

# **PRADIS**

**РАБОТА С ПРЕПРОЦЕССОРОМ PRADIS QUCS**

**ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ АВТОМАТИЗАЦИИ  
МОДЕЛИРОВАНИЯ НЕСТАЦИОНАРНЫХ ПРОЦЕССОВ  
В МЕХАНИЧЕСКИХ СИСТЕМАХ И СИСТЕМАХ ИНОЙ  
ФИЗИЧЕСКОЙ ПРИРОДЫ**

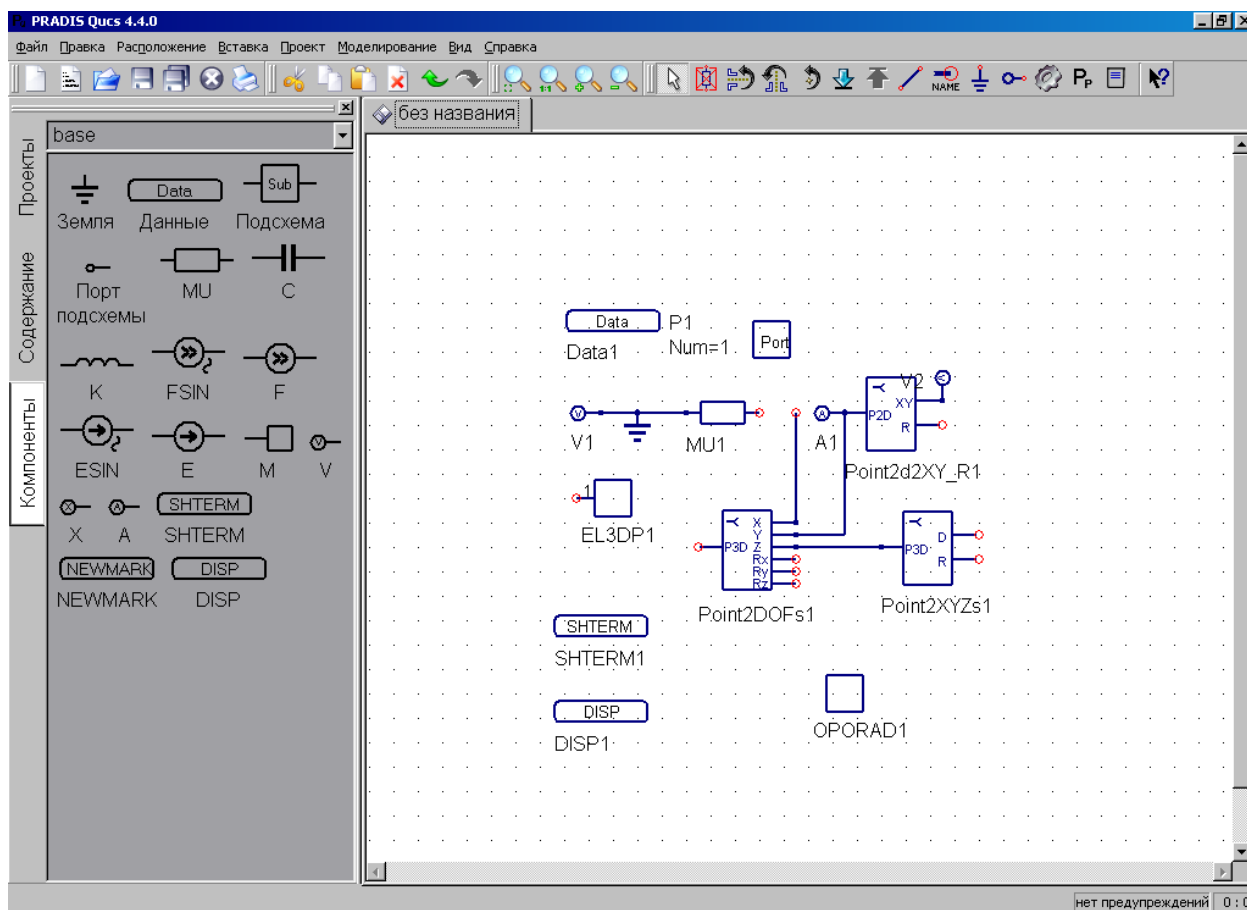
**ВЕРСИЯ 4.4**

## Содержание

1. Введение.....	3
2. Описание схемного редактора.....	3
2.1. Модули.....	3
2.2. Понятие узла в препроцессоре.....	4
2.3. Имя.....	4
2.4. Схема.....	4
2.5. Элементы.....	4
2.6. Специальные элементы.....	5
2.7. Ground (Земля).....	5
2.8. Data (Данные).....	5
2.9. Подсхема .....	5
2.10. Порт подсхемы.....	6
2.11. PSL Include файл (PSLInclude).....	6
2.12. PSL фрагмент (PSLSubScheme).....	7
2.13. Модели.....	7
2.14. ПРВП.....	8
2.15. ПГО.....	9
2.16. Объекты.....	10
2.17. SHTERM и NEWMARK .....	11
2.18. DISP .....	11
2.19. Сплиттеры или расщепители.....	11
3. Работа с подсхемами.....	12
3.1. Создание подсхемы.....	12
3.2. Создание схемного обозначения подсхемы.....	12
3.3. Назначение параметров подсхемы.....	13
4. Моделирование.....	13
5. Справка.....	14
6. Возможные ошибки при работе.....	14
6.1 Ошибки, обнаруживаемые препроцессором.....	14
6.2 Ошибки, обнаруживаемые транслятором Python.....	14
6.3 Ошибки, обнаруживаемые транслятором PSL.....	15
7. Рекомендации к использованию.....	15

## 1. Введение

Препроцессор PRADIS Qucs предназначен для создания схем на основе моделей PRADIS. Препроцессор обладает богатыми возможностями для редактирования схем и работает напрямую с системным каталогом PRADIS. По созданной схеме препроцессор генерирует файл задания на языке PPL, которые затем запускается на расчет.



## 2. Описание схемного редактора

### 2.1. Модули

Для удобства и все элементы системного каталога распределяются по модулям. Всегда есть базовый модуль base. Остальные модули могут зависеть от версии или создаваться самим пользователем. Важно, что может быть только один уровень вложенности модулей. В модуле можно выделить такие элементы:

- Модели
- ПРВП
- ПГО
- Объекты
- Узлы
- Параметры

В препроцессоре пользователь оперирует моделями, ПРВП, ПГО и объектами.

Модули в списке модулей описываются в том же порядке, в каком они описаны в системном каталоге.

Подробнее информацию по модулям можно получить из описания системного каталога.

## 2.2. Понятие узла в препроцессоре

Важное отличие от старых версий PRADIS – в определении узла. В новой версии PRADIS необходимо различать понятие узел и степень свободы. Узел – это набор степеней свободы, а не одна степень свободы.

По умолчанию в версии 4.4 в модуле base есть следующие типы узлов:

- DOF1 – одна степень свободы (соответствует старому понятию узла);
- Point – пространственный узел с поступательным и вращательным движением (3 поступательные и 3 вращательные степени свободы);
- Point2d – плоский узел с поступательным и вращательным движением (2 поступательные степени свободы и 1 вращательная);
- XY – плоский узел с 2-я поступательными степенями;
- XYZ – пространственный узел с 3-мя поступательными степенями.

Соответственно, выход модели может нести в себе несколько степеней свободы. Недопустимо соединять напрямую между собой разнотипные узлы – в лучшем случае препроцессор выведет ошибку при генерации PPL. В худшем случае, ошибка вообще не обнаружится и будет неправильно посчитана задача. Диагностика ошибки данного типа может происходить на всех этапах генерации задания.

Если необходимо выделить какую-то одну степень свободы или набор степеней из сложного узла, то используются, так называемые, расщепители или сплиттеры. Более детально смотрите главу Сплиттеры.

Для лучшего визуального восприятия типа узла необходимо при создании элементов маркировать цветом выводы элементов согласно следующему стандарту:

- DOF1 - темно-синий #000080
- Point - зеленый #00FF00
- Point2d – синий #0000FF
- XY - светло-синий #0080FF
- XYZ - светло-зеленый #00FF80

Пользователь может дать какому-либо узлу собственное имя и использовать его в дальнейшем в переменных. По умолчанию имена называются так: `_netN`, где N- целое число.

## 2.3. Имя

Любое имя в препроцессоре (элементов, схемы и т.д.) не должно содержать кириллицы и специальных символов (пробелы, запятые, тире и т.д.) и не начинаться с цифры. Это ограничение связано с тем, что по данному имени генерируется переменная на языке Python.

## 2.4. Схема

Схема или фрагмент. Важно корректно задать имя файл схемы. Вообще схема, как и любой ее элемент, является объектом в задании, которое генерируется для расчета. Поэтому имя схемы должно удовлетворять общим правилам.

## 2.5. Элементы

Схема состоит из элементов соединенных между собой узлами. Элемент может быть как непосредственно какой-либо моделью, ПРВП, ПГО, или вспомогательным.

Для элементов в зависимости от типа задаются различные свойства. Но почти для всех элементов требуется ввести имя элемента. Имя элемента генерируется автоматически из типа элемента. Но пользователь может изменить его, важно соблюдать правила именования.

## 2.6.Специальные элементы

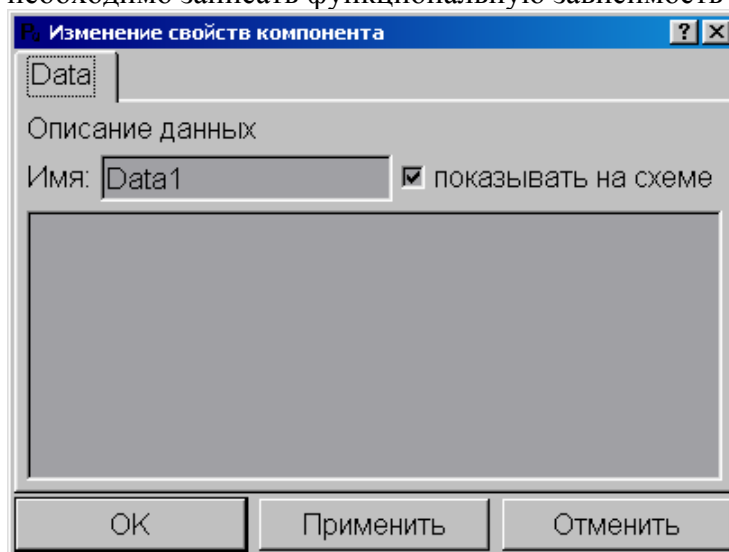
Специальные элементы не входят в системный каталог, а предназначены для создания схемы, определения ее свойств, вспомогательных функций.

## 2.7.Ground (Земля)

Элемент «База» или «Земля» предназначен для определения базовых узлов схемы.

## 2.8.Data (Данные)

Предназначен для задания переменных схемы и их значений. Удобно использовать, когда для разных элементов схемы задается один и тот же параметр с одинаковым значением или когда необходимо записать функциональную зависимость параметров.



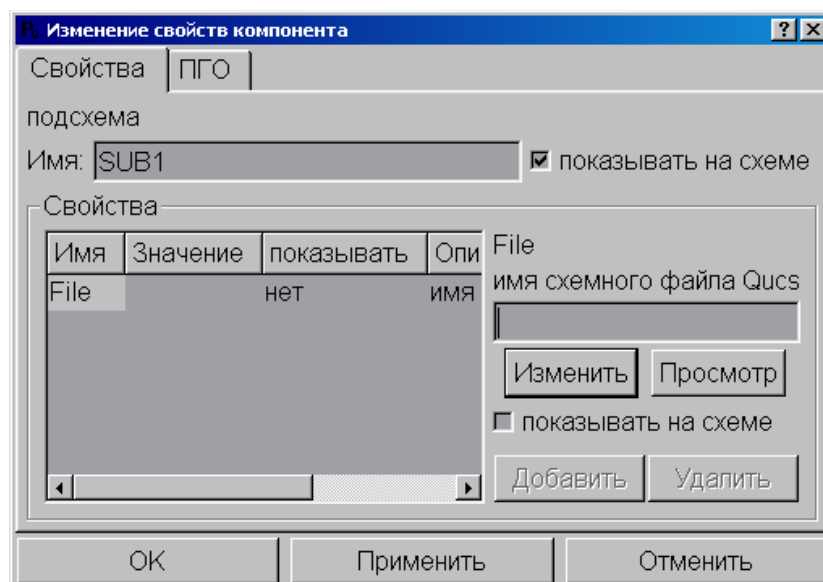
Требуется ввести текст на языке Python (PPL), в котором определяются требуемые переменные и их зависимость. А в элементах в параметры вместо значений вводятся названия переменных.

Введенный текст затем будет подставляться в задание на PPL сразу после описания подсхем или операторов `import`, но гарантированно перед описанием схемы.

Может быть несколько элементов `Data`. Тогда они располагаются в тексте программы в порядке добавления их в схему.

## 2.9.Подсхема

Элемент добавляет в схему фрагмент из другого файла. Необходимо ввести имя файла фрагмента. После задания файла автоматически загрузится рисунок данной подсхемы с внешними узлами.



С помощью команд «Войти в подсхему» и «Выйти» можно входить в подсхему.

При генерации текста задания подсхема будет генерироваться прямо в текст задания основной схемы, перед описанием основной схемы.

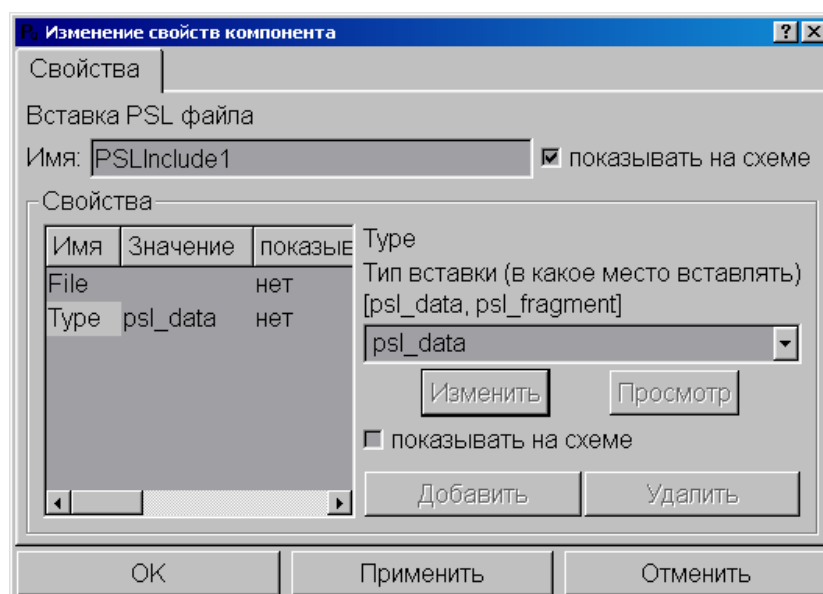
Подробнее о создании подсхем смотрите главу Создание подсхем.

## 2.10. Порт подсхемы

Элемент определяет внешние узлы текущей схемы. Подключается к узлу, который должен быть внешним.

## 2.11. PSL Include файл (PSLInclude)

Элемент позволяет вставить оператор Include для добавления какого-либо файла в PSL задание. В параметрах элемента необходимо указать файл, который вставляется и тип вставки. Тип вставки указывает, в какое место PSL задания требуется добавить оператор Include.



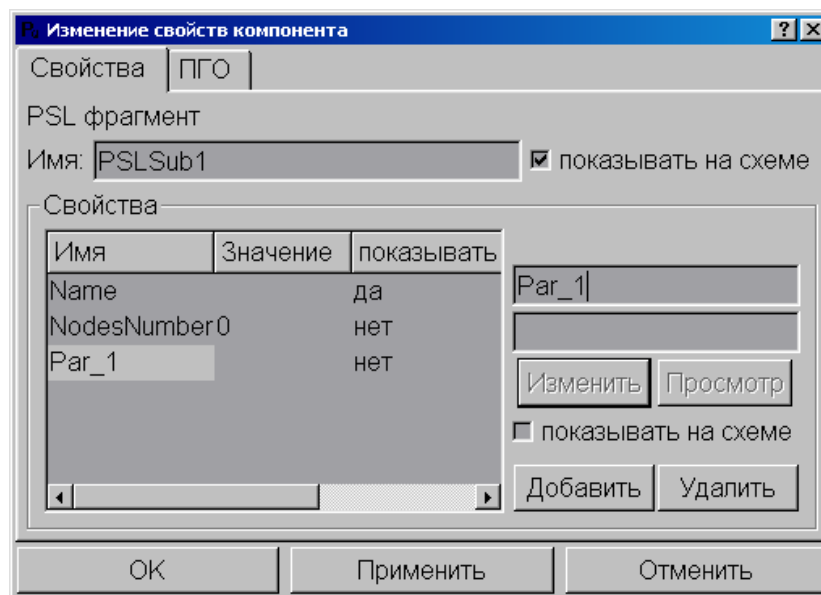
Тип psl\_data добавит файл в начало блока \$DATA, как правило, используется для добавления описания параметров. Тип psl\_fragment добавит файл перед началом блока \$FRAGMENT, это служит для добавления файлов с описанием фрагментов на PSL. Это позволит использовать в схеме элемент PSLSubScheme (PSL фрагмент).

## 2.12.PSL фрагмент (PSLSubScheme)

Элемент добавляет в схему фрагмент, описанный в стороннем PSL файле. Сам PSL файл должен быть добавлен с помощью элемента PSLInclude.

Необходимо определить имя фрагмента, число узлов и задать параметры.

Так как PSL файл не анализируется, то его число узлов и их тип неизвестен. Необходимо определить число узлов, причем необязательно, чтобы это были отдельные степени свободы подсхемы, т.е. DOF1. Тип узла определяется в дальнейшем типом узла у элемента, который подключается к данному узлу фрагмента. Поэтому все узлы фрагмента должны быть подключены к элементам с явным описанием типа узлов. Иначе будет ошибка при трансляции PPL файла, а она тяжело идентифицируется.



Для фрагмента можно определить параметры. Для параметра определяется имя и значение. Для этого необходимо выбрать переменную Par\_1. при этом появится возможность добавить или изменить параметр, введя имя, значение и нажав кнопку Добавить, либо Применить. Параметры можно удалять с помощью кнопки Удалить.

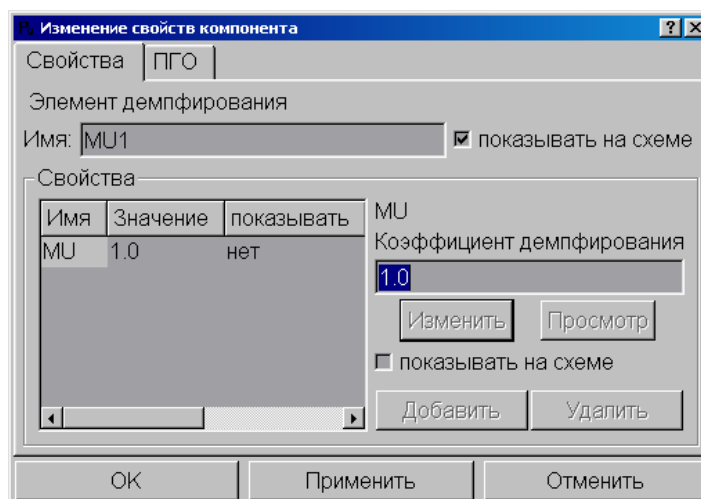
Остальные элементы в препроцессоре генерируются непосредственно из системного каталога и в дальнейшем создаются в файле задания.

## 2.13.Модели

Модели распределены по модулям. Базовые модели включены в модуль base. Остальные модели распределяются по модулям исходя из интересов пользователя или разработчика моделей.

Модели в списке идут в том же порядке, как и в описании модуля в системном каталоге (т.е. в порядке добавления в системный каталог).

Для модели можно определить имя, свойства и ПГО.



В свойствах для какого-либо параметра задается его значение. При этом значение может быть выражением на языке Python. В дальнейшем эта строка без изменений будет вставлена в задание RPL. Таким образом, можно задавать списки для сложных параметров (например, координаты точек), можно задавать переменные из блока Data, можно задавать функциональную зависимость от параметров элемента Data.

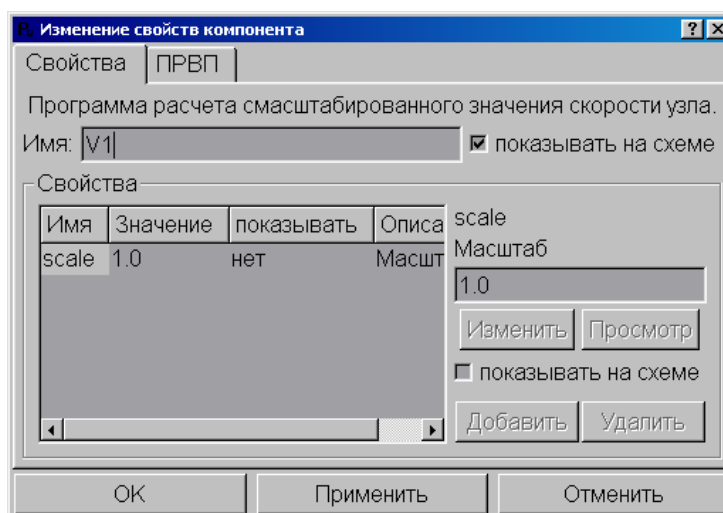
Свойство задается по нажатию Enter или кнопки Применить (или ОК). В случае нажатия Enter курсор переходит на следующий параметр или закрывает диалог, если все параметры введены.

Для параметра можно включить флаг отображения данного параметра в схеме.

На закладке ПГО задаются ПГО и ее параметры для отображения данного элемента схемы. Подробнее об этом диалоге в описании ПГО.

## 2.14.ПРВП

ПРВП является сложным объектом, поскольку в нем совмещаются различные свойства. Не все свойства ПРВП PRADIS могут задаваться в данной версии препроцессора.



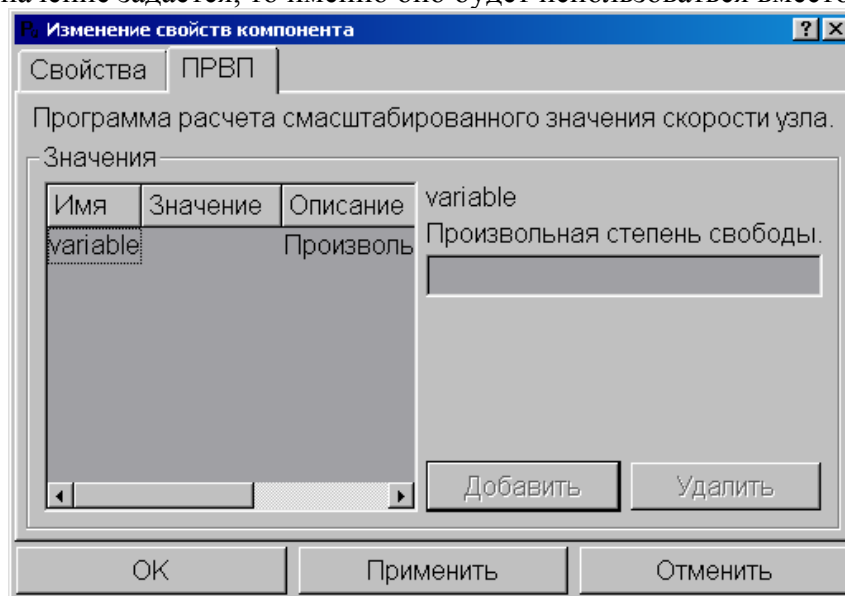
Параметры ПРВП на закладке Свойства задаются аналогично параметрам модели.

Входными переменными для ПРВП могут быть узлы (точнее перемещения, скорости и ускорения степеней свободы), значения сил, рабочих переменных и переменных состояния моделей.

Каким образом отличается, какая входная переменная подключена к узлу, а какая к другому типу переменной? По умолчанию, все входные переменные ПРВП являются



узлами. И, соответственно, выводы ПРВП подключаются к каким-либо узлам схемы. Но для каждой входной переменной ПРВП на закладке ПРВП можно задать нужное значение. При этом, если значение задается, то именно оно будет использоваться вместо узла.



Причем выражение для переменной должно соответствовать выражению для переменной ПРВП в RPL. Т.е. могут быть следующие значения:

net1 – имя узла. Т.е. в ПРВП будет передаваться узел;

net1.X(), net1.V(), net1.A() – в ПРВП передаются перемещение, скорость или ускорение узла соответственно;

Model1.I(1) – передается ток с 1-го вывода модели Model1;

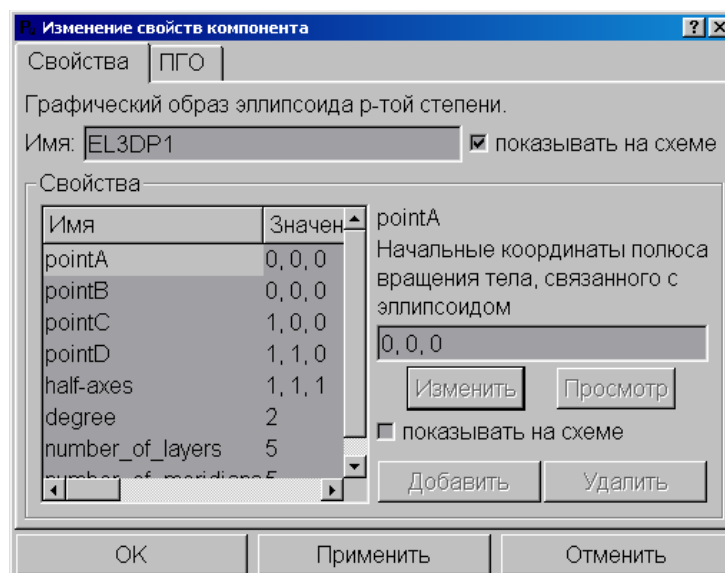
Model1.W(5) – передается 5-я рабочая переменная модели Model1;

Model1.S(10) – передается 10-я переменная состояния.

Необходимо быть осторожными в случае сложных узлов (более 1-й степени свободы). При передаче они разворачиваются в список, т.е. набор степеней свободы. Таким образом, например, узел типа Point – это 6 входных переменных. Это важно учитывать, поскольку это не отслеживается на уровне препроцессора, и даже на уровне RPL. Сообщение об ошибке будет выведено только транслятором PSL, а на этом уровне будет довольно сложно разобраться в ошибке.

## 2.15.ПГО

ПГО в PRADIS могут, как принадлежать моделям, так и быть независимыми. ПГО, принадлежащие моделям, назначаются в диалоге для моделей. Рассмотрим независимые ПГО.

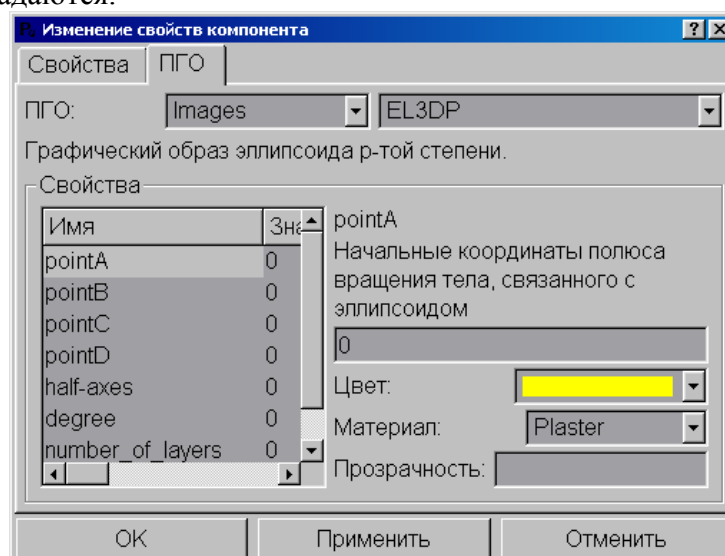


На панели свойства задаются параметры ПГО, так же как и для обычных моделей.

Важно, что параметры для ПГО, привязанных к модели, задаются на панели ПГО в диалоге моделей!

Панель ПГО позволяет выбрать тип ПГО (модуль и ПГО), задать параметры ПГО и задать параметры слоя. Для независимых ПГО тип уже определен, а параметры задаются на панели свойства.

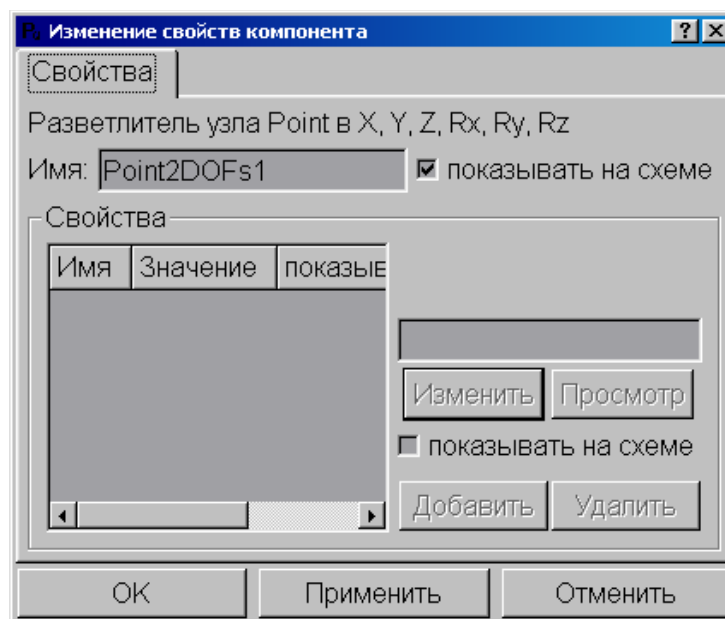
Рассмотрим параметры слоя. Это цвет, материал и прозрачность. Может быть 14 цветов, согласно PRADIS, 8 видов материалов и прозрачность от 0 до 1. Такие параметры, как угол зрения, не задаются.



## 2.16.Объекты

В PPL был введен новый тип элемента – объекты. Объект – это либо оператор (SHTERM, DISP и т.д.), либо объект несущий вспомогательную функцию, например, какое-либо преобразование. В PPL он представляется в виде элемента класса объекта.

В препроцессоре объект является элементом со свойствами, имеет узлы и параметры. Зкладка Свойства аналогична такой же закладке для моделей. На ней задаются параметры объекта.



К узлам объекта могут подключаться любые другие объекты. В PPL в объект будет передаваться список узлов для обработки.

При генерации PPL генерируется импорт модуля объекта и при создании объекта в него передаются список узлов и параметров.

Существуют специальные виды объектов. Они фиксированы и это: SHTERM, NEWMARK и DISP.

## 2.17.SHTERM и NEWMARK

Эти объекты описывают операторы раздела \$RUN в PSL. Соответственно, они отражают свойства оператора RUN\$. А именно, на закладке Вывод задается список выходных переменных с интервалами, а на закладке Свойства задаются параметры операторов.

На закладке Вывод представлен список выходных переменных. Новые переменные можно добавить из выпадающего списка. Причем выпадающий список формируется автоматически исходя из наличия ПРВП в схеме в данный момент. После выбора выходной переменной в списке, можно задать ее интервал и добавить в список командой Добавить. Удалить переменную из списка можно с помощью кнопки Удалить, предварительно выделив нужную переменную.

На панели Свойств задаются параметры операторов, такие как конечное время интегрирования, настройки решателя.

## 2.18.DISP

Этот объект определяет свойства оператора DISP раздела \$PRINT. Диалог аналогичен диалогу операторов SHTERM и NEWMARK.

Важно, что в случае если ни одна выходная переменная не будет задана на панели Вывод, то транслятор PSL выдаст ошибку.

## 2.19.Сплиттеры или расщепители

Расщепители являются объектами класса splitters. Они предназначены для выделения отдельных степеней свободы в сложных узлах. На уровне PPL они ассоциируют степени свободы узлов друг с другом так, чтобы в PSL эти степени свободы были одинаковыми.

Сейчас все сплиттеры помещены в один модуль splitters. Классы сплиттеров на PPL помещены в пакете pradis.splitters.

Пользователь может разработать свои сплиттеры при необходимости.

Если пользователь соединяет некорректно узлы, то может быть несколько вариантов:

- ошибка возникнет при попытке сгенерировать PPL файл
- ошибка возникнет при запуске PSL файла на расчет
- ошибка не возникнет (но расчет будет неверным) в случае путаницы узлов DOF1, например, механического и гидравлического узла.

### 3. Работа с подсхемами

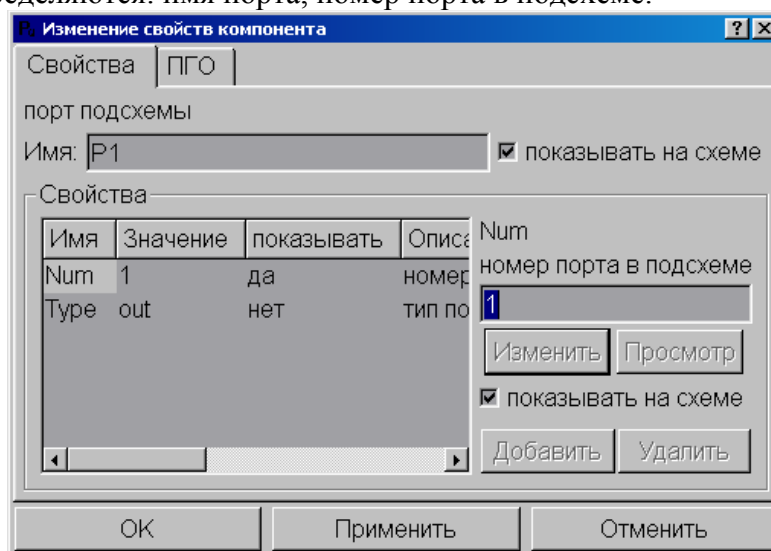
Работа с подсхемами включает в себя несколько этапов:

- создание схемы
- создание схемного обозначения подсхемы
- назначение параметров подсхемы
- использование подсхемы в схеме

#### 3.1. Создание подсхемы.

Подсхема создается так же, как схема, только с внешними узлами. Для создания внешних узлов используются Порты подсхем. Ими пользователь обозначает внешние узлы подсхемы.

Для порта определяются: имя порта, номер порта в подсхеме.



#### 3.2. Создание схемного обозначения подсхемы

Чтобы на схеме подсхема отличалась от других подсхем, необходимо создать для нее собственное обозначение. Для этого перейдем в режим редактирования изображения (меню Файл/Изменить обозначение схемы).

При этом возникнет поле с простым четырехугольником и обозначения портов по числу определенных внешних узлов. Слева вместо компонентов будет возможность использовать графические примитивы. На основе этих графических примитивов пользователь может рисовать нужное изображение.

Некоторые рекомендуемые стандарты:

- цвет линий: темно-синий #000080
- толщина линий: 2

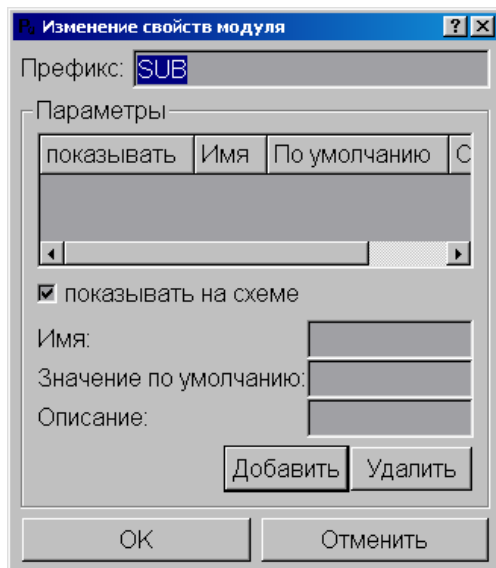
Обозначение запишется в тот же файл, что и подсхема.

Важно, что текст этого обозначения в дальнейшем можно использовать при создании моделей.

### 3.3. Назначение параметров подсхемы

У подсхемы могут быть параметры. Данные параметры могут быть использованы в блоке Data и в параметрах элементов. Например, в зависимости от параметра схемы в блоке Data могут рассчитываться переменные для элементов подсхемы.

Чтобы добавить в подсхему параметры требуется перейти в режим создания обозначения схемы и вызвать свойства подсхемы (дважды нажав на название Sub или через контекстное меню).



Для параметра можно задать имя, значение по умолчанию, описание и флаг показа на схеме. Используя кнопки Добавить/Удалить можно оперировать параметрами.

Соответственно данные параметры возникнут в дальнейшем в свойствах подсхемы при редактировании схемы. И их можно будет переназначить в основной схеме.

В блоке Data параметры можно использовать непосредственно под этими именами.

Использование подсхемы в схеме уже описано выше.

## 4. Моделирование

Запустить на расчет схему можно с помощью команды Моделировать (Меню Моделирование / Моделировать). При этом выполняются следующие задачи:

- генерируется PPL задание в файл <имя схемы>.sch.py (при этом имя фрагмента схемы будет <имя схемы>);
- запускается транслятор Python для обработки PPL файла;
- в ходе обработки генерируется PSL задание в файле <имя схемы>.sch.psl;
- PSL задание запускается на расчет.

Сообщения об ошибках в консоль (для PPL файла) и в файл SYSPRINT.TXT для PSL. Посмотреть сообщения PSL можно с помощью команды Показать лог-файл (Меню Моделирование).

Можно генерировать PPL и PSL задания без запуска на расчет: с помощью команд показать PPL файл и показать PSL файл (меню Моделирование).

Для автоматического запуска после расчета постпроцессора можно в настройках документа (Меню Файл/Настройки документа) можно установить флаг Открыть просмотр данных после моделирования (по умолчанию включен). Поскольку при тестовых запусках возможно большое количество ошибок этот флаг лучше отключать, а последние результаты расчета можно посмотреть с помощью команды Постпроцессор (меню Моделирование).

## 5. Справка

Команда Содержание справки (меню Справка) показывает справку по PRADIS Qucs.

Команда Каталог модулей (меню Справка) открывает помощь по системному каталогу по модулям элементов.

## 6. Возможные ошибки при работе

PRADIS Qucs дает широкие возможности по созданию схем для PRADIS. Однако в следствие этого в препроцессоре довольно слабая защита от ошибок пользователя. Постараемся описать наиболее вероятные ошибки начинающего пользователя и способы их диагностики.

Ошибки могут быть обнаружены на следующих этапах решения задачи:

- в препроцессоре при генерации PPL задания (проверяется соответствие типов узлов);
- при трансляции PPL файла (интерпретатором Python);
- при трансляции PSL файла (интерпретатором PRADIS).

При обнаружении ошибки препроцессор покажет сообщение об ошибке с указанием некорректных параметров.

К сожалению, в текущей версии интерпретатор Python выводит ошибку только в консоль при запуске задания. Ошибку легко увидеть, если автоматически запускается постпроцессор. Тогда консольное окно не закрывается, пока не закроется постпроцессор. Либо можно запустить PPL файл задания с консоли: "python имяфайла.py".

Ошибку PSL файла легко увидеть загрузив SYSPRINT.TXT файл с помощью команды Показать лог-файл.

Замечание. В случае если задача уже была посчитана (т.е. существуют файлы результатов) и при последующем расчете произошла ошибка, но постпроцессор был автоматически запущен, то будут показаны прошлые результаты. Будьте внимательней.

### 6.1 Ошибки, обнаруживаемые препроцессором

В текущей версии на уровне препроцессора обнаруживаются только ошибки несоответствия типов узлов.

Желательно использовать в схеме одни настройки сетки. При изменениях сетки может возникнуть ситуация, когда соединения не попадают на узлы.

### 6.2 Ошибки, обнаруживаемые транслятором Python

Все имена должны быть на латинице и начинаться с букв. В случае ошибки PPL интерпретатор выведет ошибку о неправильном имени объекта. Даже имя схемы должно быть латиницей. Не должно быть пробелов в имени, использует символ «\_». Имена не должны начинаться с «\_», так как это зарезервированные имена и могут быть пересечения.

В параметрах элементов можно использовать только переменные или константы. При этом переменные используются для создания блока \$DATA, а константы пишутся непосредственно в параметры элемента. Поэтому нельзя использовать выражения, вызовы функция и т.д. в параметрах. Используйте для этого элемент Data.

В элементе Data выражения должны быть записаны согласно правилам языка Python. При этом начальный отступ нулевой.

Важно. В языке Python отступы очень важны, они определяют блоки функций. Кроме того, отступы делаются либо пробелами, либо табуляцией и их нельзя смешивать. Подробнее смотрите документацию по Python.

Есть элементы, в которых не описываются типы узлов. Узлы таких элементов должны быть подключены к элементам с явным описанием типов узлов. Иначе интерпретатор выдаст ошибку, что переменная узла не инициализирована.

Генерируемый PPL файл можно посмотреть с помощью команды Показать PPL файл.

### **6.3 Ошибки, обнаруживаемые транслятором PSL**

В рассчитываемой схеме должен присутствовать элемент DISP, SHTERM или NEWMARK с установленными выходными переменными. Иначе PSL транслятор выдаст ошибку: не установлены выходные переменные.

При добавлении элемента PSLSubScheme не забывайте описать файл, в котором описан PSL фрагмент, с помощью элемента PSLInclude.

Генерируемый PSL файл можно посмотреть с помощью команды Показать PSL файл.

## **7. Рекомендации к использованию**

Используйте такие возможности как:

- Redo/Undo;
- вырезать/вставить/скопировать. Причем эта команда будет действовать даже при копировании в отдельно запущенную версию программы;
- проект, в случае сложных проектов с множеством подсхем;
- элементы можно деактивировать/активировать, т.е. отключать от схемы, не удаляя их;
- можно настроить сетку на удобный шаг (Файл/Настройки документа / Сетка ). Важно, что новые элементы будут ставиться по новой сетке и выводы могут не совпадать со старыми линиями. Для выравнивания элементов удобно использовать команду Выровнять по сетке;
- при просмотре текста используется внутренний текстовый редактор. Можно настроить собственный редактор в настройках приложения;
- можно выбрать собственный цвет фона в настройках приложения;
- с помощью команды Заменить (меню Правка) можно изменить значение какого-либо параметра сразу во всех элементах;
- для лучшего осмотра схемы используйте команды масштабирования вида.