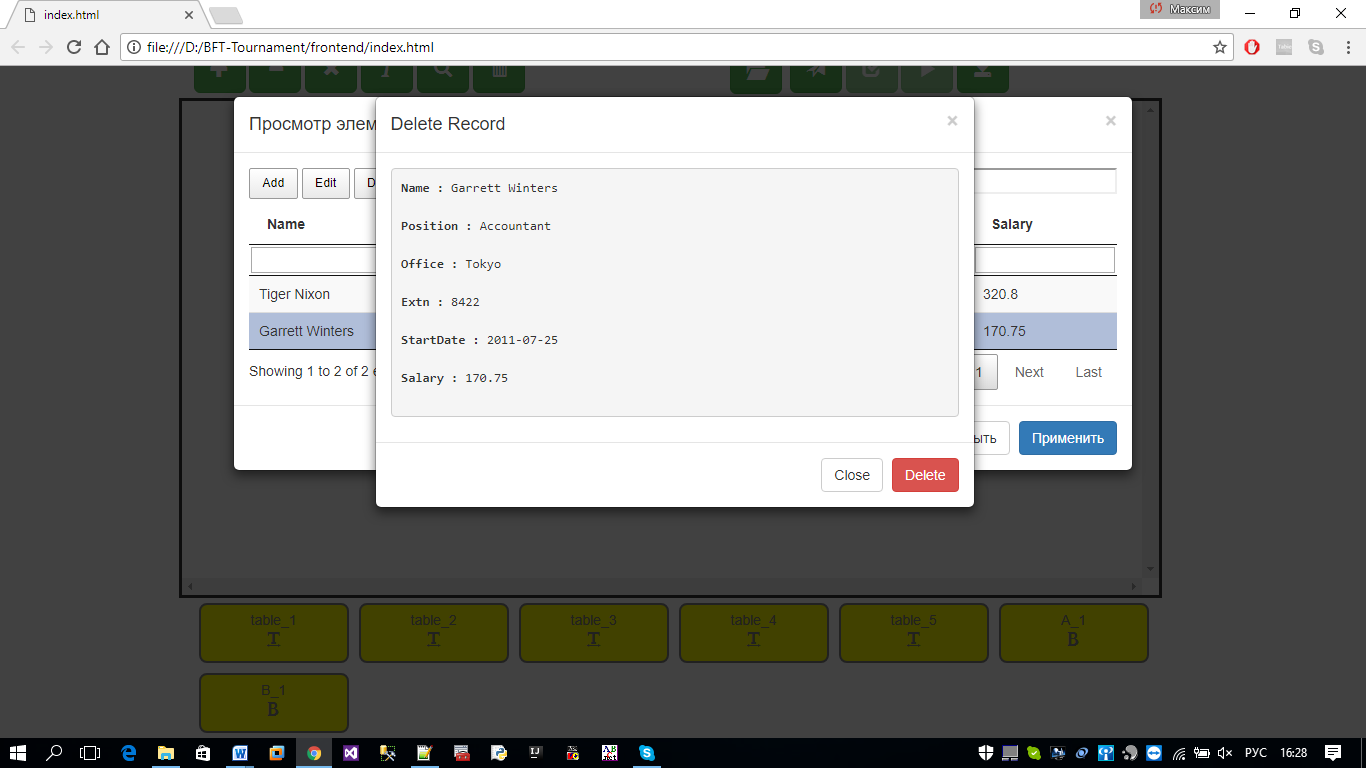
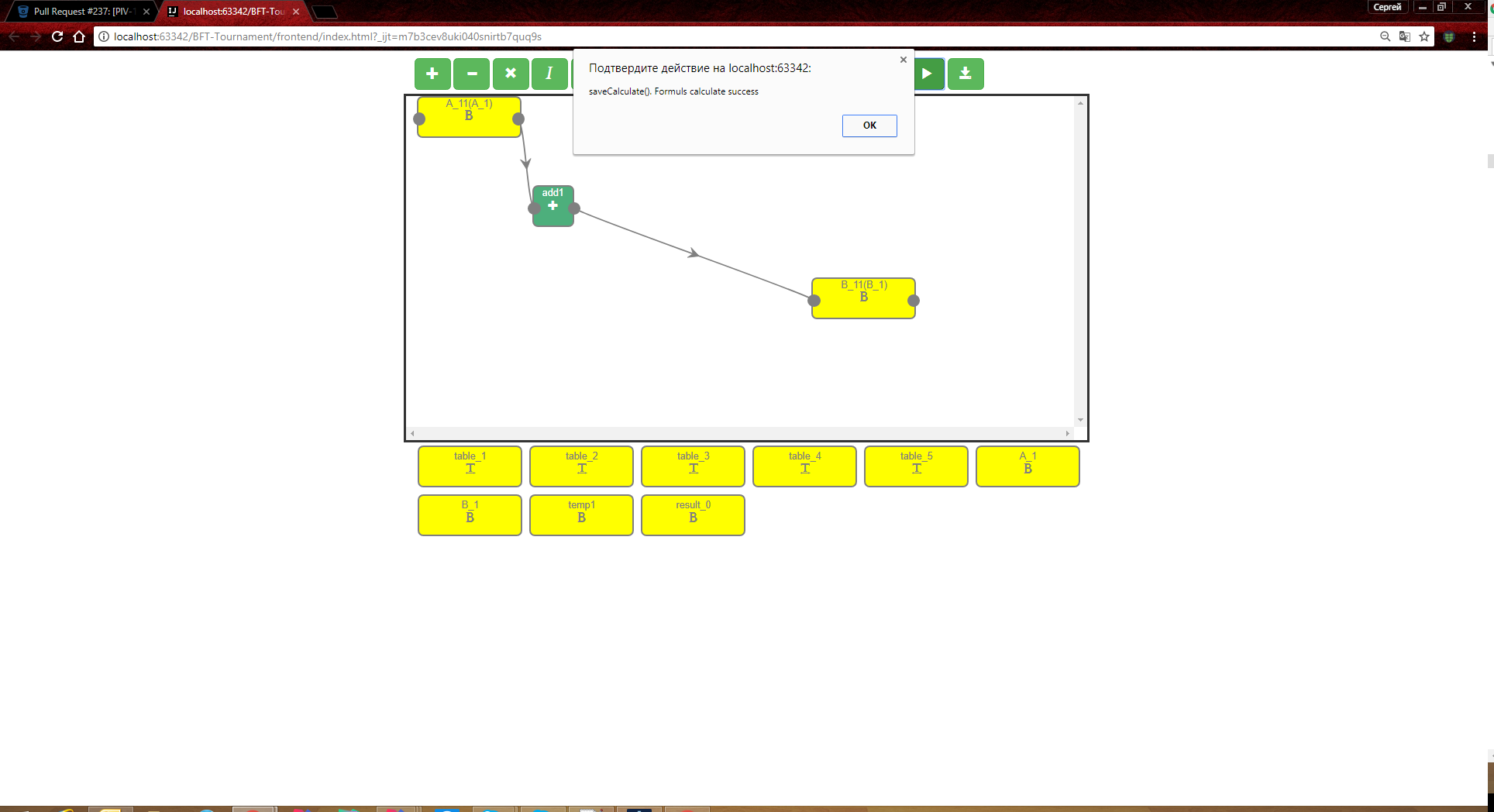


Рис.14. Удаление табличной записи

## Проверка и решение схем на сервере

Корректность составленных формул можно проверить по кнопке проверки формул. После нажатия на кнопку, появится всплывающая подсказка с результатом проверки (Рис.15.). В случае некорректной формулы, выдается ошибка (Рис.16.).

 Рис.15. Результат проверки формулы

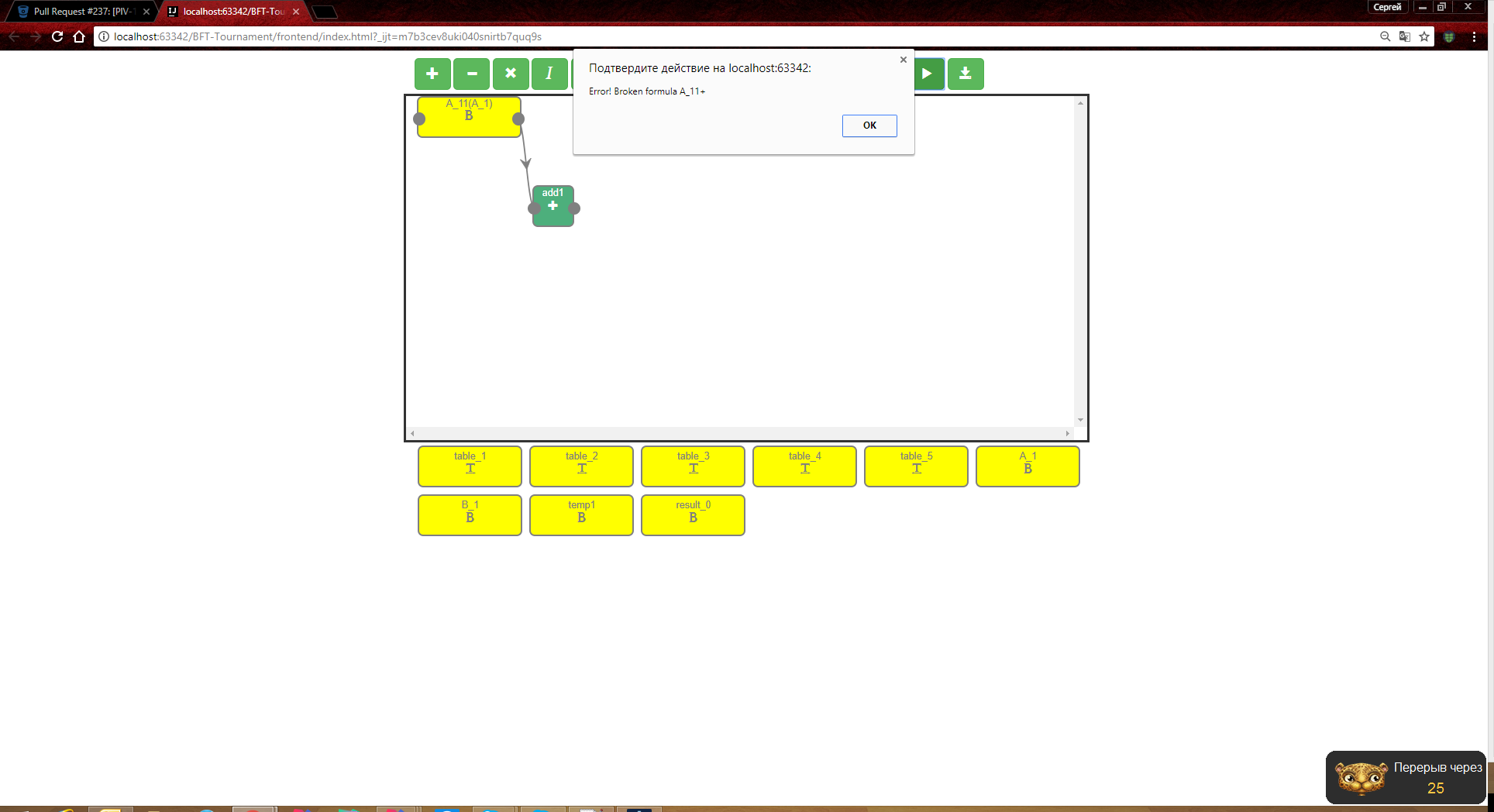


Рис.16. Результат при проверке некорректной формулы

Элементы типа result нельзя использовать в дальнейших вычислениях (Рис.17.).

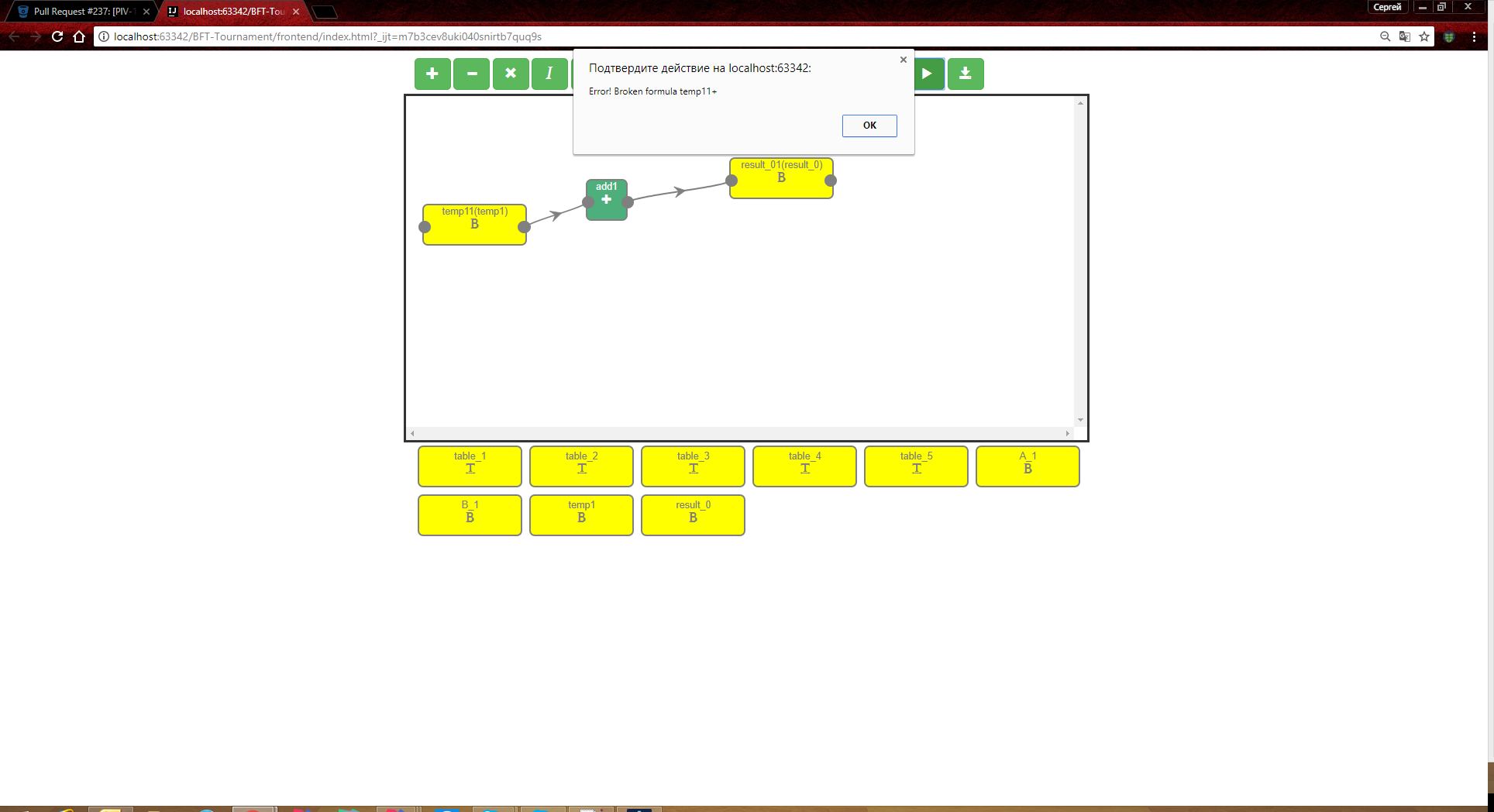


Рис.17. Результат при использовании недопустимых элементов в формуле

## Экспорт схем в выходной файл

Составленные формулы и элементы можно сохранить в выходной файл, для последующей работы.

Для этого необходимо нажать на кнопку выгрузки схем (Рис.18.).



Рис.18. Экспорт схем и данных в выходной файл

Для загрузки состояния необходимо обновить страницу (F5) и загрузить файл состояния. После загрузки состояния необходимо отправить данные на сервер.

# Сервер приложения

## Описание установки и развертывания

Для запуска сервера предоставляемся war файл. Который необходимо развернуть на сервере.

По стандарту нам необходим Tomcat 7.0.78. Java 1.8. Подключение к БД настраивается в файле config.properties, который необходимо положить в папку где находится файл для запуска сервера tomcat.

## Общий функционал

Сервер поддерживает работу с файлами заданного формата. Существует набор определенных операций, которые поддерживает сервер через POST запрос, где параметр

* action – название действия, которое будет обрабатывать сервер.
* sessionId – уникальный номер(айди ) сессии пользователя.
* data – входной набор данных

## REST API сервера с описанием структуры входных и выходных данных

Существующие action:

* start\_service(не передается sessionId) – создание сессии. Запись входных данных в базу. Возвращает id сессии.
* calculate – вычисление переданных на сервер формул
* save\_data – сохранение данных после фильтрации или изменения таблицы\переменной
* close\_session(не передается data) – закрытие сессии. Удаление всех данных принадлежащих данной сессии (в UI не используется(возможна реализация в будущем))
* При передаче неизвестного action в ответ вернется hello(использовалось для тестирования)

# Обработка ошибок

При возникновении ошибки создается объект с текстом ошибки – массив содержащий один элемент error.

[{

“error”: “Текст ошибки”

}]

# Функции сервера

Сервер поддерживает следующие операции **: Сложение, вычитание, умножение**, деление числовых переменных. Сложение, вычитание, умножение, деление таблиц возможно только в том случае, если таблица представляет собой числовую матрицу. Так же существуют дополнительные ограничения для умножения и деления таблиц. Умножать можно только те таблицы у которых число столбцов первой таблицы равно числу строк второй таблицы(см. статью в википедии по умножению матриц). При делении таблица на которую делят должна содержать обратную матрицу(см. статью в википедии по делению матриц).

При сложении\вычитании таблицы с числовым параметром возникает ошибка.

Нельзя использовать арифметические действия с таблицами состоящими не только из чисел.