

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Ковровская государственная технологическая
академия имени В.А. Дегтярева»

Эксплуатация модуля подбора пневмоцилиндров
Документация к парсеру и модулю



Дегтев И.А., 2023 г.

Оглавление

Введение	4
Глава 1 – эксплуатация.....	6
1.1 Сбор актуальных данных	6
1.2 Подбор пневмоцилиндра.....	8
1.3 Пример выполнения.....	11
Глава 2 – документация.....	13
2.1 Парсер электронных каталогов.....	13
2.2 Модуль SimInTech	18
2.3 Подпрограмма python для модуля	22
Заключение	23

Список терминов

Парсер – программа для сбора и систематизации информации, размещенной на различных сайтах.

Информационный ресурс – сайт в сети интернет, позволяющий получить доступ к программным решениям и результатам их деятельности в рамках данной работы.

Скрипт - последовательность команд, инструкций на сценарном языке, используемая для автоматизации рутинных задач.

Введение



Автоматизация проектирования и математического моделирования стала неотъемлемой частью многих областей промышленности, науки и технологических процессов. Математическое моделирование является широко распространенным методом в научных исследованиях и экспериментах, позволяющим изучать поведение системных комплексов при различных условиях в условиях идеальной ситуации. Это становится возможным благодаря имитационным моделям, которые позволяют увидеть, как система будет вести себя в реальности при определенных входных условиях.

Однако, открытым остается вопрос унификации и стандартизации изделий в рамках единых баз данных. Сам факт наличия изделий, разрозненных в разных каталогах является существенной проблемой в виду увеличения количества рутинных операций по подбору и выбору нужного изделия. Комплекс, представленный в данной работе способствует решению данной проблемы.

Доступ к ресурсам

Доступ к ресурсам предоставляется путем перехода на веб-адрес в сети интернет. В таблице 1 приведены ссылки и QR-коды для доступа к общей информационной странице и к ресурсу с исходными файлами проекта.

Таблица 1 – Ссылки на ресурсы

Описание и ссылка на ресурс	QR-код
Ресурс с общей информационной страницей о проекте https://gammakyle.github.io/other_projects/pneumo_finder/	
Ресурс с исходными файлами проекта, результатами работы, демонстрационными файлами: https://github.com/gammakyle/pneumatic_parser	

Глава 1 – эксплуатация

Данная глава предназначена в первую очередь для конструкторов, желающих изучить принцип взаимодействия с готовыми программными решениями и модулем для SimInTech, и решивших применить данные технологии в своей работе для облегчения рутинных задач.

1.1 Сбор актуальных данных

Сбор актуальных данных нужен в случае, если пользователь хочет собрать наиболее свежие данные из электронных каталогов. Сбор следует проводить раз в квартал, или после добавления новых ресурсов в общий обход разработчиками.

Для использования возможностей парсера необходимо получить его копию на общей информационной странице или на ресурсе с исходным кодом, а также выполнить следующие требования:

- Установить Python v3;
- Проверить подключение к сети интернет;
- Установить все зависимости, обозначенные в файле `Readme.md`, находящемся на информационном ресурсе.

После этого можно приступать к формированию базы данных. В каталоге «`pneumatic_parser`» будет находиться файл «`pneumatic_parser.py`». Запустите его. Будет открыта консоль с выводом адреса страницы, которую в данный момент обрабатывает программа.

Стоит отметить, что время сбора информации с одной страницы в среднем составляет 1с 340мс, поэтому требуется не закрывать программу до окончания работы. На двадцать тысяч строчек в среднем тратится около одной рабочей смены. По окончании работы программы будет сформирован готовый табличный файл «pneumatic_armature_parser_table.csv», содержащий актуальную информацию на момент парсинга. В файле будут следующие поля:

- Столбец А – производитель;
- Столбец В – стандарт изготовления;
- Столбец С – тип хода;
- Столбец D – тип поршня;
- Столбец Е – диаметр поршня;
- Столбец F – ход штока;
- Столбец G – рабочее давление;
- Столбец H – минимальная температура;
- Столбец I – максимальная температура;
- Столбец J – тип подключения;
- Столбец K – ссылка на источник.

Полученные данные можно импортировать в excel для более удобной работы, если по умолчанию строчки не переводятся в отдельные ячейки из-за локальных настроек операционной системы. Для этого создайте чистую книгу, после чего переключитесь на вкладку «Данные». Выберите пункт «Из текста», после чего выберите файл с таблицей, а в качестве разделителя укажите «Запятая» (рис. 1).

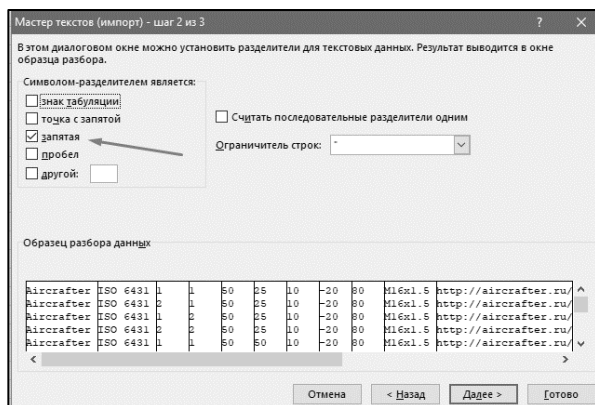


Рис. 1 – формирование таблицы

1.2 Подбор пневмоцилиндра

Для подбора пневмоцилиндра в математической модели SimInTech необходимо добавить в Ваш проект файлы из каталога pneuma_finder. Если Вы не проводили сбор актуальных данных из пункта 1.1, то также следует добавить в проект таблицу из каталога parser_table_results.

Скопируйте модуль из файла pneumo_finder.prt в проект SimInTech. Модуль принимает на вход четыре значения, представленные на рисунке 2.

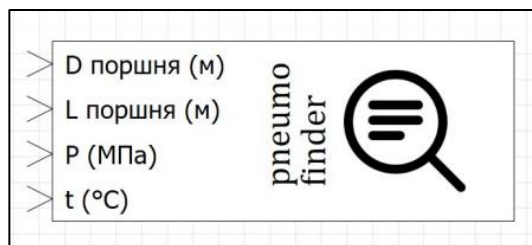


Рис. 2 – модуль подбора изделий

Подключите все входные параметры в позиции модуля. При двойном щелчке на модуль откроется внутренние составляющие, а именно блоки с переводными коэффициентами, блок алгоритма и типы штока/действия пневмоцилиндра. Вы можете изменить коэффициенты нижних блоков, отвечающих за тип штока/действия, чтобы актуализировать выборку необходимых изделий под Ваш проект.

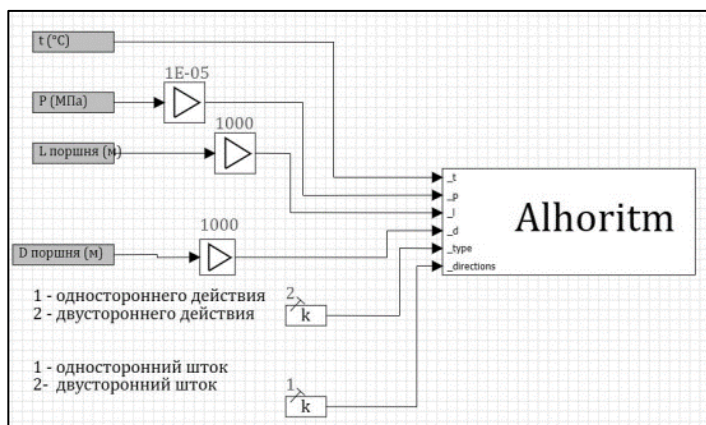


Рис. 3 – внутреннее устройство модуля

При проектировании стоит учесть тот фактор, что в модуле установлены переводные коэффициенты. При необходимости Вы можете их удалить. На вход в модуль поступают следующие величины:

- Диаметр поршня – D , м;
- Длина штока – L , м;
- Давление – P , МПа;
- Температура – t , °C.

Готовый образец проекта с внедренным модулем изображен на рисунке 4.

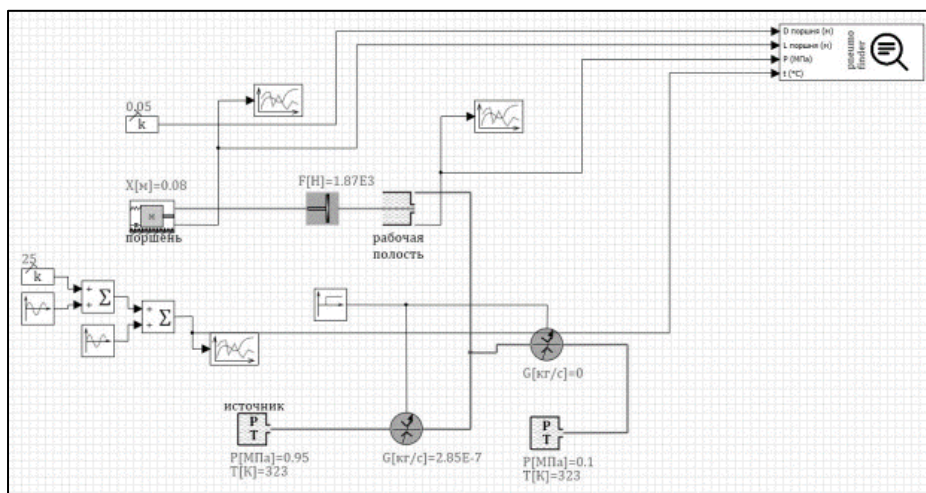
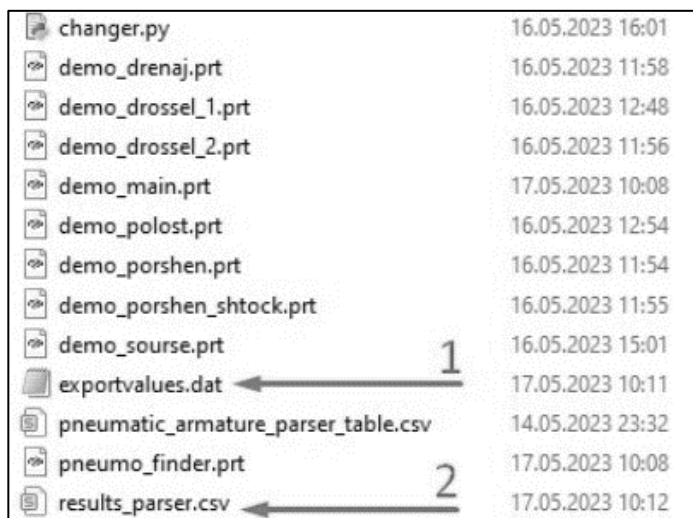


Рис. 4 – образец проекта с внедренным модулем.

Производится расчет математической модели. По окончании расчетов в модуле накапливаются параметры, например, давление. Оно может меняться на протяжении всей симуляции, однако модуль будет накапливать и выдавать максимальное, при которой система может работать. Как только моделирование будет остановлено пользователем, сработает скрипт finalization, который запустит процедуры поиска и подбора. Начало и конец работы внешнего скрипта обозначаются всплывающими диалоговыми сообщениями. После окончания работы

внешнего модуля подбора в папке с проектом появятся два новых файла - exportvalues.dat и results_parser.csv. В первом заключены значения, которые были получены после математического моделирования. Во втором файле находятся все подходящие пневмоцилиндры (рис. 5).



changer.py	16.05.2023 16:01
demo_drenaj.prt	16.05.2023 11:58
demo_drossel_1.prt	16.05.2023 12:48
demo_drossel_2.prt	16.05.2023 11:56
demo_main.prt	17.05.2023 10:08
demo_polost.prt	16.05.2023 12:54
demo_porshen.prt	16.05.2023 11:54
demo_porshen_shtock.prt	16.05.2023 11:55
demo_sourse.prt	16.05.2023 15:01
exportvalues.dat	17.05.2023 10:11
pneumatic_armature_parser_table.csv	14.05.2023 23:32
pneumo_finder.prt	17.05.2023 10:08
results_parser.csv	17.05.2023 10:12

Рис. 5 – результат работы программы.

Данные о подходящих изделиях содержатся в форме таблицы и могут быть отсортированы дополнительно.

1.3 Пример выполнения

Для наглядного примера в проекте присутствует папка «pneuma_finder_demo». В ней находятся необходимые для демонстрации файлы. Допустим, есть задача подобрать пневмоцилиндр для некоторого агрегата, способного работать в центральной европейской части России. Температура окружающей среды - от -20 до 80 градусов (с учетом температуры, выделяемой агрегатом и самонагревом). Пневмоцилиндр

должен обеспечивать передвижение стальной заглушки. Давление, подаваемое источником питания - 9.5 бар, необходимое перемещение - 150миллиметров. Диаметр поршня должен составлять 50 миллиметров. Собираем модель и устанавливаем параметры, как в условии. После запуска наблюдаем результат (рис. 6)

A1		f _x		E.MC											
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
1	E.MC	ISO 15552	2	1	50	150	10	-20	80	M16x1.5	https://industriation.ru/fvbc50150r/				
2	Pemaks	ISO 15552	2	1	50	150	10	-20	80	M16x1.5	https://industriation.ru/dmcea-50-150/				
3	Pneumax	ISO 15552	2	1	50	150	10	-30	80	M16x1.5	https://industriation.ru/1390-50-0150-01-p/				
4	Pneumax	ISO 15552	2	1	50	150	10	-30	80	M16x1.5	https://industriation.ru/1391-50-0150-01-p/				
5	Festo	ISO 15552	2	1	50	150	12	-20	80	M16x1.5	https://industriation.ru/2098974/				
6	Festo	ISO 15552	2	1	50	150	12	-20	80	M16x1.5	https://industriation.ru/1922633/				
7	Festo	ISO 15552	2	1	50	150	12	-20	80	M16x1.5	https://industriation.ru/3659477/				
8	E.MC	ISO 15552	2	1	50	150	10	-20	80	M16x1.5	https://industriation.ru/tbc50150r/				
9															

Рис. 6 – выборка актуальных решений под заданные параметры.

Таким образом, из двадцати двух тысяч различных изделий было выбрано восемь, подходящих под условия, собранные в результате выполнения математической модели.

Глава 2 – документация

Данная глава предназначена в первую очередь для разработчиков, желающих изучить принцип работы парсера и модуля, а также обладающих возможностью модифицировать проект под собственные нужды. Все файлы с подробными комментариями содержатся на ресурсе с исходными файлами.

2.1 Парсер электронных каталогов

Основной принцип работы парсера – загрузка веб-страниц на сторону клиента, их разбор, правка и выделение необходимых компонентов из общего кода страницы.

Всего в проекте находится восемь файлов, основной из которых – «pneumatic_parser.py». Он ссылается на остальные файлы python при выполнении функционала программы. Полный перечень файлов, входящих в комплект, распространяемый через GitHub, можно изучить в таблице 2.

Таблица 2 – Исходные файлы парсера

№	Наименование файла	Описание
1	aircrafter_parser.py	Модуль, отвечающий за разбор страницы ресурса aircrafter
2	dalse_parser.py	Модуль, отвечающий за разбор страницы ресурса dalse

Продолжение таблицы 2

3	exporter.py	Модуль, отвечающий за сохранение собранных данных в таблицу
4	data_pages.txt	Исходник с адресами ресурсов, необходимых к обходу
5	pneumatic_parser.py	Основной файл программы
6	substring_search.py	Модуль, отвечающий за определение текущего источника информации
7	industrialization_parser.py	Модуль, отвечающий за разбор страницы ресурса industrialization
8	test.py	Модуль для тестовых реализаций механизмов обработки

Следует понимать, что под разные электронные каталоги необходимо составлять отличающиеся решения, так как сами электронные каталоги отличаются друг от друга. В дальнейшем будет рассмотрено несколько наиболее распространенных вариантов, которые можно обработать наиболее похожими на представленные способами.

Для возможности модификации парсера необходимо получить его копию на общей информационной странице или на ресурсе с исходным кодом, а также выполнить следующие требования:

- Установить Python v3;
- Проверить подключение к сети интернет;
- Установить все зависимости, обозначенные в файле Readme.md, находящемся на информационном ресурсе.

Общий механизм работы «pneumatic_parser.py» можно изучить на рисунке 7.

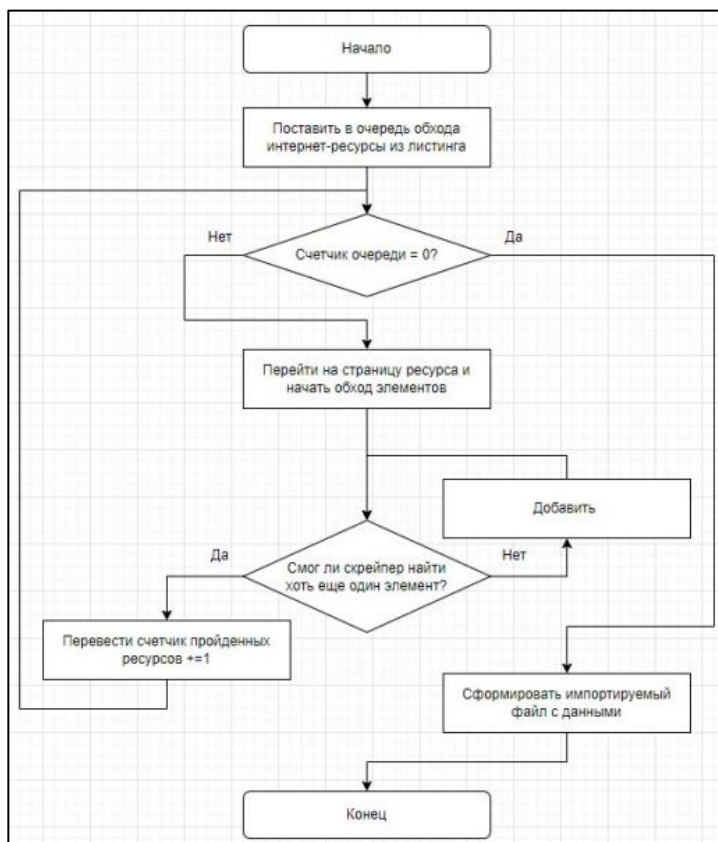


Рис. 7 – общий алгоритм работы парсера.

Механизм обхода страниц работает следующим образом:

```
_datafile = open("data_pages.txt", "r")
while True:
    _url_line = _datafile.readline()
    _url_line = _url_line.strip()
    if not _url_line:
```

```
        break
    _url_type = search_substring(_url_line)
    if _url_type == "your_resource_name":
        parse_your_resource_name (_url_line, _HEADERS)
_datafile.close
```

Где «your_resource_name» - название ресурса. В таком случае в файле «datapages.txt» указывается перечень страниц для обхода, которые могут являться как страницами с изделиями в случае, если на одной странице находится множество изделий, или же на каталог с каточками изделий, если ресурс подразумевает наличие одного продукта на одной странице.

В каждом обработчике страниц без исключения включен код, отсылающий к функции сохранения в общую таблицу:

```
List_of_data = [_name_object, _standard, _type_cylinder, _type_movie,
                _piston_diameter, _stroke_length, _operating_pressure, _left_value,
                _right_value, _stem_thread, _url_link_down]
to_csv(List_of_data)
```

Сам код, отвечающий за сохранение данных в таблицу достаточно прост:

```
def to_csv(List_of_data):
    with open('pneumatic_armature_parser_table.csv', 'a',
encoding="utf-8") as f_object:
        writer_object = writer(f_object, delimiter=";",
lineterminator='\n')
        writer_object.writerow(List_of_data)
    f_object.close()
```


Каждый из обработчиков страниц содержит часть, отвечающую за выгрузку страницы и генерацию кода для дальнейшей обработки, а также набор базовых переменных для дальнейшей передачи на сохранение. К конкретным элементам на странице парсер обращается через полный путь XPath.

```
_url_now = _url.replace('#',str(_num_page))
_webpage = requests.get(_url_now, headers=_HEADERS)
_soup = BeautifulSoup(_webpage.content, "html.parser")
_dom = etree.HTML(str(_soup))
_webpage_down = requests.get(_url_link_down, headers=_HEADERS)
_soup_down = BeautifulSoup(_webpage_down.content, "html.parser")
_dom_down = etree.HTML(str(_soup_down))
_name_object =
fix_encoding(_dom_down.xpath('/html/body/div[3]/div[2]/itemproduct/div
[2]/div[1]/div[1]/div[3]/div/div[2]/div[3]/div[2]/div/a')[0].text).rep
lace("\t", "")
_standard = 1
_piston_diameter = 1
_type_cylinder = 1
_type_movie = 1
_stroke_length = 1
_operating_pressure = 1
_operating_temperature = 1
_left_value, _right_value = None, None
_stem_thread = 1
```

После этого этапа начинаются отличия. Яркий пример для сравнения - industriation.ru и aircrafter.ru. На первом ресурсе каждый отдельный пневмоцилиндр обладает собственной уникальной страницей. На втором ресурсе представлены страницы для целой серии изделий, и, соответственно, если в первом случае достаточно просто найти и очистить данные, то во втором необходим обход страницы с целью поиска и

формирования списка всех возможных изделий из различных конфигураций.

Для случаев с каталогом уникальных страниц добавляется дополнительный цикл с обходом карточек. Следует учитывать, что среди карточек могут находиться рекламные предложения, которые нужно исключать.

```
for _num_card in range (1,49):  
    if(_num_card != 7 and _num_card !=42):  
        try:  
            _link_down =  
( '/html/body/div[3]/div[2]/div[3]/div[2]/div/div[2]/div[4]/div[1]/div[  
%d]/div/div[1]/a/@href' %_num_card)  
            _url_link_down =  
fix_encoding(_dom.xpath(_link_down)[0])  
        except:  
            pass
```

2.2 Модуль SimInTech

Модуль представляет из себя субмодель с четырьмя линиями входа, каждая из которых получает на вход математическое значение (рис. 8).

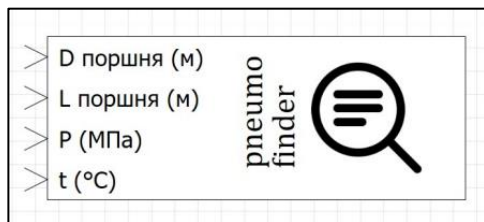


Рис. 8 – модуль подбора изделий

Чтобы отредактировать порты внешнего модуля, следует выполнить следующие действия: ПКМ – свойства объекта – порты (рис. 9).

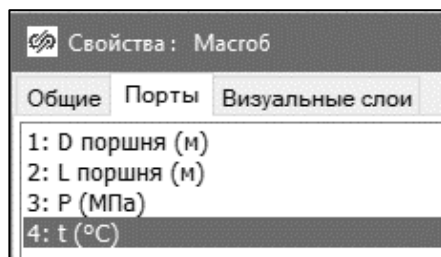


Рис. 9 – изменение внешних портов

Для нескольких из входных значений внутри субмодели используются переводные коэффициенты. Так, например, идет перевод МПа в бар, а также метров в миллиметры. Помимо входных значений существует несколько внутренних констант. Одна из них отвечает за тип действия. По умолчанию стоит двойка, т.е. пневмоцилиндр двустороннего действия. Вторая - за тип штока. По умолчанию стоит односторонний. Данные константы можно сменить вручную, чтобы изменить настройки выборки.

Внутри блока существуют следующие переменные:

- `_type` - тип действия;
- `_directions` - тип штока;
- `_d` - диаметр поршня;
- `_l` - длина штока;
- `_pmax` - максимальное давление;
- `_tmin` - минимальная температура;
- `_tmax` - максимальная температура;

Чтобы отредактировать порты внутреннего модуля, следует выполнить следующие действия: ПКМ – свойства объекта - порты (рис. 10).

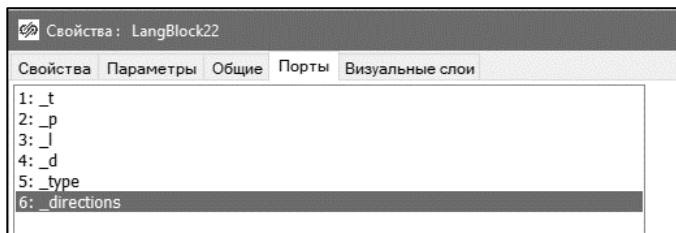


Рис. 10 – изменение внутренних портов

Для нескольких из входных значений внутри субмодели используются переводные коэффициенты. Так, например, идет перевод МПа в бар, а также метров

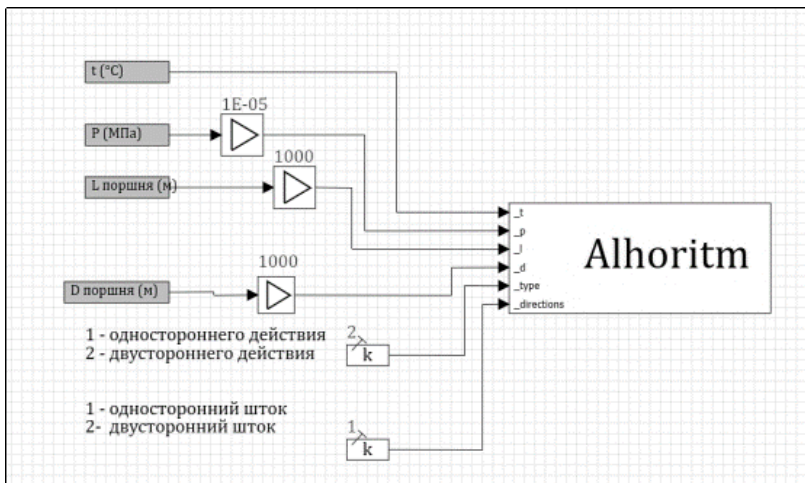


Рис. 11– внутренняя часть модуля

Начало скрипта – объявление всех поступающих значений и их типов. Такая конструкция является точкой входа, и позволяет принимать значения каждый шаг исполнения математической модели:

```
input
_t:double, _p:double, _l:double, _d:double, _type:double, _directions:doubl
e;
```

На протяжении всего времени моделирования работает обработка входящих значений. Так, удастся найти максимальную температуру, давление, минимальную температуру.

```
if time=0 then _tmax=_t else
if goodstep then _tmax=max(_tmax, _t);

if time=0 then _tmin=_t else
if goodstep then _tmin=min(_tmin, _t);

if time=0 then _pmax=_p else
if goodstep then _pmax=max(_pmax, _p);
```

По окончании моделирования срабатывает скрипт finalization, который в свою очередь выполняет все процедуры такие как подготовка файла вывода, вывод значений, запуск подпрограммы changer.py.

```
finalization
deletefile("exportvalues.dat")
f_id = createfile("exportvalues.dat", -1)
writelnutf8(f_id, _type + " " + _directions + " " + _d + " " + _l + "
" + _pmax + " " + _tmin + " " + _tmax);
freeobject(f_id);
shellexec("open", "changer.py", "", "");
end;
```

2.3 Подпрограмма python для модуля

Общий смысл работы таков: программа, содержащаяся в файле `changer.py` открывает документ `exportvalues.dat` и получает оттуда значения, рассчитанные в ходе математического моделирования. После этого находятся все значения, которые удовлетворяют условиям. По мере прохождения всего тела таблицы, программа записывает обнаруженные совпадения в файл `results_parser.csv`.

```
with open("results_parser.csv", "w") as _file:
    for line in lines:
        data = line.strip().split(",")
        try:
            if len(data) >= 9:
                c = float(data[2])
                d = float(data[3])
                e = float(data[4])
                f = float(data[5])
                g = float(data[6])
                h = float(data[7])
                i = float(data[8])
                if c == _val3 and d == _val4 and e == _val5 and f ==
_val6 and g >= _val7 and h <= _val8 and i >= _val9:
                    _file.write(line)
```

После окончания работы подпрограммы будет выдано окно о завершении работы. В папке проекта будет сформирован файл `results_parser.csv`, содержащий результаты работы программы.

Заключение

На основе данного руководства были разобраны варианты взаимодействия с частями комплекса как в конструкторской деятельности, так и в деятельности разработчика. Автор работы считает, что дальнейшее развитие подобных механизмов для различных изделий как из сферы пневматических устройств, так и в целом для большинства изготавливаемых изделий сможет решить вопрос унификации и каталогизации, что позволит более рационально подходить к вопросу выбора изделий.