

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

**ЗАДАНИЕ №4. СИСТЕМА КОНСТРУИРОВАНИЯ И РАСЧЁТА
ЭЛЕКТРИЧЕСКИХ СХЕМ**

РУКОВОДСТВО ПРОГРАММИСТА

студента 2 курса 251 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Синкевича Артема Александровича

Проверил
к. ф.-м. н.

И. А. Батраева

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Назначение и условия применения программы	6
1.1 Назначение программы	6
1.2 Возможности программы	6
1.3 Условия, необходимые для выполнения программы	6
1.3.1 Требования к аппаратному обеспечению	6
1.3.2 Требования к программному обеспечению	6
2 Минимальное работоспособное приложение	7
3 Основное окно программы	8
3.1 Новый файл	8
3.2 Открытие файла	8
3.3 Сохранение в файл	8
3.4 Добавление элемента	8
3.5 Вычисление характеристик схемы	9
3.6 Выполнено размещение добавляемого элемента	9
3.7 Выбран изменяемый элемент в виджете отрисовки	9
3.8 Выбраны точки для измерения напряжения в виджете отрисовки ..	9
3.9 Выбран элемент для измерения тока в виджете отрисовки	9
3.10 Изменилось состояние виджета отрисовки	10
4 Виджет отрисовки схемы	11
4.1 Состояния отрисовки	11
4.2 Константы	11
4.3 Вспомогательные функции	11
4.4 Сигналы	12
4.5 Размеры виджета	12
4.6 Начало размещения элемента	12
4.7 Начало изменения, переключения, удаления элемента, измере- ния тока	12
4.8 Начало измерения напряжения	12
4.9 Масштабирование	12
4.10 Нажатие кнопки мыши	12
4.11 Перемещение указателя мыши	13
4.12 Отпускание кнопки мыши	13

4.13	Отрисовка виджета	13
5	Диалог добавления элемента	15
5.1	Поля	15
5.2	Выбран элемент в списке	15
5.3	Нажата кнопка «ОК»	15
5.4	Прочее	15
6	Диалог изменения элемента	16
6.1	Поля	16
6.2	Установка элемента	16
6.3	Получение свойств элемента	16
6.4	Нажата кнопка «ОК»	16
7	Класс управления электрической схемой и элементами	17
7.1	Элементы-образцы	17
7.2	Фабрики элементов	17
7.3	Добавление элемента	17
7.4	Получение элемента	17
7.5	Удаление элементов	17
7.6	Чтение схемы из JSON-документа	18
7.7	Запись схемы в JSON-документ	18
8	Электрический элемент	19
8.1	Поля	19
8.2	Конструктор	19
8.3	Обновление свойств	19
8.4	Заполнение таблицы свойств	19
8.5	Запись в JSON-объект	19
8.6	Получение крайних точек	19
8.7	Отрисовка элемента	19
8.8	Изображение элемента	20
8.9	Обновление элемента при прохождении тока	20
8.10	Прочее	20
8.11	Фабрики элементов	20
8.11.1	Создание по позиции, ориентации и списку свойств	20
8.11.2	Создание из JSON-объекта	20
9	Электрические элементы	21

9.1	Провод	21
9.2	Резистор	21
9.3	Выключатель	21
9.4	Электрический нагреватель	21
9.5	Электрическая лампа	21
9.6	Источник напряжения (ЭДС)	21
9.7	Источник тока	22
10	Класс вычисления характеристик схемы	23
10.1	Функции, доступные извне	23
10.1.1	Вычисление характеристик схемы	23
10.1.2	Нахождение напряжения между двумя точками	23
10.1.3	Нахождение тока через элемент	23
10.2	Система непересекающихся множеств	23
10.3	Первый этап — нахождение вершин и элементов схемы	23
10.4	Второй этап — нахождение напряжений и токов	24

ВВЕДЕНИЕ

В данном документе приведено руководство программиста для программной системы конструирования и расчёта электрических схем.

В данном документе, в разделе «Назначение и условия применения программы» указаны назначение и возможности программы, доступные пользователю, условия, необходимые для выполнения программы (требования к аппаратному, программному обеспечению и т.п.)

В разделе «Минимальное работоспособное приложение» представлена инструкция по компиляции программы.

В каждом следующем разделе описывается класс, его назначение, поля и функции.

1 Назначение и условия применения программы

1.1 Назначение программы

Данная программная система предназначена для конструирования, визуализации электрических схем постоянного тока и расчёта их основных характеристик, т.е. величины тока в каждой точке и величины напряжения между двумя произвольными точками.

1.2 Возможности программы

1. Добавление, перемещение и удаление электрических элементов в зоне сборки схемы;
2. Изменение свойств установленных элементов, переключение выключателей;
3. Расчёт основных характеристик электрической схемы;
4. Визуализация электрической схемы: изображение электрической цепи, текущего состояния выключателей, ламп и электронагревателей;
5. Вычисление напряжения между любыми двумя точками схемы, а также величины тока через элемент;
6. Сохранение созданной схемы в файле и загрузка её из файла.

1.3 Условия, необходимые для выполнения программы

1.3.1 Требования к аппаратному обеспечению

Рекомендуемые процессор и видеокарта — любые, удовлетворяющие требованиям операционной системы.

Рекомендуемый объём свободной оперативной памяти (не занятой операционной системой и другими программами) — 64 Мб.

1.3.2 Требования к программному обеспечению

Необходима операционная система Windows 7 или новее, или операционная система семейства Linux.

2 Минимальное работоспособное приложение

Для компиляции программы можно использовать любой современный компилятор C++ (MSVC, GCC, Clang), также нужно подключить фреймворк Qt. Исходный код организован в виде проекта Visual Studio, и его можно скомпилировать или сконвертировать в другой формат проекта (например .pro для Qt Creator) с помощью расширения Qt VS Tools. Для запуска программы потребуется стандартная библиотека языка C++, установленная в системе, и библиотеки Qt, находящиеся в одной папке с исполняемым файлом. Минимальный необходимый набор (сборка для Windows): Qt5Core.dll, Qt5Gui.dll, Qt5Widgets.dll, platforms\qwindows.dll, styles\qwindowsvistastyle.dll. После запуска приложения можно или создать электрическую схему самому, или использовать какую-либо из приведённых в качестве примера.

3 Основное окно программы

Графический интерфейс основного окна содержится в файле `MainWindow.ui` (созданный в Qt Designer), а в классе `MainWindow` происходит обработка нажатий кнопок и элементов меню. Часть событий нажатий кнопок (приближение / отдаление, перемещение, изменение, переключение, удаление элемента, измерение напряжения и тока) передаются в виджет отрисовки схемы (`RenderArea`). Остальные обрабатываются в самом классе:

3.1 Новый файл

Функция `on_newFileAction_triggered` удаляет все элементы с помощью функции `clearElements` класса управления схемой (`ElectricalElementsManager`) и обновляет состояние виджета отрисовки схемы.

3.2 Открытие файла

Функция `on_openFileAction_triggered` открывает диалоговое окно выбора файла, открывает файл для чтения (если невозможно, то показывает сообщение об ошибке) и создаёт JSON-документ из файла, который передаётся в функцию `readElementsFromJson` класса управления схемой (при ошибках также показывается сообщение). После этого обновляется состояние виджета отрисовки схемы.

3.3 Сохранение в файл

Функция `on_saveFileAction_triggered` открывает диалоговое окно выбора файла, открывает файл для записи (если невозможно, то показывает сообщение об ошибке), создаёт JSON-документ, который передаётся в функцию `writeElementsToJson` класса управления схемой (при ошибках также показывается сообщение) и записывает его в файл.

3.4 Добавление элемента

Функция `on_addElementPushButton_clicked` либо отменяет размещение элемента (при повторном нажатии), вызывая функцию `startPlacingElement` с номером элемента `-1` у виджета отрисовки схемы, либо (из обычного состояния) открывает диалог добавления элемента (`AddElementDialog`).

3.5 Вычисление характеристик схемы

Функция `on_calculateCircuitPushButton_clicked` вызывает функцию вычисления характеристик схемы `calculateCircuit` у соответствующего класса (`ElectricalCircuitCalculator`) и, возможно, показывает сообщение об ошибке. При успешном вычислении обновляет состояние виджета отрисовки схемы.

3.6 Выполнено размещение добавляемого элемента

Функция `on_renderArea_elementPlaced` вызывается, когда в виджете отрисовки выбрано место для добавляемого элемента, и принимает выбранные положение и ориентацию. Функция получает выбранные в диалоге номер элемента и его свойства и пытается создать нужный элемент с помощью фабрики, полученной с помощью функции `getFactory` класса управления схемой. В случае успеха элемент добавляется с помощью функции `addElement` того же класса.

3.7 Выбран изменяемый элемент в виджете отрисовки

Функция `on_renderArea_elementEdited` вызывается, когда в виджете отрисовки выбран изменяемый элемент, и принимает положение и ориентацию выбранного элемента. Функция получает введенные в диалоге свойства и пытается обновить их с помощью функции `updateFromProperties` у элемента (класс `ElectricalElement`).

3.8 Выбраны точки для измерения напряжения в виджете отрисовки

Функция `on_renderArea_voltageMeasured` вызывается, когда в виджете отрисовки выбраны две точки для измерения напряжения, и принимает положения этих точек. Функция пытается получить напряжение с помощью функции `getVoltage` класса вычисления характеристик схемы. При отсутствии ошибок значение выводится в соответствующее поле.

3.9 Выбран элемент для измерения тока в виджете отрисовки

Функция `on_renderArea_currentMeasured` вызывается, когда в виджете отрисовки выбран элемент для измерения тока, и принимает положение и ориентацию этого элемента. Функция пытается получить ток с помощью функции

`getCurrent` класса вычисления характеристик схемы. При отсутствии ошибок значение выводится в соответствующее поле.

3.10 Изменилось состояние виджета отрисовки

Функция `on_renderArea_renderingStateChanged` вызывается, когда изменяется состояние отрисовки у соответствующего виджета. Функция обновляет состояния кнопок по следующему правилу: в обычном состоянии все кнопки активны, иначе активна только кнопка текущего состояния (для перехода в обычное).

4 Виджет отрисовки схемы

Виджет отрисовки схемы представляет для пользователя зону сборки, содержащую сетку точек, на которой показывается схема в текущем состоянии и размещаемый или выбираемый элемент.

4.1 Состояния отрисовки

`RenderingState` — это перечисление возможных состояний отрисовки: `Normal` (обычное), `Placing` (размещение добавляемого элемента), `Moving` (перемещение), `Editing` (изменение), `Toggling` (переключение), `Removing` (удаление), `MeasuringVoltage` (выбор точек для измерения напряжения), `MeasuringCurrent` (выбор элемента для измерения тока).

4.2 Константы

Константы для размеров сетки: `GRID_POINTS_DISTANCE` (расстояние между вершинами сетки), `GRID_BORDER_OFFSET` (отступ от края сетки), `GRID_POINTS_COUNT` (количество вершин по каждой стороне сетки), `GRID_POINT_SIZE` (длина стороны квадрата-точки сетки), `GRID_SIZE` (длина стороны всей сетки).

Также содержатся используемые кисти (чёрная, белая, красная, пурпурная, жёлтая, зелёная, синяя), перья (чёрное, красное, пурпурное, жёлтое, зелёное, синее) и перья с пунктирной линией (для рамок) (красное, пурпурное, жёлтое, зелёное, синее).

4.3 Вспомогательные функции

- `getGridCoordinates` — получение точки сетки по данным координатам (в пикселях);
- `getRealCoordinates` — получение координат (в пикселях) точки сетки;
- `calcZoomedCoordinates` — приведение масштабированных координат (указателя мыши) к изначальному масштабу;
- `findBrush` — нахождение кисти для состояния отрисовки;
- `findPen` — нахождение пера для состояния отрисовки;
- `findDashPen` — нахождение пера с пунктирной линией для состояния отрисовки.

4.4 Сигналы

- `elementPlaced` — выбрано место для размещаемого элемента;
- `elementEdited` — выбран изменяемый элемент;
- `voltageMeasured` — выбраны две точки для измерения напряжения;
- `currentMeasured` — выбран элемент для измерения тока;
- `renderingStateChanged` — изменилось состояние отрисовки.

4.5 Размеры виджета

Функции `sizeHint` и `minimumSizeHint` возвращают размер виджета (в пикселях) с учётом текущего масштаба.

4.6 Начало размещения элемента

Функция `startPlacingElement` вызывается из основного окна (при отмене размещения) или из диалога добавления элемента, изменяет текущее состояние отрисовки, обновляет номер выбранного элемента.

4.7 Начало изменения, переключения, удаления элемента, измерения тока

Функции `startEditingElement`, `startTogglingElement`, `startRemovingElement`, `startMeasuringCurrent` вызываются из основного окна и изменяют текущее состояние отрисовки.

4.8 Начало измерения напряжения

Функция `startMeasuringVoltage` вызывается из основного окна, изменяет текущее состояние отрисовки и сбрасывает выбранную точку сетки.

4.9 Масштабирование

Функции `zoomIn` и `zoomOut` изменяют масштаб от 2^{-2} (25%) до 2^2 (400%) ($2^0 = 1$ — изначальный масштаб) и обновляют виджет.

4.10 Нажатие кнопки мыши

Функция `mousePressEvent` обрабатывает нажатие кнопок мыши (начало клика). При нажатии левой кнопки устанавливается флаг `clicking`, при нажатии правой — изменяется ориентация размещаемого или выбираемого элемента.

4.11 Перемещение указателя мыши

Функция `mouseMoveEvent` обрабатывает перемещение указателя мыши — обновляет текущую точку, выбираемую указателем.

4.12 Отпускание кнопки мыши

Функция `mouseReleaseEvent` обрабатывает отпускание кнопок мыши — рассматривается только конец клика левой кнопкой мыши. Сначала обновляется точка, выбираемая указателем, и в зависимости от состояния отрисовки:

1. Обычное — ничего не происходит;
2. Размещение добавляемого элемента — если выбранное место для элемента пустое, то вызывается соответствующий сигнал (`elementPlaced`);
3. Перемещение элемента — если выбирается элемент для перемещения: проверяется, что в данном месте есть элемент, он удаляется из схемы и становится перемещаемым; если размещается перемещаемый элемент: если место пустое, то элемент добавляется на это место;
4. Изменение элемента — если в выбранном месте есть элемент, то открывается диалог изменения свойств (`EditElementDialog`) для этого элемента;
5. Переключение элемента — если в выбранном месте есть элемент и это выключатель, то изменить его состояние;
6. Удаление элемента — удалить элемент в выбранном месте (с помощью функции `removeElement` класса управления схемой);
7. Измерение напряжения — если нет выбранных точек, то текущая точка будет первой, иначе — второй, и будет вызван сигнал (`voltageMeasured`);
8. Измерение тока — если в выбранном месте есть элемент, то для него вызывается сигнал `currentMeasured`.

4.13 Отрисовка виджета

Функция `paintEvent` отрисовывает виджет: сначала сетка из точек, затем все элементы (у них вызывается `render`), после, в зависимости от состояния:

1. Размещение добавляемого или перемещаемого элемента — отрисовывается нужный элемент-образец или перемещаемый элемент в заданном указателем месте;
2. Измерение напряжения — отрисовывается выбираемая точка;
3. Выбор перемещаемого элемента, изменение, переключение, удаление элемента, измерение тока — элемент и рамка вокруг него отрисовыва-

ются с нужным цветом.

5 Диалог добавления элемента

Графический интерфейс диалогового окна добавления элемента содержится в файле `AddElementDialog.ui`, а в классе `AddElementDialog` происходит обработка действий в окне. При добавлении элемента сначала открывается этот диалог, пользователь выбирает нужный элемент и вводит свойства, и тогда начинается размещение элемента в зоне сборки.

5.1 Поля

1. `elementIndex` — номер элемента (в списке элементов-образцов класса управления схемой) или -1, если не выбран;
2. `elementProperties` — список строк, представляющих свойства выбранного элемента.

5.2 Выбран элемент в списке

Функция `on_elementsListWidget_itemSelectionChanged` вызывается, когда в списке выбран элемент. Функция очищает таблицу свойств и заполняет её для нового элемента.

5.3 Нажата кнопка «ОК»

Функция `on_okButton_clicked` вызывается, когда нажата кнопка «ОК». Функция заполняет поля выбранным элементом и его свойствами и закрывает диалог.

5.4 Прочее

1. `getElementIndex` — получение номера элемента;
2. `getElementProperties` — получение свойств элемента.

6 Диалог изменения элемента

Графический интерфейс диалогового окна изменения элемента содержится в файле `EditElementDialog.ui`, а в классе `EditElementDialog` происходит обработка действий в окне. При изменении элемента сначала в зоне сборки выбирается элемент, затем открывается этот диалог, пользователь вводит новые свойства, и тогда обновляется элемент.

6.1 Поля

`elementProperties` — список строк, представляющих свойства элемента.

6.2 Установка элемента

Функция `setElement` заполняет таблицу свойств для данного элемента.

6.3 Получение свойств элемента

Функция `getElementProperties` возвращает список свойств элемента.

6.4 Нажата кнопка «ОК»

Функция `on_okButton_clicked` вызывается, когда нажата кнопка «ОК». Функция заполняет список свойств и закрывает диалог.

7 Класс управления электрической схемой и элементами

Класс `ElectricalElementsManager` управляет получением, добавлением, удалением элементов из схемы, созданием схемы из JSON-документа и сохранением её в JSON, а также элементами-образцами, используемыми в списке в диалоге добавления элемента и при размещении добавляемого элемента.

7.1 Элементы-образцы

Поля:

1. `exampleElements` и `exampleElementsVertical` — элементы-образцы с горизонтальной и вертикальной ориентацией;
2. `exampleElementsNames` — названия элементов-образцов;
3. `exampleElementsIcons` — изображения элементов-образцов.

Функции:

1. `getExampleElement` — позволяет получить элемент-образец по номеру и ориентации;
2. `fillElementsList` — заполняет список элементами-образцами;
3. `fillPropertiesTable` — заполнение таблицы свойств элементом-образцом с заданным номером.

7.2 Фабрики элементов

Поле `elementsFactories` содержит все фабрики элементов, а функция `getFactory` позволяет получить нужную фабрику для элемента с заданным номером.

7.3 Добавление элемента

Функция `addElement` добавляет заданный элемент в схему.

7.4 Получение элемента

Функция `getElement` позволяет получить элемент схемы в заданной позиции и с данной ориентацией, если он существует.

7.5 Удаление элементов

Функция `removeElement` удаляет элемент из схемы, если он существует, и уничтожает его при необходимости. Функция `clearElements` удаляет и уничтожает все элементы схемы.

7.6 Чтение схемы из JSON-документа

Функция `readElementsFromJson` считывает все элементы из JSON-документа: документ интерпретируется как массив JSON-объектов, из которых создаются элементы с помощью функции `readJsonAndCreate` у фабрики соответствующего элемента.

7.7 Запись схемы в JSON-документ

Функция `writeElementsToJson` записывает все элементы в JSON-документ: для каждого элемента создаётся JSON-объект, из которых образуется массив.

8 Электрический элемент

Базовый класс электрического элемента — `ElectricalElement`, от которого наследуются все остальные элементы. Также для всех элементов есть фабрики, базовый класс — `ElectricalElementFactory`.

8.1 Поля

1. `location` — позиция центра-точки сетки;
2. `orientation` — ориентация элемента (горизонтальная или вертикальная);
3. `resistance` — сопротивление элемента (в Омах).

8.2 Конструктор

Конструктор создаёт элемент с заданными позицией, ориентацией и сопротивлением.

8.3 Обновление свойств

Функция `updateFromProperties` получает свойства элемента из списка строк, проверяет их корректность и обновляет элемент, если все свойства корректны.

8.4 Заполнение таблицы свойств

Функция `fillPropertiesTable` очищает, добавляет столбцы и заполняет нужные строки в таблице свойств элемента, которая используется в диалогах добавления (`AddElementDialog`) и изменения элемента (`EditElementDialog`).

8.5 Запись в JSON-объект

Функция `writeJson` записывает все свойства элемента в JSON-объект.

8.6 Получение крайних точек

Функция `getEndpoints` возвращает две крайние точки элемента: в горизонтальной ориентации — сначала левая, потом правая, в вертикальной — сначала верхняя, потом нижняя.

8.7 Отрисовка элемента

Функция `render` отрисовывает элемент в нужном состоянии отрисовки и с учётом свойств элемента. Функция используется в виджете отрисовки схемы.

8.8 Изображение элемента

Функция `getIcon` создаёт изображение элемента с помощью функции отрисовки. Это изображение используется для элементов-образцов в диалоге добавления элемента.

8.9 Обновление элемента при прохождении тока

Функция `onCurrentFlow` обновляет состояние элемента при прохождении через него тока. Используется для активации ламп и электронагревателей.

8.10 Прочее

1. `getLocation` — получение позиции элемента;
2. `getOrientation` — получение ориентации;
3. `setLocation` — установка позиции;
4. `setOrientation` — установка ориентации;
5. `getResistance` — получение сопротивления элемента;
6. `operator==` — оператор сравнения (по позиции и ориентации).

8.11 Фабрики элементов

8.11.1 Создание по позиции, ориентации и списку свойств

Функция `create` принимает позицию, ориентацию элемента и свойства в виде списка строк, проверяет их корректность и создаёт элемент.

8.11.2 Создание из JSON-объекта

Функция `readJsonAndCreate` принимает JSON-объект, получает из него позицию, ориентацию и свойства, проверяет их корректность и создаёт элемент.

9 Электрические элементы

В данной программе реализовано несколько электрических элементов:

9.1 Провод

Провод представлен классом `Wire`. Элемент не обладает дополнительными свойствами, а сопротивление всегда равно нулю.

9.2 Резистор

Резистор представлен классом `Resistor`. Элемент обладает заданным постоянным сопротивлением (строго больше нуля).

9.3 Выключатель

Выключатель представлен классом `Switch`. Это элемент, контакты которого можно замыкать или размыкать, поэтому он обладает дополнительным свойством `toggled` и функцией `toggle`, переключающей выключатель. В замкнутом состоянии ведёт себя как провод (нулевое сопротивление), в разомкнутом — как отсутствующий элемент (бесконечное сопротивление).

9.4 Электрический нагреватель

Электронагреватель представлен классом `ElectricHeater`. Это элемент, аналогичный резистору, но показывающий нагрев красным цветом при прохождении через него достаточного тока (0.1 A). Для этого используется поле `activated`, изменяемое функцией `onCurrentFlow` при обновлении характеристик схемы, и используемое в отрисовке.

9.5 Электрическая лампа

Электрическая лампа представлена классом `ElectricLamp`. Это элемент, аналогичный резистору, но показывающий излучение света жёлтым цветом при прохождении через него достаточного тока (0.1 A). Для этого используется поле `activated`, как и у электронагревателя.

9.6 Источник напряжения (ЭДС)

Источник напряжения представлен классом `VoltageSource`, обладающим дополнительным свойством `voltage` (напряжение) и использующим `resistance` для внутреннего сопротивления источника. Идеальный источник напряжения (при нулевом внутреннем сопротивлении) позволяет получить на

своих конечных точках заданное напряжение, не зависящее от тока, протекающего через источник. Реальный источник напряжения моделируется с помощью идеального источника и внутреннего сопротивления (указанного пользователем), включённых последовательно.

9.7 Источник тока

Источник тока представлен классом `CurrentSource`, обладающим дополнительным свойством `current` (ток) и использующим `resistance` для внутреннего сопротивления источника. Через идеальный источник тока (при бесконечном внутреннем сопротивлении) протекает заданный ток, вне зависимости от напряжения на его конечных точках. Реальный источник тока моделируется с помощью идеального источника и внутреннего сопротивления (указанного пользователем), включённых параллельно.

10 Класс вычисления характеристик схемы

Класс `ElectricalCircuitCalculator` позволяет вычислить характеристики схемы (напряжения, токи) и обновить состояния всех элементов (ламп, электронагревателей).

10.1 Функции, доступные извне

10.1.1 Вычисление характеристик схемы

Функция `calculateCircuit` запускает два этапа обработки схемы и, при неудаче, возвращает сообщение об ошибке.

10.1.2 Нахождение напряжения между двумя точками

Функция `getVoltage` проверяет, что данные точки принадлежат схеме, и вычисляет напряжение как разность потенциалов.

10.1.3 Нахождение тока через элемент

Функция `getCurrent` проверяет, что данный элемент является допустимым (нельзя вычислить ток для проводов и выключателей из-за нулевого или бесконечного сопротивления), и вычисляет ток с помощью закона Ома (по напряжению и сопротивлению), или обрабатывает особые случаи источников напряжения (для них токи вычисляются на втором этапе обработки схемы) и источников тока (у идеального источника ток постоянный, у неидеального рассчитывается с учётом внутреннего сопротивления).

10.2 Система непересекающихся множеств

СНМ используется на первом этапе обработки схемы. Для реализации используются поля `dsuParent` (номер предка для каждого элемента множества) и `dsuCompSz` (размер компоненты для каждого элемента множества) и функции `dsuLeader` (нахождение лидера компоненты по её элементу, используется эвристика сжатия путей) и `dsuUnite` (объединение двух компонент, используется ранговая эвристика на основе размеров поддеревьев).

10.3 Первый этап — нахождение вершин и элементов схемы

Нахождение вершин и элементов схемы реализовано в функции `findNodesAndElements`. Сначала очищаются все поля (СНМ, компоненты, элементы, источники напряжения и тока, потенциалы в вершинах схемы, токи

через источники напряжения). Затем каждой использованной точке схемы сопоставляются элементы, подключённые к ней. Затем инициализируется СНМ и с её помощью находятся компоненты связности (точки сетки, связанные проводами и замкнутыми выключателями), и каждой точке сопоставляется компонента и наоборот. Далее записываются пары (элемент, крайние точки) для подходящих элементов (с ненулевым и небесконечным сопротивлением), и источники напряжения и тока сохраняются в соответствующие им поля (для них также записываются крайние точки, а для источника напряжения — ещё и внутренняя средняя точка, через которую соединяются идеальный источник и внутреннее сопротивление).

10.4 Второй этап — нахождение напряжений и токов

Нахождение напряжений и токов реализовано в функции `findVoltagesAndCurrents`. Для этого используется алгоритм MNA, основанный на законах Кирхгофа (<https://lpsa.swarthmore.edu/Systems/Electrical/mna/MNA3.html>). Он состоит в решении уравнения $Ax = z$, где A — матрица, задаваемая сопротивлениями и вершинами подключения источников напряжения, x — вектор-столбец, содержащий напряжения во всех вершинах (компонентах точек) схемы и токи через источники напряжения, z — вектор-столбец, определяемый токами источников токов и напряжениями источников напряжений. Это уравнение можно представить как систему линейных алгебраических уравнений, а значит и решить с помощью алгоритма Гаусса (функция `solveLinearEquations`). После решения системы потенциалы в каждой вершине схемы и токи через источники напряжения сохраняются в соответствующих полях, и для всех допустимых элементов обновляется состояние в зависимости от тока, проходящего через них (для ламп и электронагревателей).