JMCAD (JMCADRTS, JMCADRTC)

1 Описание

Программный комплекс ***JMCAD*** предназначен для анализа динамики и проектирования самых разнообразных систем и устройств. По своим возможностям он является альтернативой аналогичным программным продуктам ***LabView***, ***Simulink***, ***VisSim***, ***МВТУ*** и др. Удобный редактор структурных схем, обширная библиотека типовых блоков и встроенный язык программирования позволяют реализовывать модели практически любой степени сложности, обеспечивая при этом наглядность их представления. Программный комплекс ***JMCAD*** успешно применяется для проектирования систем автоматического управления, следящих приводов и роботов-манипуляторов, тепловых энергетических установок, а также для решения нестационарных краевых задач (теплопроводность, гидродинамика и др.).

Широко используется в учебном процессе, позволяя моделировать различные явления в физике, электротехнике, в динамике машин и механизмов и т.д. Может функционировать в кластерах, в том числе и в режиме удаленного доступа к технологическим и информационным ресурсам.

Для пользователей удобство работы с ***JMCAD*** обусловлено также локализацией интерфейса на различные языки и наличием обширной документации.

Версии ***JMCAD*** доступны с исходными текстами ядра, библиотек и является открытой системой с полной документацией и набором демонстрационных примеров. Также в состав комплекса входят модули для обеспечения максимальной производительности и контроля в реальном времени (***JMCADRTS***, ***JMCADRTC***).

Программный комплекс ***JMCAD*** разработан с использованием языка ***Java*** (http://java.sun.com) и может быть использован в различных операционных системах (***Windows***, ***Linux***, ***Solaris***, ***Unix*** и т.д.).

2 Применение

Программный комплекс ***JMCAD*** реализует следующие режимы работы:

- МОДЕЛИРОВАНИЕ, обеспечивающий:
  - моделирование процессов в непрерывных, дискретных и гибридных динамических системах, в том числе и при наличии обмена данными с внешними программами и устройствами;
  - редактирование параметров модели в режиме «on-line»;
  - расчет в реальном времени или в режиме масштабирования модельного времени;
  - рестарт и воспроизведение результатов моделирования;
  - динамическую обработку сигналов.
- ОПТИМИЗАЦИЯ, позволяющий решать задачи:

  - минимизации (максимизации) заданных показателей качества;
  - нахождения оптимальных параметров проектируемой системы в многокритериальной постановке при наличии ограничений на показатели качества и оптимизируемые параметры.
- АНАЛИЗ, обеспечивающий:
  - расчет и построение характеристик статических и динамических систем;
  - расчет передаточных функций;
  - визуализацию результатов анализа статически и динамически.
- СИНТЕЗ, позволяющий конструировать регуляторы:
  - по заданным желаемым частотным характеристикам;
  - по заданному расположению доминирующих полюсов.
- КОНТРОЛЬ И УПРАВЛЕНИЕ, позволяющий создавать виртуальные прототиры:
  - пультов управления с измерительными приборами и управляющими устройствами;
  - мнемосхем с мультимедийными и анимационными эффектами.
  -

К достоинствам ***JMCAD*** относятся:

- открытость за счет с использованием языка ***Java*** и реализации нескольких механизмов обмена данными с внешними программами;
- возможность использовать в различных операционных системах (***Windows***, ***Linux***, ***Solaris***, ***Unix*** и т.д.);
- простота построения сложных моделей благодаря использованию вложенных структур, векторизации сигналов и алгоритмов типовых блоков, удобным средствам задания параметров и уравнений;
- эффективные численные методы;
- большое число обучающих и демонстрационных примеров с подробными комментариями.

3 Структура

Программный комплекс имеет три отдельных независимых блока ***JMCAD***, ***JMCADRTS***, ***JMCADRTC***. Каждый блок может работать самостоятельно, а также при создании распределенных систем можно использовать их совместно. Совместное использование позволяет создавать сложные распределенные системы с возможностью быстрого и легкого развития системы. Такая функциональная особенность позволяет вносить изменения в работающую систему без ее остановки. Для этого достаточно создать разветвление и дублирующий блок, на котором и производить разработку, а после окончания и тестирования этот блок включить в систему как основной. Заменив тем самым старый блок без остановки работы всей системы.

JMCAD – основной блок для создания и редактирования моделей (рис.

1). Также возможен вариант его использования для запуска модели в режиме работы (рис. 2). Запуск модели в режиме работы производится через командную строку с помощью параметра ***-single***.

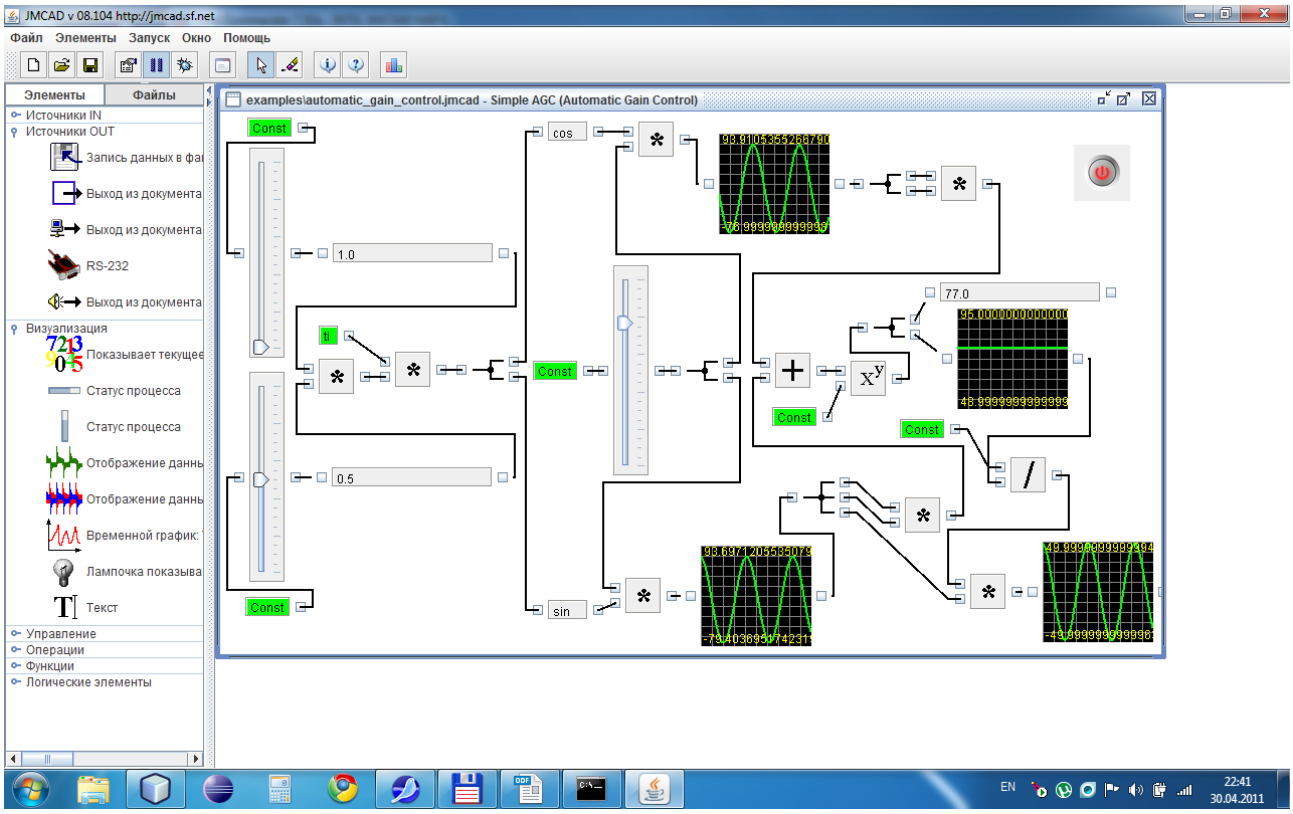

Рис. 1

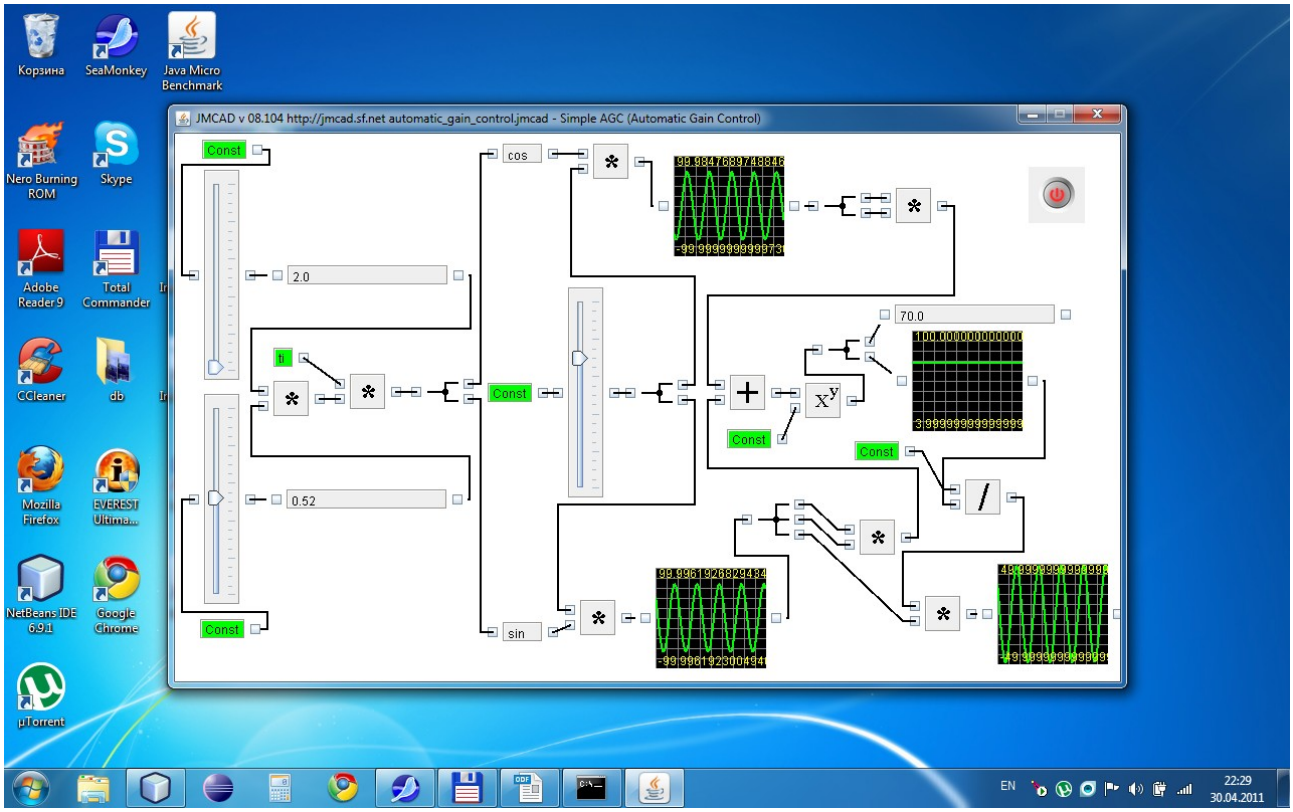

Рис. 2

JMCADRTS - блок для запуска модели в режиме работы (рис. 2). Запуск модели в режиме работы производится через командную строку.

JMCADRTC - блок для запуска интерфейса контроля и управления моделью (рис. 3). Запуск интерфейса контроля и управления моделью производится через командную строку.

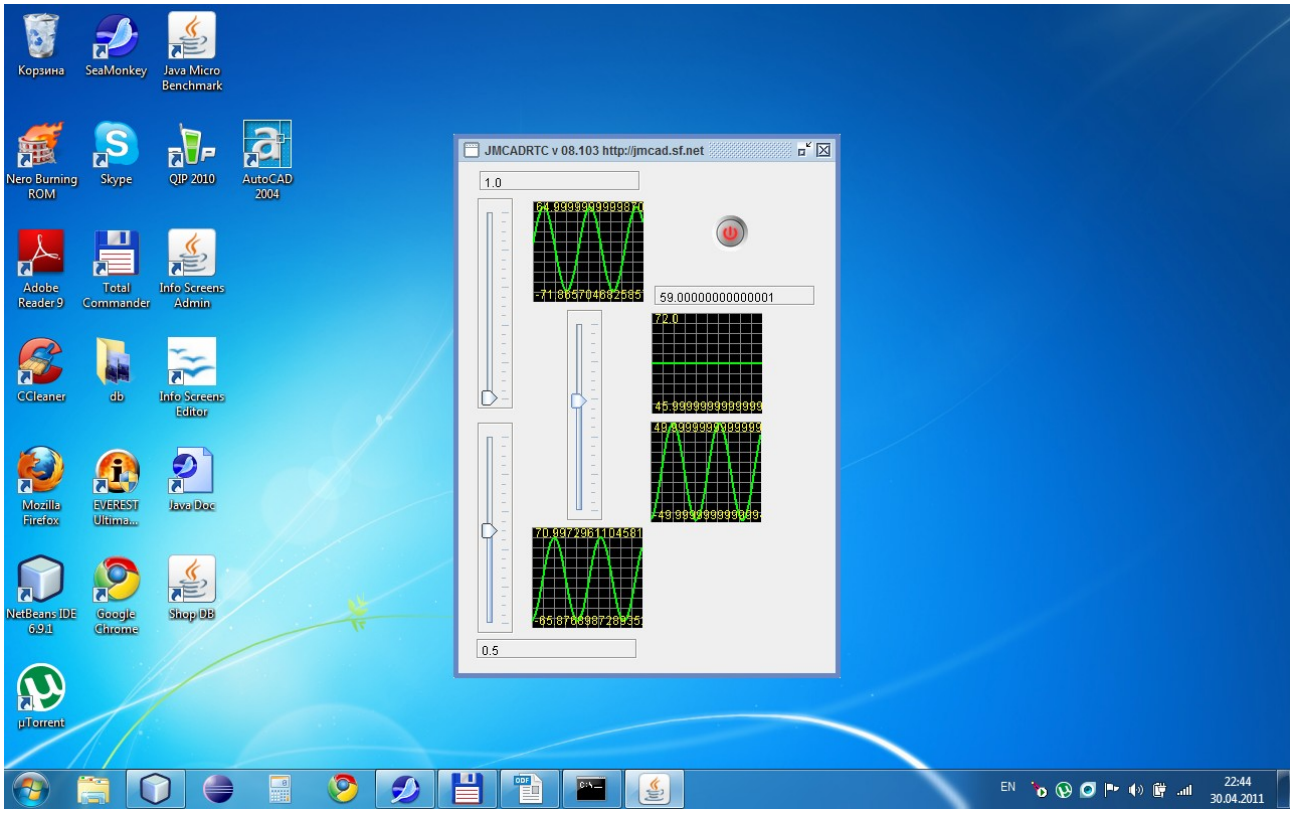

Рис. 3

4 Установка системы

Программный комплекс ***JMCAD*** может быть установлен на любой компьютер как локально так и с возможностью работы по сети.

Для установки системы ***JMCAD*** необходимо установить ***Java*** (http://java.sun.com/javase/downloads/). Рекомендуется устанавливать версию ***Java SE Runtime Environment 7 (JRE)*** или выше для запуска программного комплекса, а для разработки использовать версию ***Java SE Development Kit 7 (JDK)*** или выше.

Программный комплекс ***JMCAD*** можно загрузить с сервера http://jmcad.sf.net. На сервере доступны два варианта дистрибутива:

- ***JMCAD-XX.XXX-bin.zip*** – дистрибутив программного комплекса готовый для работы;
- ***JMCAD-XX.XXX-all.zip*** - дистрибутив программного комплекса для работы и разработчиков, содержащий исходные коды;

Для установки ***JMCAD*** необходимо разархивировать файл ***JMCAD-***

XX.XXX-bin.zip или ***JMCAD-XX.XXX-all.zip*** на жесткий диск. При разархивировании дистрибутива будет создана директория с тем же именем что и у файла архива. В директории будут находится файлы соответствующие назначению дистрибутива.

5 Запуск и использование

Для запуска программного комплекса используются файлы скриптов ****.bat*** (****.sh***). Примеры файлов скриптов для запуска ***JMCAD, JMCADRTS***, ***JMCADRTC*** находятся в директории ***examples***.

5.1 JMCAD

Для запуска программного комплекса ***JMCAD*** используются файлы скриптов ***__JMCAD.*.bat*** (***__JMCAD.*.sh***). Вместо символа * выбирается файл соответствующий операционной системе где будет использоваться программный комплекс.

Также возможен вариант его использования для запуска модели в режиме работы (рис. 2). Запуск модели в режиме работы производится через командную строку с помощью параметра ***-single***. Командная строка для запуска модели в режиме работы выглядит следующим образом:

java [parameters] <CLASSPATH> JMCAD -single <MODEL>

где:

[parameters] — параметры для виртуальной машины ***Java***. Для больших моделей при нехватке оперативной памяти необходимо указывать значение параметра ***-Xmx1000m***, который определяет доступный размер оперативной памяти для виртуальной машины ***Java***. Также можно использовать параметр ***-server*** для увеличения скорости расчета. Более детально о значениях и их применении возможно узнать из документации по виртуальной машине ***Java***. Данные параметры являются необязательными и могут не указываться.

<CLASSPATH> - используется для подключения используемых библиотек классов. Все библиотеки которые необходимы для работы модели должны быть доступны виртуальной машине ***Java***. Существует несколько вариантов указания виртуальной машине ***Java*** где искать библиотеки классов. Детально это описано в документации по виртуальной машине ***Java***. Для данного случая в строке описания обязательно должны присутствовать библиотеки которые необходимы для запуска и работы модели. Основной из всех библиотек классов есть библиотека находящаяся в ***jar*** архиве ***JMCAD.jar*** и содержащая класс для запуска ***JMCAD***.

<MODEL> - параметр командной строки который определяет файл модели (****.jmcad***) для запуска.

5.2 JMCADRTS

Запуск модели в режиме работы производится через командную строку. Командная строка для запуска модели в режиме работы выглядит следующим образом:

java [parameters] <CLASSPATH> JMCADRTS <MODEL>

где:

[parameters] **—** параметры для виртуальной машины ***Java***. Для больших моделей при нехватке оперативной памяти необходимо указывать значение параметра ***-Xmx1000m***, который определяет доступный размер оперативной памяти для виртуальной машины ***Java***. Также можно использовать параметр ***-server*** для увеличения скорости расчета. Более детально о значениях и их применении возможно узнать из документации по виртуальной машине ***Java***. Данные параметры являются необязательными и могут не указываться.

<CLASSPATH> - используется для подключения используемых библиотек классов. Все библиотеки которые необходимы для работы модели должны быть доступны виртуальной машине ***Java***. Существует несколько вариантов указания виртуальной машине ***Java*** где искать библиотеки классов. Детально это описано в документации по виртуальной машине ***Java***. Для данного случая в строке описания обязательно должны присутствовать библиотеки которые необходимы для запуска и работы модели. Основной из всех библиотек классов есть библиотека находящаяся в ***jar*** архиве ***JMCADRTS.jar*** и содержащая класс для запуска ***JMCADRTS***.

<MODEL> - параметр командной строки который определяет файл модели (****.jmcad***) для запуска.

5.3 JMCADRTC

Запуск интерфейса контроля и управления моделью производится через командную строку которая выглядит следующим образом:

java [parameters] <CLASSPATH> JMCADRTC <MODEL>

где:

[parameters] **—** параметры для виртуальной машины ***Java***. Для больших моделей при нехватке оперативной памяти необходимо указывать значение параметра ***-Xmx1000m***, который определяет доступный размер оперативной памяти для виртуальной машины ***Java***. Также можно использовать параметр ***-server*** для увеличения скорости расчета. Более детально о значениях и их применении возможно узнать из документации по виртуальной машине ***Java***.

Данные параметры являются необязательными и могут не указываться.

<CLASSPATH> - используется для подключения используемых библиотек классов. Все библиотеки которые необходимы для работы модели должны быть доступны виртуальной машине ***Java***. Существует несколько вариантов указания виртуальной машине ***Java*** где искать библиотеки классов. Детально это описано в документации по виртуальной машине ***Java***. Для данного случая в строке описания обязательно должны присутствовать библиотеки которые необходимы для запуска и работы модели. Основной из всех библиотек классов есть библиотека находящаяся в ***jar*** архиве ***JMCADRTC.jar*** и содержащая класс для запуска ***JMCADRTC***.

<MODEL> - параметр командной строки который определяет файл модели (****.jmcad***) для запуска.

6 Разработка

Программный комплекс ***JMCAD*** является открытой системой с возможностью разработки новых модулей. Система разработана с использованием языка ***Java*** (http://java.sun.com), что позволяет использовать ее в различных операционных системах (***Windows***, ***Linux***, ***Solaris***, ***Unix*** и т.д.).

В основе системы находится ядро которое контролирует работу системы. Ядро отвечает за запуск, работу и остановку модели. В программном комплексе используется три ядра - ***JMCAD, JMCADRTS***, ***JMCADRTC***. Принцип работы заключается в том что ядро контролирует передачу данных между элементами, а все действия в элементе выполняются в автономно. Ядро синхронизирует передачу данных между элементами и работает в три этапа (запуск, работа, остановка):

- ***запуск*** — ядро запускает метод ***calc_pre();*** для всех элементов модели. Метод ***calc_pre();*** используется для подготовки элемента к работе;
- ***работа*** — ядро запускает метод ***calc();*** в цикле для всех элементов модели с указанной задержкой. Для элементов которые являются источниками вызывается метод ***start(long t0, long ti, long dt);***. Если указано значение 0 то система работает в реальном времени;
- ***остановка*** — ядро запускает метод ***calc_post();*** для всех элементов модели. Метод ***calc_post();*** используется для подготовки элемента к остановке работы.

6.1 Структура классов

В основу системы положен модульный принцип. Такой подход позволяет легко разрабатывать новые элементы. Назначение и функции элементов системы распределяются по различным файлам:

- ***JMCAD*.java*** — ядро системы и графический интерфейс;

- ***JMCAD_Internationalize_xx_XX.properties*** — локализация интерфейса. В место символов ***xx*** и ***XX*** указываются символы определяющие страну и язык;
- ***JMCAD.menu*** — меню визуальных элементов;
- ***_*.java*** — визуальные элементы. Также для удобства файлы которые относятся к данному визуальному элементу имеют аналогичное имя.

6.2 Компиляция

Для компиляции и упаковки всех классов в ***jar*** архив используется пакетный файл ***__make_jar.bat*** (***__make_jar.sh***). Эти пакетные файлы находятся для каждого блока в директории с исходными файлами данного блока:

- ***src – JMCAD*** (***JMCAD.jar***);
- ***src_rts – JMCADRTS*** (***JMCADRTS.jar***);
- ***src_rtc – JMCADRTC*** (***JMCADRTC.jar***).

6.3 Создание визуальных элементов

Все модели в ***JMCAD*** создаются из визуальных элементов, которые определяют эффективность этой модели. В системе присутствует большой набор стандартных типовых элементов. Но всегда есть объективная необходимость увеличивать набор стандартных типовых элементов, добавляя новые элементы или модифицируя уже имеющиеся. Для этого в программном комплексе ***JMCAD*** предусмотрен простой способ создания новых элементов.

Для создания базовой структуры визуального элемента используется наследование базового класса для всех визуальных элементов ***JMCADObject***. Этот класс содержит переменные и методы необходимые для создания новых визуальных элементов.

Основные переменные:

- ***in*** — массив входных значений. По умолчанию размер массива равен 0. Размер массива определяет количество входов в элемент. Приходящее значение на вход сохраняется в ячейке массива соответствующее индексу входа;
- ***out*** — массив выходных значений. По умолчанию размер массива равен 0. Размер массива определяет количество выходов с элемента. После того как на все входы элемента ***in*** получены значения запускается метод ***calc();*** который выполняет описанные в нем действия и после чего все значения находящиеся в массиве ***out*** передаются на другие элементы;
- ***in_text*** — массив имен входных значений. Размер массива должен совпадать с массивом входных значений ***in***. Имена описываются латинскими буквами и цифрами. Должны начинаться с буквы и не содержать спецсимволов;

- ***out_text*** — массив имен выходных значений. Размер массива должен совпадать с массивом выходных значений ***out***. Имена описываются латинскими буквами и цифрами. Должны начинаться с буквы и не содержать спецсимволов;
- ***ToolTipText*** — текстовая подсказка появляющаяся при наведении курсора мыши на элемент;
- ***w, h*** — ширина и высота элемента. Значения задаются в конструкторе;
- ***isGeneranor*** — переменная определяет тип элемента стандартный или источник. По умолчанию переменная имеет значение ***false***, что определяет элемент как стандартный и ядро запускает метод ***calc();***. При указании переменной значения ***true*** элемент имеет тип источника и ядро запускает метод ***start(long t0, long ti, long dt);***
- ***isVisual*** — переменная определяет видимость элемента при использовании его в интерфейсе контроля и управления моделью (рис. 3). По умолчанию имеет значение ***false***, что делает элемент невидимым в интерфейсе контроля и управления моделью;
- ***panel_c*** — центральная панель элемента;
- ***edit_panel*** — панель создания графического интерфейса, который может использоваться для редактирования свойств элемента. При двойном щелчке на элементе, в режиме редактирования модели, открывается диалоговое окно содержащее эту панель.

Основные методы:

- ***calc_pre()*** — метод используется для подготовки элемента к работе;
- ***calc()*** — метод будет вызываться на каждом шаге работы модели с указанной задержкой. Для элементов которые являются источниками вызывается метод ***start(long t0, long ti, long dt)***. Если указано значение 0 то система работает в реальном времени;
- ***calc_post()*** — метод используется для подготовки элемента к остановке работы;
- ***start(long t0, long ti, long dt)*** — метод для элементов которые являются источниками и будет вызываться на каждом шаге работы модели с указанной задержкой. Если указано значение 0 то система работает в реальном времени;
- ***edit_pre()*** — метод используется для создания графического интерфейса при редактирования свойств элемента. Графический интерфейс располагается на панели ***edit_panel***, которая потом отображается при двойном щелчке на элементе в диалоговом окне.
- ***edit_post()*** — метод вызывается если при изменении свойств элемента в диалоговом окне была нажата кнопка подтверждения изменений;
- ***paint_info(Graphics g)*** — метод обеспечивает доступ к графическим методам данного элемента;
- ***parse(String pst)*** — метод обеспечивает считывание данных из модели. Тело метода должно начинаться со строки ***super.parse(pst)***;

- ***write(RandomAccessFile fout)*** — метод обеспечивает сохранение параметров элемента в модели. Тело метода должно начинаться со строки ***super.write(fout)***.

6.4 Создание и локализация документации

Программный комплекс ***JMCAD*** содержит алгоритм автоматического отображения документации для каждого элемента и формирует общую документацию.

При двойном щелчке на элементе, в режиме редактирования модели, открывается диалоговое окно содержащее кнопку вызова окна с документацией по данному элементу.

Для отображение документации в директории ***help*** должен находиться файл с документацией по данному элементу. Файл имеет формат представления данных ***HTML*** и должен иметь такое же имя как класс элемента. Для локализации документации к имени файла добавляется префикс ***_xx_XX***. Где ***xx*** определяющий страну, а ***XX*** определяет язык.

6.5 Локализация интерфейса пользователя

Локализация интерфейса производится стандартными методами языка ***Java***. Строки для локализации находятся в файлах для локализации интерфейса пользователя ***JMCAD_Internationalize_xx_XX.properties***. Где в место символов ***xx*** и ***XX*** указываются символы определяющие страну и язык. Более детально о методики локализации в ***Java*** можно узнать из документации по программированию на ***Java***.