

УТВЕРЖДАЮ
Начальник тех.отдела
_____ Губин А.Н.
“ _____ ” _____ 2017

МОДУЛЬ ПРОГРАММНОЙ ДОКУМЕНТАЦИИ «СПБ-ГУТ»

Текст программы

ЛИСТ УТВЕРЖДЕНИЯ

643.ГУТ.1302016-01 12 01-ЛУ

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Руководитель разработки
Начальник ИСТ-311
_____ Губин А.Н.
“ _____ ” _____ 2017

Ответственный исполнитель
Начальник гр. РиВ АСУТП
ИСТ-311
_____ Турашов И.М.
“ _____ ” _____ 2017

Исполнитель
Вед. инженер ИСТ-311
_____ Булыгин Г.А.
“ _____ ” _____ 2017

УТВЕРЖДЕНО

643.ГУТ.1302016-01 12 01-ЛУ

МОДУЛЬ ПРОГРАММНОЙ ДОКУМЕНТАЦИИ «СПБ-ГУТ»

Текст программы

643.ГУТ.1302016-01 12 01

Листов 11

Инь. № подл.	Подпись и дата	Взам. инв. №	Инь. № дубл.	Подпись и дата

АННОТАЦИЯ

В данном программном документе приведен частичный текст модуля документации, предназначенного для отображения документов, созданных в ходе проектирования корпоративной информационной системы.

Полный текст всех html-документов не приводится, так как многие из них являются типовыми.

Основная задача модуля документации – обеспечить доступность, структурировать программную, сетевую и рабочую документацию.

Модуль документации реализует следующие функции:

- Отображение программной, сетевой и рабочей документации
- Отображение информации о разработчиках
- Добавление и удаление программной, сетевой или рабочей документации

Данные функции модуля документации позволяют поддерживать организованный доступ к проектной документации.

Оформление программного документа «Текст программы» произведено по требованиям ЕСПД (ГОСТ 19.101-77 ¹, ГОСТ 19.103-77 ², ГОСТ 19.104-78* ³, ГОСТ 19.105-78* ⁴, ГОСТ 19.106-78* ⁵, ГОСТ 19.401-78 ⁶, ГОСТ 19.604-78* ⁷).

¹ ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов

² ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов

³ ГОСТ 19.104-78* ЕСПД. Основные надписи

⁴ ГОСТ 19.105-78* ЕСПД. Общие требования к программным документам

⁵ ГОСТ 19.106-78* ЕСПД. Общие требования к программным документам, выполненным печатным способом

⁶ ГОСТ 19.401-78 ЕСПД. Текст программы. Требования к содержанию и оформлению

⁷ ГОСТ 19.604-78* ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом

СОДЕРЖАНИЕ

Аннотация	2
Содержание	3
1. Текст html-документов модуля документации	4
2. Текст сервера модуля документации	6
Лист регистрации изменений	11

1. ТЕКСТ HTML-ДОКУМЕНТОВ МОДУЛЯ ДОКУМЕНТАЦИИ

Основной документ index.html (осуществляет запуск модуля, задающего фреймовую структуру):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <link href="styles.css" rel="stylesheet">
</head>
<frameset cols="*">
  <frame src="title.html" name="main">
</frameset>
</html>
```

nav.html (панель навигации):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link href="styles.css" rel="stylesheet">
</head>
<body>
  <ul>
    <li><a target="content" href="/type_doc.html">Просмотр</a></li>
    <li><a target="content" href="add_doc.html">Добавить комплект документов</a></li>
    <li><a target="content" href="del_doc.html">Удалить комплект документов</a></li>
  </ul>
  <input type="button" onclick="history.back();" value="Назад"/>
</body>
</html>
```

type_doc.html (html-документ для отображения видов документации):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link href="styles.css" rel="stylesheet">
</head>
<body>
  <ul>
    <li><a target="content" href="work_doc">Рабочая документация</a></li>
    <li><a target="content" href="net_doc">Сетевая документация</a></li>
    <li><a target="content" href="program_doc">Программная документация</a></li>
  </ul>
</body>
</html>
```

net_doc.html (типовый фреймовый документ для просмотра документации, содержащий специальные шаблонные теги):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Сетевая документация</title>
  <link href="styles.css" rel="stylesheet">
</head>
<body>
  <ol>
    {% for doc, name in docs %}
      <li><a href="{{ doc }}">{{ name }}</a></li>
    {% endfor %}
  </ol>
</body>
</html>
```

add_doc.html (типовый документ для добавления документации документации):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link href="styles.css" rel="stylesheet">
</head>
<body>
  <ul>
    <li><a target="content" href="add_work_doc.html">Добавить рабочую документацию</a></li>
    <li><a target="content" href="add_net_doc">Добавить сетевую документацию</a></li>
    <li><a target="content" href="add_prog_doc">Добавить программную документацию</a></li>
  </ul>
</body>
</html>
```

del_doc.html (типовый документ для удаления документации документации):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link href="styles.css" rel="stylesheet">
</head>
<body>
  <form action="/del" method="POST" enctype="multipart/form-data">
    Удалить файл: <input type="text" name="del">
    <input type="submit" value="submit ">
  </form>
</body>
</html>
```

2. ТЕКСТ СЕРВЕРА МОДУЛЯ ДОКУМЕНТАЦИИ

```
#!/usr/bin/env python
import os

from flask import Flask, send_from_directory, render_template, request, redirect, url_for, flash
from werkzeug.utils import secure_filename

UPLOAD_FOLDER = 'doc'
ALLOWED_EXTENSIONS = set(['txt', 'pdf'])

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

@app.route('/<path:filename>')
def index(filename="index.html"):
    return send_from_directory('.', filename)

@app.route('/work_doc')
def work_doc():
    docs = os.walk('doc/work_doc')
    elements = []
    for el in docs:
        elements.append(el)

    docs = {}
    scheme = []
    for doc in elements[0][2]:
        scheme.append((elements[0][0] + '/' + doc, doc[:-4]))
    docs['schemes'] = scheme

    cab_con = []
    for doc in elements[1][2]:
        cab_con.append((elements[1][0] + '/' + doc, doc[:-4]))
    docs['cab_cons'] = cab_con

    equip = []
    for doc in elements[4][2]:
        equip.append((elements[4][0] + '/' + doc, doc[:-4]))
    docs['equips'] = equip

    rooms = []
    for doc in elements[2][2]:
        rooms.append((elements[2][0] + '/' + doc, doc[:-4]))
    docs['rooms'] = rooms

    tracts = []
    for doc in elements[3][2]:
        tracts.append((elements[3][0] + '/' + doc, doc[:-4]))
    docs['tracts'] = tracts

    return render_template('work_doc.html', docs=docs)
```

```

@app.route('/net_doc')
def net_doc():
    docs = os.walk('doc/net_doc')
    elements = []
    for el in docs:
        elements.append(el)

    docs = []
    for doc in elements[0][2]:
        docs.append((elements[0][0] + '/' + doc, doc[: -4]))
    return render_template('net_doc.html', docs=docs)

@app.route('/prog_doc')
def prog_doc():
    docs = os.walk('doc/prog_doc')
    elements = []
    for el in docs:
        elements.append(el)

    docs = []
    for doc in elements[0][2]:
        docs.append((elements[0][0] + '/' + doc, doc[: -4]))
    return render_template('prog_doc.html', docs=docs)

def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

@app.route('/add_net_doc', methods=['GET', 'POST'])
def net_doc_upload_file():
    if request.method == 'POST':
        # check if the post request has the file part
        if 'file' not in request.files:
            flash('No file part')
            return redirect(request.url)
        file = request.files['file']

        if file.filename == '':
            flash('No selected file')
            return redirect(request.url)
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'] + "/net_doc", filename))

            return redirect(url_for('net_doc'))
    return """
<!doctype html>
<title>Загрузить сетевую документ</title>
<h1>Загрузить новую сетевую документацию</h1>
<form method=post enctype=multipart/form-data>
<p><input type=file name=file>
<input type=submit value=Загрузить>
</form>
"""

```



```

@app.route('/add_prog_doc', methods=['GET', 'POST'])
def prog_doc_upload_file():
    if request.method == 'POST':
        # check if the post request has the file part
        if 'file' not in request.files:
            flash('No file part')
            return redirect(request.url)
        file = request.files['file']

        if file.filename == "":
            flash('No selected file')
            return redirect(request.url)
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'] + "/prog_doc", filename))

            return redirect(url_for('prog_doc'))
    return """
<!doctype html>
<title>Загрузить программную документ</title>
<h1>Загрузить новую программную документацию</h1>
<form method=post enctype=multipart/form-data>
<p><input type=file name=file>
<input type=submit value=Загрузить>
</form>
"""

@app.route('/add_cab_con', methods=['GET', 'POST'])
def cab_con_upload_file():
    if request.method == 'POST':
        # check if the post request has the file part
        if 'file' not in request.files:
            flash('No file part')
            return redirect(request.url)
        file = request.files['file']

        if file.filename == "":
            flash('No selected file')
            return redirect(request.url)
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'] + "/work_doc/cab_con", filename))

            return redirect(url_for('work_doc'))
    return """
<!doctype html>
<title>Загрузить схему кабельных соединений</title>
<h1>Загрузить новую схему кабельных соединений</h1>
<form method=post enctype=multipart/form-data>
<p><input type=file name=file>
<input type=submit value=Загрузить>
</form>
"""

```

```

@app.route('/add_equip', methods=['GET', 'POST'])
def equip_upload_file():
    if request.method == 'POST':
        # check if the post request has the file part
        if 'file' not in request.files:
            flash('No file part')
            return redirect(request.url)
        file = request.files['file']

        if file.filename == "":
            flash('No selected file')
            return redirect(request.url)
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'] + "/work_doc/equip", filename))

            return redirect(url_for('work_doc'))
    return ""
<!doctype html>
<title>Загрузить схему расположения оборудования в шкафах</title>
<h1>Загрузить новую схему расположения оборудования в шкафах</h1>
<form method=post enctype=multipart/form-data>
<p><input type=file name=file>
<input type=submit value=Загрузить>
</form>

```

```

@app.route('/add_rooms', methods=['GET', 'POST'])
def rooms_upload_file():
    if request.method == 'POST':
        # check if the post request has the file part
        if 'file' not in request.files:
            flash('No file part')
            return redirect(request.url)
        file = request.files['file']

        if file.filename == "":
            flash('No selected file')
            return redirect(request.url)
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'] + "/work_doc/rooms", filename))

            return redirect(url_for('work_doc'))
    return ""
<!doctype html>
<title>Загрузить план расположения оборудования и кабельных трасс</title>
<h1>Загрузить новый план расположения оборудования и кабельных трасс</h1>
<form method=post enctype=multipart/form-data>
<p><input type=file name=file>
<input type=submit value=Загрузить>
</form>

```

```
@app.route('/add_tracts', methods=['GET', 'POST'])
def tracts_upload_file():
    if request.method == 'POST':
        # check if the post request has the file part
        if 'file' not in request.files:
            flash('No file part')
            return redirect(request.url)
        file = request.files['file']

        if file.filename == '':
            flash('No selected file')
            return redirect(request.url)
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'] + "/work_doc/tracts", filename))

            return redirect(url_for('work_doc'))
    return ""
<!doctype html>
<title>Загрузить схему прохождения трактов</title>
<h1>Загрузить новую схему прохождения трактов</h1>
<form method=post enctype=multipart/form-data>
<p><input type=file name=file>
<input type=submit value=Загрузить>
</form>
"""

@app.route('/del', methods=['POST'])
def del_doc():
    if request.method == 'POST':
        os.remove(request.form['upfile'])
        print(request.form['upfile'])
        return redirect("type_doc.html")

if __name__ == "__main__":
    app.run(debug=False, port=8000)
```

[illegible]