



Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ \_\_\_\_\_ Робототехники и комплексной автоматизации (РК)  
КАФЕДРА \_\_\_\_\_ Систем автоматизированного проектирования (РК6)

---

***РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К КУРСОВОЙ РАБОТЕ ПО ДИСЦИПЛИНЕ***

***“Технологии интернет”***

***НА ТЕМУ:***

***Разработка динамических пользовательских интерфейсов  
для распределенных систем инженерного анализа***

Студент \_\_\_\_\_  
(Группа) **РК6-81**

\_\_\_\_\_  
(Подпись, дата)

**Неклюдов С.А.**  
(И.О.Фамилия)

Руководитель курсовой работы (проекта)

\_\_\_\_\_  
(Подпись, дата)

**Соколов А.П.**  
(И.О.Фамилия)

Консультант

\_\_\_\_\_  
(Подпись, дата)

**Першин А.Ю.**  
(И.О.Фамилия)

2019 г.

**Министерство образования и науки Российской Федерации**  
**Федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**«Московский государственный технический университет имени Н.Э. Баумана**  
**(национальный исследовательский университет)»**  
**(МГТУ им. Н.Э. Баумана)**

---

УТВЕРЖДАЮ

Заведующий кафедрой РК6-81  
(Индекс)

\_\_\_\_\_ А.П.Карпенко  
(И.О.Фамилия)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

**З А Д А Н И Е**  
**на выполнение курсовой работы**

по дисциплине Технологии интернет

Студент группы РК6-81 \_\_\_\_\_ Неклюдов Семен Александрович  
(Фамилия, имя, отчество)

Тема курсовой работы Разработка динамических пользовательских интерфейсов для  
распределенных систем инженерного анализа

Направленность КР: учебная

Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения КР: 25% к \_\_\_\_ нед., 50% к \_\_\_\_ нед., 75% к \_\_\_\_ нед., 100% к \_\_\_\_ нед.

**Техническое задание** 1) доработать скрипт автоматизации развертывания web-клиента docker 2) доработать интерпретатор aIni 3) разработать шаблон для отображения сгенерированных на основе файла aIni форм

**Оформление курсовой работы:**

Расчетно-пояснительная записка на \_\_\_\_\_ листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Дата выдачи задания « \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

**Руководитель курсовой работы**

\_\_\_\_\_  
(Подпись, дата)

А.П.Соколов

(И.О.Фамилия)

**Студент**

\_\_\_\_\_  
(Подпись, дата)

С.А. Неклюдов

(И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

## **АННОТАЦИЯ**

Работа посвящена разработке динамического графического пользовательского интерфейса подсистемы ввода – вывода распределенной системы инженерного анализа GCD а также автоматизации развертывания приложения WEB клиента PBC GCD. В первой части работы производились доработки подсистемы ввода-вывода PBC GCD, в частности был разработан шаблон для генерации форм по входным параметрам. Во второй части работы было выполнено ознакомление со средой развертывания и управления приложениями в средах с поддержкой контейнеризации(виртуализация на уровне операционной системы). Был доработан скрипт развертывания приложения Web – клиента для установки локали и поддержки кодировки UTF на сервере.

## **СОКРАЩЕНИЯ**

РВС – Распределенная вычислительная система.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
1. ПОСТАНОВКА ЗАДАЧИ .....	8
1.1. Концептуальная постановка задачи.....	8
2. АРХИТЕКТУРА ПРОГРАММНОЙ РЕАЛИЗАЦИИ .....	9
3. ТЕСТИРОВАНИЕ И ОТЛАДКА .....	16
4. ЗАКЛЮЧЕНИЕ.....	21
СПИСОК ЛИТЕРАТУРЫ.....	<a href="#"><u>22</u></a>

## ВВЕДЕНИЕ

Изучением методов построения динамических пользовательских интерфейсов занимались многие исследователи и разработчики программного обеспечения.

По оценкам специалистов на разработку пользовательского интерфейса затрачивается не менее половины времени, требуемого на разработку программного средства [4]. Для исследователей и разработчиков методы построения GUI имеют важное практическое значение: они позволяют снизить время разработки программного обеспечения. В системах инженерного анализа не использующих методы динамического построения графического интерфейса пользователя время, затраченное на разработку модулей подготовки данных значительно выше, чем в системах, использующих такие методы, что наглядно показано в статье [3]. Были выявлены основные тенденции развития пользовательских интерфейсов такие как: событийная архитектура[11] , использование MVC фреймворков и JavaScript технологии. На основе проведенного анализа литературы были изучены несколько подходов к автоматическому построению динамических пользовательских интерфейсов [6-10], в том числе подход, используемый в распределенной системе инженерного анализа GCD[3], изучены требования, предъявляемые к программному инструментарию PBC GCD:

1) программный инструментарий должен включать в свой состав GUI генератор, обеспечивающий автоматическое построение GUI на основе заранее определенных входных параметров;

2) для составления списка входных параметров должен быть разработан специализированный текстовый формат данных;

3) для составления списка входных параметров должен быть разработан специализированный текстовый формат данных;

4) формат файла должен поддерживать хранение скалярных типов данных;

5) должна иметься возможность расширения списка поддерживаемых типов;

6) программа-генератор GUI должна зависеть не от конкретных метаданных, а лишь от грамматики специализированного текстового формата, определяющей произвольные метаданные;

7) подготовка списка входных параметров должна быть доступной для неподготовленного специалиста, не владеющего навыками программирования, в том числе для специалистов, представляющих заказчика разрабатываемой прикладной программы;

8) подготовка списка входных параметров должна быть доступной для неподготовленного специалиста, не владеющего навыками программирования,

в том числе для специалистов, представляющих заказчика разрабатываемой прикладной программы;

9) при изменении исходного списка входных параметров соответствующий GUI должен быть перестроен автоматически;

10) процесс построения GUI должен осуществляться во время выполнения разрабатываемой прикладной программы, использующей программный инструментарий;

11) программный инструментарий должен обеспечивать возможность автоматического определения параметров различных типов;

12) специализированный текстовый формат данных, обеспечивающий возможность определения метаданных, должен предоставлять возможность группировки входных параметров по их назначению;

13) должна быть обеспечена возможность редактирования списка входных параметров независимо от программы-генератора GUI;

14) программный инструментарий должен иметь возможность встраивания в web-ориентированные приложения, приложения для мобильных платформ и приложения для операционных систем семейств Windows, Linux, macOS. Для поддержки соответствующих платформ должен быть разработан свой GUI-генератор; Основной задачей проведения аналитического обзора литературы было ознакомление с методами формирования графических пользовательских интерфейсов, в частности, подхода, используемого в PBC GCD.

Контейнеризация – метод виртуализации (предоставление набора вычислительных ресурсов абстрагированных от аппаратной архитектуры) при котором ядро операционной системы поддерживает несколько изолированных экземпляров адресного пространства виртуальной памяти операционной системы вместо одного. В статье [12] автор говорит о достоинствах технологии контейнеризации, таких как:

- 1) Запуск приложений в изолированных средах с изолированными файловыми системами.
- 2) Контейнеры имеют преимущество перед классической виртуализацией в виде небольшого расхода оперативной памяти и процессорного времени.
- 3) Контейнеры упрощают развертывание и перенос приложений.

# **1. ПОСТАНОВКА ЗАДАЧИ**

## **1.1. Концептуальная постановка задачи**

**Объект исследования:** методы автоматического формирования GUI

**Целью разработки является:** доработка подсистемы ввода — вывода PBC GCD:

- 1) доработка WEB - ориентированных GUI, основанных на формате Aini в части поддержки специальных типов параметров и отображения различных форм для ввода этих параметров;
- 2) доработка скрипта автоматизации развертывания приложения на сервере с целью поддержки кодировки UTF-8 в приложении веб - клиента.



## 2. АРХИТЕКТУРА ПРОГРАММНОЙ РЕАЛИЗАЦИИ

### 2.1 Доработка web – клиента.

Используемые технологии:

1. Клиентская часть: HTML5, CSS, JavaScript, JQuery
2. Серверная часть: Python3, Django, Docker

Разработка ведется на базе проекта comwps системы PBC GCD, а точнее, в рамках модулей wpc\_gui\_ini\_builder и iniparser, представляющих собой обработчик аjax-запросов и парсер файла входных параметров в формате aINI, а также была выполнена доработка Django шаблонов для отображения извлеченной информации.

Фреймворк Django навязывает некоторую структуру для архитектуры приложения. Выделяют 3 главные составляющие приложения, написанного при помощи Django: представления - интерпретируют действия пользователя; модели — сущности, ассоциирующиеся с данными, реагирующие на команды представлений; шаблоны отвечают за отображения данных для пользователя.

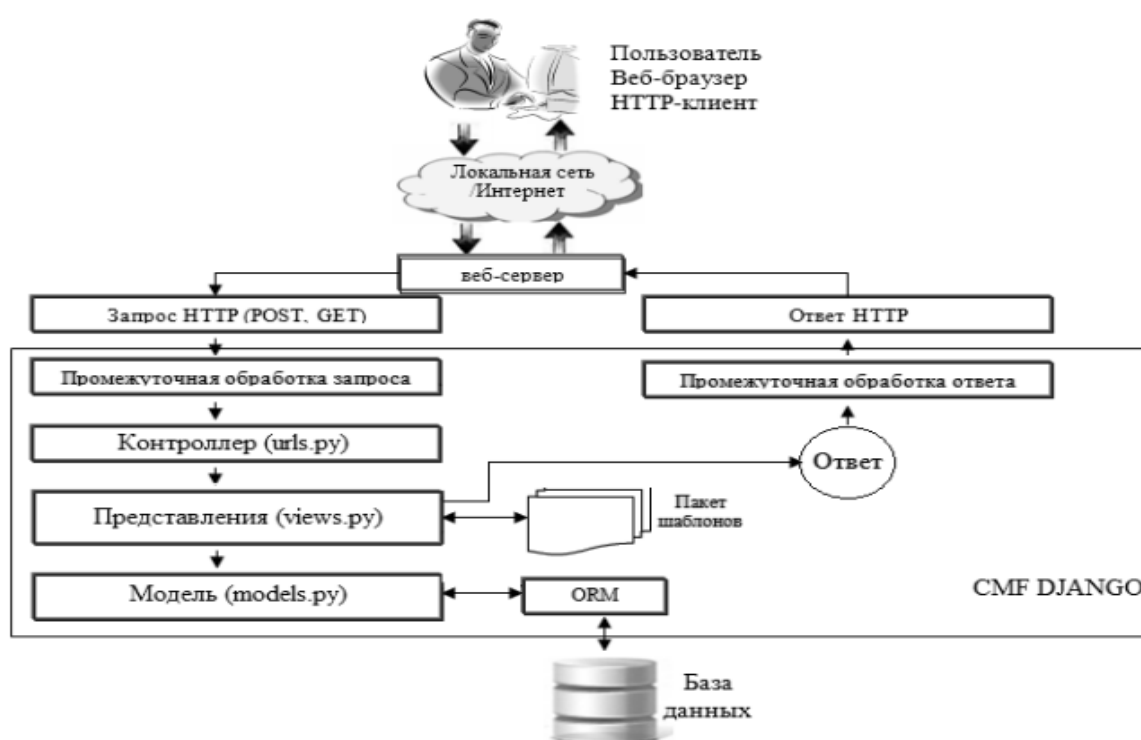


Рис. 1 . Структура приложения Django

Для понимания роли разработки в системе следует представить общую архитектуру подсистемы web — клиента comwps и последовательность выполнения аjax запроса на выполнение action item wpc\_gui\_ini\_builder:

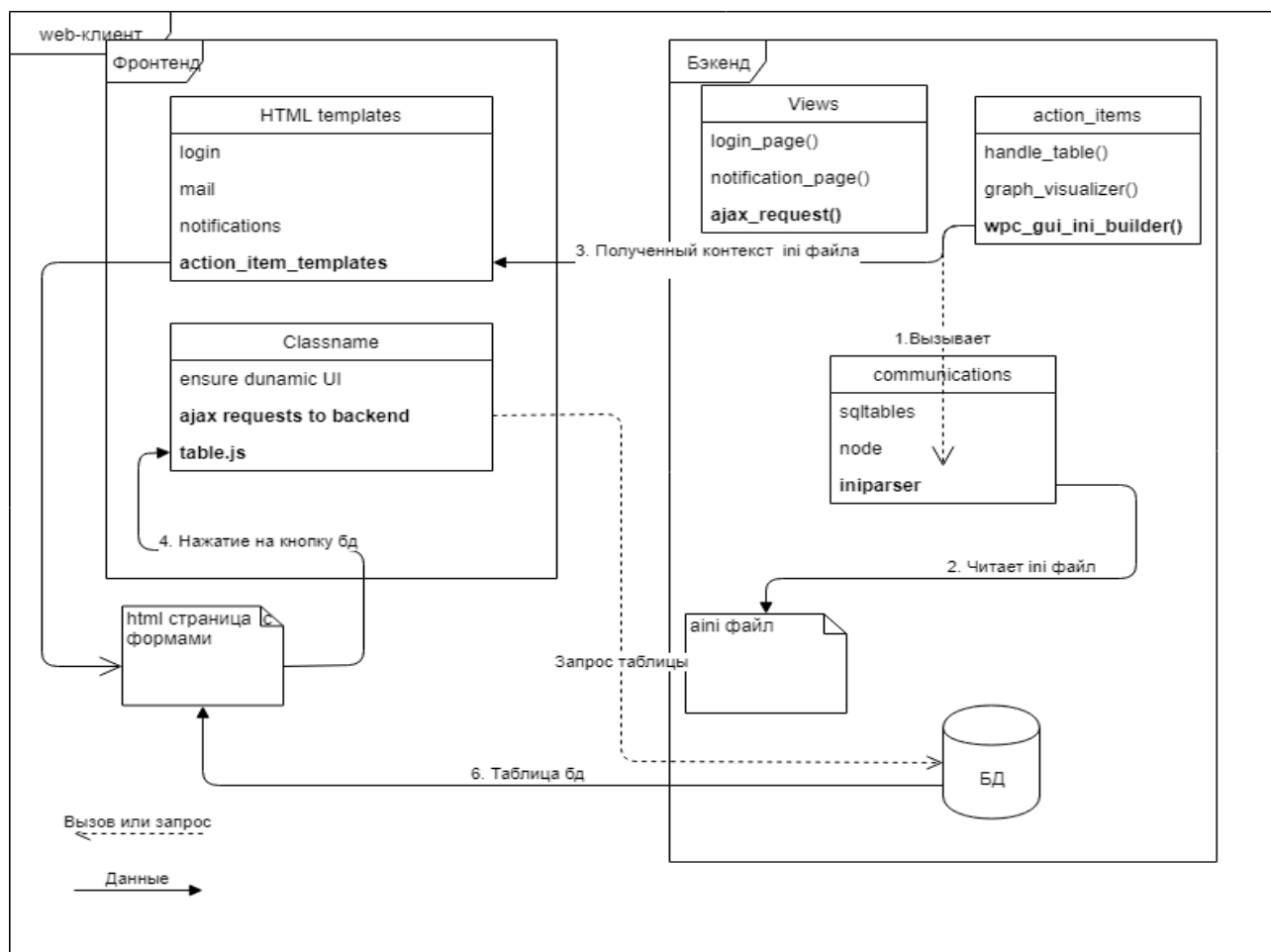


Рис. 2 . Архитектура web — клиента

На серверной части система состоит из нескольких представлений, реализующих основной функционал web — клиента (авторизация, отправка запросов), `action_item` (действия), а также парсер, классы для взаимодействия с сетью и базой данных.

Клиентская часть состоит из DJANGO шаблонов и сценариев на языке JavaScript.

Процедура генерации gui включает в себя следующие стадии:

- 1) пользователь, находясь на главной странице нажимает на поле меню, после чего отправляется ajax — запрос;

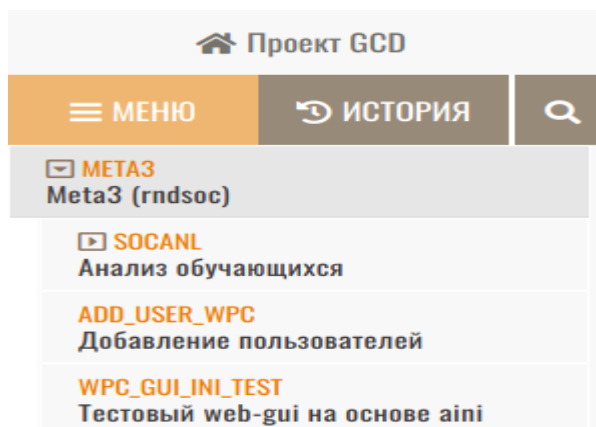


Рис. 3 Меню. Каждый элемент соответствует action item

- 2) поступивший запрос обрабатывается представлением `ajax_request`, где на основе данных из поступившего запроса запускается определенный обработчик, а также обрабатываются исключительные ситуации;
- 3) если тип соответствует `wps_gui_ini_test`, то запускается построитель GUI. В одной директории с обработчиком располагается файл формата aINI , представляющий собой несколько различных секций с определениями переменных. Переменные могут быть различных типов, могут быть обязательны для заполнения(атрибут «\*» перед объявлением), а также объявляться, но не отображаться( атрибут «-») . Обработчик представляет собой функцию, способную возвращать HTML код ini файла(рендеринг) и отправлять на обработку полученные данные по нажатию кнопки «Обработать»;

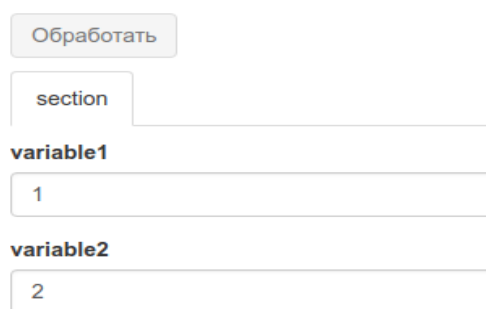


Рис. 4 Пример сгенерированного GUI

- 4) в обработчике происходит разбор файла входных параметров, расположенного в соответствующей директории при помощи `iniparser` и формируется контекст для рендеринга по HTML шаблону. Парсинг осуществляется построчно, в контекст записывается информация о секциях, атрибутах, переменных, комментариях. На основе регулярных выражений определяется тип значения. Поддерживаются несколько типов данных: целые, вещественные числа, диапазоны, массивы, пары `x-y` для представления функций одной переменной, ссылки на записи таблиц баз данных и др. Разбор этих строк основан на механизме регулярных выражений и реализован с помощью стандартного модуля `re` языка Python. Итоговый контекст представляет собой список, содержащий необходимые для генерации формы параметры, такие как тип переменной, информация о секции и атрибутах. Этот список передается в шаблонизатор, где каждому типу данных ставится в соответствие некоторая часть шаблона. Так, на основе переменной логического параметра генерируется чекбокс, а на основе переменной типа целого числа — обычное поле ввода со значением по умолчанию.

Для каждого типа указывается соответствующий placeholder — строка подсказка;

<pre>ptString = 0 ptArray = 1 ptTwoDimList = 2 ptSet = 3 ptCheck = 4 ptFile = 5 ptTree = 6 ptRange = 7 ptUnitValue = 8 ptDBTableValue = 9 ptSubParam = 10 ptNum = 11 ptFloatNum = 12 ptFX = 13 ptUnknown = -1</pre>	<pre>&lt; Строка &lt; Массив: (abc;def;ghi)   &lt; Двумерный массив: ((abc;def;ghi) ; (abc;def;ghi)) &lt; Множество с заданным значением: [abc]{abc def ghi}   &lt; Флаг [0]{0 1}   &lt; Файл: [file.ext]   &lt; Дерево, комплексный параметр     &lt; Диапазон: [5; 0:10; 1], где 5 - заданное значение, 0 - минимальное значение, 10 - максимальное значение,       &lt; Значение с размерностью: 20[[GPa]]       &lt; Значение из таблицы INIили базы данных: [somepk]\$sys.table       &lt; Субпараметр, используемых для построения параметра типа ptTree     &lt; Целое число       &lt; Число с плавающей точкой       &lt; Функция одного переменного     &lt; Неизвестный тип данных</pre>
---	---

## 5. Поддерживаемые параметры

Пример сформированного интерпретатором контекста для файла входных данных:

Листинг 1. Пример файла aIni

```
[section]

variable1=1 // Параметр X

variable2=2 // Параметр Y
```

Листинг 2. Контекст

```
{

    'action_item': 'wpc_gui_ini_builder',

    'action_class': 'PYWEBS',

    'aini_name': 'WPC_GUI_INI_TEST',

    'processing': 'PYTHON_CALLER',

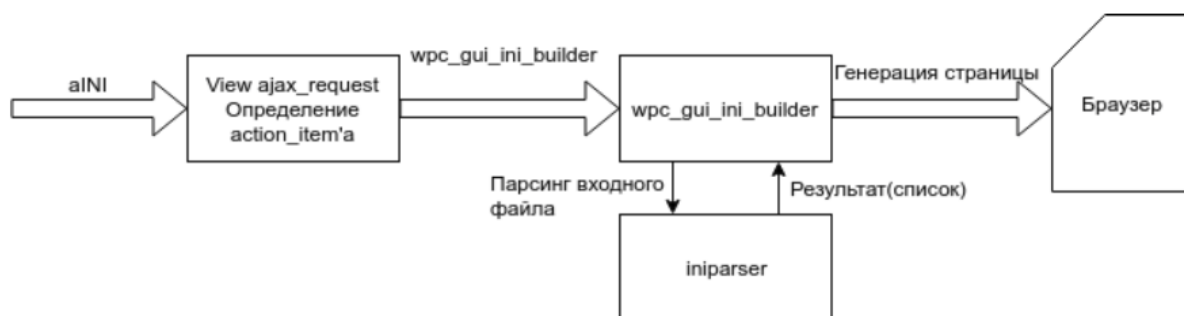
    'proc_ai_class': 'PLUGIN',

    'all_data':

        OrderedDict([('section',
            OrderedDict([('variable1', {'value': '1', 'type':
                <enu_INIParamType.ptNum: 11>)), ('variable2',
                {'value': '2', 'type': <enu_INIParamType.ptNum:
                11>})])])])])})
```

В процессе парсинга были выделены значения 2 переменных, а также секция и комментарии.

5) на основе содержимого контекста происходит генерация HTML страницы



в соответствии с HTML шаблоном;

Рис. 6. Схема обработки запроса

## 2.2 Доработка скрипта автоматизации развертывания приложения Docker:

### Листинг 4. Docker скрипт

```
FROM python:latest
USER root
RUN apt-get update
RUN apt-get install -y apache2 apache2-dev libapache2-mod-wsgi-py3
RUN apt-get update
RUN apt-get install -y locales locales-all
WORKDIR /opt
ADD . /opt/comwpc
WORKDIR /opt/comwpc
RUN locale-gen "ru_RU.UTF-8"
RUN update-locale LC_ALL="ru_RU.UTF-8"
RUN export LC_ALL=ru_RU.UTF-8
RUN pip3 install mod_wsgi
RUN pip3 install -r requirements.txt
RUN a2enmod ssl
RUN echo "\nServerName localhost\n" >> /etc/apache2/apache2.conf
#RUN echo "127.0.0.1 localhost\n 127.0.1.1 sa2systems.ru" > /etc/hostname
RUN echo "Listen 8080\n\
\n\
<VirtualHost *:8080>\n\
    Alias /static /opt/comwpc/dev/static/static_root\n\
    ServerAdmin asapelkin@sa2systems.ru\n\
    ServerName sa2systems.ru\n\
\n\
    AddDefaultCharset UTF-8\n\
    WSGIDaemonProcess sa2systems.webclient lang='ru_RU.UTF-8' locale='ru_RU.UTF-8' python-
path=/usr/local/lib/python3.7/site-packages/:/opt/comwpc/dev/\n\
WSGIProcessGroup sa2systems.webclient\n\
\n\
    WSGIScriptAlias / /opt/comwpc/dev/server/wsgi.py process-group=sa2systems.webclient\n\
\n\
    <Directory /opt/comwpc/dev/>\n\
        Require all granted\n\
    <Files wsgi.py>\n\
        Require all granted\n\
    </Files> \n\
    </Directory>\n\
\n\
    CustomLog /var/log/webclient common\n\
    ErrorLog /var/log/webclient_error\n\
\n\
</VirtualHost>" > /etc/apache2/sites-available/django.conf
RUN a2ensite django
WORKDIR /opt/comwpc/dev
RUN echo "yes" | python3 manage.py collectstatic --clear
RUN python3 manage.py makemigrations
RUN python3 manage.py migrate
RUN chmod -R 755 /opt/comwpc
RUN chown -R www-data.www-data /opt/comwpc
WORKDIR /opt/comwpc
EXPOSE 8008
CMD /usr/sbin/apache2ctl -D FOREGROUND
```

Данный сценарий конфигурирует контейнер таким образом, чтобы в нем был возможен запуск приложения веб – клиента. Аналогией может послужить процесс установки разработчиком или администратором утилит на новую систему. В данном сценарии на образ диска устанавливается веб – сервер Apache с различными модулями, устанавливается стандартный пакет локалей,

заполняется конфигурационный файл веб – сервера и запускается web – клиент PBC GCD.

### 3. ТЕСТИРОВАНИЕ И ОТЛАДКА

3.1 Для тестирования был подготовлен aINI файл входных параметров:

```
1 [Author]//Идентификация автора разработки
2 AuthorName=[alsokolo]$sys.user
3 tbl=[dscra]$GCDDDB.com.slvr
4
5 *AuthorSid=sa//SID автора
6 -OutputFile=@AuthorSid@_@CodeObjectName@.res//
7
8 [Generator Parametr]//Параметры генерации
9 CopyObjectToRep=[1]{0|1}//Перенести объект генерации в репозиторий (ONLINE-MODE)
10 RepPath=H:\gcdrep//Путь к репозиторию
11 TemplatesPath=H:\gcdrep\dev\res\Templates//Путь к каталогу с шаблонами
12 TemporaryPath=H:\gcdrep\_tmp//Путь к каталогу временного хранения
13
14 [Object parameters]//Параметры генерируемого объекта
15 CodeObjectName=CodeGeneration//Наименование объекта (varchar(25))
16 -Pressure=34[MPa]// Давление
17 Description=Прикладное использование программного обеспечения генерации кода в процессе реализации НИР
18 ParametersFile=[EVN_MSU_LEC1_02_10_2018.imp]//Имя файла дополнительных параметров
19 TemplateSID=[EVN]$gen.tmpls//Тип генерируемого объекта из ЕД
20
21 [Project data]//Идентификация объекта
22 ComplexSID=[rk6]$sys.cmplx//Идентификатор комплекса
23 SolutionSID=[met]$sys.solun//Идентификатор решения
24 ProjectSID=[rvs]$sys.prjct//Идентификатор проекта
25
26 // Список названий используемых материалов
27 [Material Names]
28 MaterialSIDs={1,2,3}// Идентификаторы материалов в соответствии с геометрией v.2.2
29 1$name=EGLASS// Наименование материала #1
30 1$anisotropy=[auNO]{auNO|auLCS|auSegmented}// Метод учета анизотропии
31 1$LCS=((1;0;0);(0;1;0);(0;0;1))// Определение ЛСК для материала с индексом 1
32 1$anisotropy_data=@file@.ani// Информация по анизотропии
33 1$strength_criterion=SQR
34
35 2$name=EGLASS// Наименование материала #2
36 2$anisotropy=[auNO]{auNO|auLCS|auSegmented}// Метод учета анизотропии
37 2$LCS=((1;0;0);(0;1;0);(0;0;1))// Определение ЛСК для материала с индексом 2
38 2$anisotropy_data=@file@.ani// Информация по анизотропии
39 2$strength_criterion=SQR
40 arr = (1;2;3;'axdxa')
41 diap = [1;0;3;1]
```

Листинг 6. Файл aINI для тестирования подсистемы ввода-вывода

В результате были сформированы 5 секций, содержащих различные формы для ввода параметров, а также была осуществлена поддержка табличных данных:

Обработать

Author Generator Parametr Object parameters Project data Material Names

AuthorName

alsokolo \$

tbl

dscra \$

SID автора

sa

Рис. 1. Секция Author.



Первые две записи представляют собой ссылку на таблицу базы данных. Пользователю предоставлена возможность выбрать первичный ключ таблицы. При нажатии на кнопку справа от формы ввода сформируется модальное окно с таблицей. Заданной параметром вида:

[<первичный ключ>]\${<база данных>.<схема>.<id таблицы>

Также на этой форме не отображается скрытый параметр OutputFile (начинающийся атрибутом “-”)

Инструменты решения задач (решатели)								
Имя таблицы: GCDDb.com.slvrs								
Количество столбцов: 11; Количество строк: 68								
	slvr Идентификатор инструмента решения задачи (решатели)	dscra Описание	mdlid Идентификатор модели	cmsid Идентификатор вычислительного метода	ntmdl Идентификатор сетевой модели	preci Требуемая точность	itlim Лимит по итерациям	dimid Идентификатор размерности
<input type="checkbox"/>	ACKLEY_PSO_OPT	Решатель задачи поиска минимума функции Экли	ELASTIC	PSO	PSO_MODEL	1e-07	2500	3D
<input type="checkbox"/>	ANN_MLP	Анализ на основе модели многослойного персептрона	NN_MLP	ANN	MLP_MODEL			1D
<input type="checkbox"/>	ANN_MLP_TRAIN	Инструмент обучения	NN_MLP	ANN	MLP_TRAIN			1D

Рис. 2. Пример таблицы, сформированной после нажатия на кнопку.

Author
Generator Parametrs
Object parameters
Project data
Material Names

☒ Перенести объект генерации в репозиторий (ONLINE-MODE)

Путь к репозиторию

Путь к каталогу с шаблонами

Путь к каталогу временного хранения

Рис. 3. Секция Generator Parametrs. Представлены строчные пути к файлам и каталогам, а также логическая переменная для выбора объекта генерации.

Метод учета анизотропии

auNO

Определение ЛСК для материала с индексом 2

1	0	0
0	1	0
0	0	1

Информация по анизотропии

@file@.ani

2\$strength\_criterion

SQR

arr

1 2 3 'axdxa'

diap

Текущий индекс От: До: Шаг:

1 0 3 1

Рис. 4. Сгенерированная форма для ввода параметров типа множества выбора, двумерного массива, строк, а также одномерного массива и диапазонов.

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения курсовой работы был доработан интерпретатор. Добавлены следующие функциональные возможности:

- 1) обработка интерпретатором различных типов данных, поддерживающихся в формате aINI;
- 2) доработаны шаблоны для отображения параметров следующих типов данных: диапазоны, массивы(одномерные и двумерные), ссылки на таблицу базы данных;

## СПИСОК ЛИТЕРАТУРЫ

1. Федоров А.В. Современный подход в проектировании грамотного пользовательского интерфейса. //Научный вестник Воронежского Государственного Архитектурно-Строительного Университета. Сер. информационные технологии в строительных, социальных и экономических системах. - 2015. - №2. - С. 97-100.
2. Бубарева О.А. Методы проектирования эффективных экранных интерфейсов. //Информация и образование. - 2018 - №10. - С. 91-94.
3. Соколов А.П., Першин А.Ю. Программный инструментарий для создания подсистем ввода данных при разработке систем инженерного анализа. //Программная инженерия – 2017 - №8 – С. 543 – 552.
4. Литвинов В.Л., Онтологическое проектирование пользовательских интерфейсов//Информационные системы и технологии в моделировании и управлении. Сборник материалов III Всероссийской научно-практической конференции с международным участием, посвященной 100-летию Крымского федерального университета имени В.И. Вернадского. - 2018 – стр. 33-37
5. Чернов В.В. К проблеме разработки Веб – интерфейсов. //Фундаментальные исследования. - 2012 - №11-2. - С. 463 - 465.
6. Грибова В.В., Черкезишвили Н.Н., Развитие онтологического подхода для автоматизации разработки пользовательских интерфейсов с динамическими данными. //Информационные технологии, 2010, №10, -С. 54–58.
7. Глазков С.В., Ронжин А.Л., Контекстно-зависимые методы автоматической генерации многомодальных пользовательских веб-интерфейсов. //Тр. СПИИРАН. - 2012. - 21(2012) – С.170–183.
8. Zhizhimov O. L. Explain Services on ZooSPACE Platform and Adaptive User Interfaces // CEUR Workshop Proceedings. 2015. Vol. 1536. P. 30—36.
9. Пискунов С. В., Кратов С. В., Остапкевич М. Б., Веселов А. В. Использование сборочной технологии для построения пользовательских

интерфейсов сетевой информационно-вычислительной системы //

Проблемы информатики. 2010. № 4. С. 41—48

10. Ramon O. S., Cuadrado J. S., Molina J. G. Model-driven reverse engineering of legacy graphical user interfaces // Automated Software Engineering. 2014. Vol. 21, Issue 2. P. 147—186. DOI: 10.1007/s10515-013-0130-2.
11. Нешляева А.В. Актуальность инструмента построения графического интерфейса пользователя в облачных средах разработки // Альманах современной науки и образования. - 2013 - №5 – С. 134-136.