

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Национальный исследовательский университет

"Высшая Школа Экономики"

Факультет компьютерных наук

Департамент программной инженерии

СОГЛАСОВАНО

Научный руководитель, приглашенный
преподаватель департамента
программной инженерии

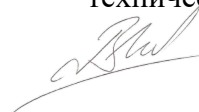


Сосновский Г. М.

«13» мая 2023 г.

УТВЕРЖДАЮ

Академический руководитель
образовательной программы
«Программная инженерия», кандидат
технических наук



Шилов В. В.

«13» мая 2023 г.

Генератор документации "Радость Научника"

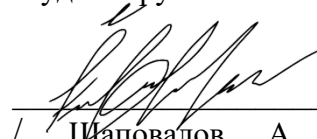
Текст программы

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.06.11-01 12 01-1-ЛУ

Исполнитель:

студент группы БПИ217



/ Шаповалов А. С. /

«13» мая 2023 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	RU.17701729.06.11-01 12 01-1

Москва 2023

УТВЕРЖДЕН

RU.17701729.06.11-01 12 01-1-ЛЮ

Инв. № подл	Подп. и дата	Инв. № дубл.	Подп. и дата	Взам. инв. №
RU.17701729.06.11-01 12 01-1				

Генератор документации "Радость Научника"
Текст программы
RU.17701729.06.11-01 12 01-1
Листов 20

Москва 2023

СОДЕРЖАНИЕ

1	ТЕКСТ ПРОГРАММЫ.....	4
1.1	Вот.PY	4
1.2	TECH_SPEC_DATA.PY	14
1.3	EXPLANATORY_NOTE_DATA.PY	17
1.4	TITLE_PAGE_DATA.PY.....	19

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

АННОТАЦИЯ

«Генератора документации "Радость Научника

В настоящем документе «Текст программы» приведен текст «Генератора документации "Радость Научника"». Программа разработана на языке Python.

Среда разработки – PyCharm 2021.1.

Настоящий документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.101-77 Виды программ и программных документов;
- 2) ГОСТ 19.102-77 Стадии разработки;
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов;
- 4) ГОСТ 19.104-78 Основные надписи;
- 5) ГОСТ 19.105-78 Общие требования к программным документам;
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом;
- 7) ГОСТ 19.401-78 Текст программы. Требования к содержанию и оформлению.

Изменения к данному документу оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

1 ТЕКСТ ПРОГРАММЫ

1.1 Репозиторий

Код программы находится в публичном GitHub-репозитории по ссылке:

<https://github.com/shapovv/MentorsJoy>

1.2 Bot.py

```
import logging
import os
import time
import uuid

from docx2pdf import convert
from docxtpl import DocxTemplate
from dotenv import load_dotenv
import telebot
from telebot import types

from explanatory_note_data import explanatory_note_questions
from tech_spec_data import tech_spec_questions
from title_page_data import title_page_questions

OUTPUT_FOLDER = "output_files"

if not os.path.exists(OUTPUT_FOLDER):
    os.makedirs(OUTPUT_FOLDER)

logging.basicConfig(filename='errors.log', level=logging.ERROR, format='%(asctime)s -
%(message)s',
                    datefmt='%Y-%m-%d %H:%M:%S')

load_dotenv()
bot = telebot.TeleBot(os.getenv('TOKEN'))
last_chat_id = None

chat_steps = {}

current_document_type = {}

title_page_code_blocks = {}
tech_spec_code_blocks = {}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

```
explanatory_note_code_blocks = {}
file_id_dict = {}
```

```
def generate_file_id():
    return str(uuid.uuid4())
```

```
def store_file_id(filename):
    file_id = generate_file_id()
    file_id_dict[file_id] = filename
    return file_id
```

```
def get_filename_by_file_id(file_id):
    return file_id_dict.get(file_id)
```

```
def create_code_blocks(chat_id):
    title_page_code_blocks[chat_id] = {key: " for _, key in title_page_questions}
    tech_spec_code_blocks[chat_id] = {key: " for _, key in tech_spec_questions}
    explanatory_note_code_blocks[chat_id] = {key: " for _, key in
explanatory_note_questions}
```

```
def create_ask_and_save_handlers(code_blocks, question, code_block_key):
    def ask_handler(message):
        msg = bot.send_message(message.chat.id, question, parse_mode='HTML')
        bot.register_next_step_handler(msg, save_handler)
```

```
    def save_handler(message):
        if message.text.startswith("/start"):
            handle_start_after_creation(message)
            return
        if message.text.startswith("/stop"):
            handle_stop(message)
            return
        chat_id = message.chat.id
        if chat_id not in code_blocks:
            code_blocks[chat_id] = {}
        code_blocks[chat_id][code_block_key] = message.text
        next_step_handler(message, chat_steps[message.chat.id])
```

```
    return ask_handler, save_handler
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

```

title_page_handlers = [create_ask_and_save_handlers(title_page_code_blocks, question,
key) for question, key
                        in title_page_questions]
tech_spec_handlers = [create_ask_and_save_handlers(tech_spec_code_blocks, question,
key) for question, key in
                      tech_spec_questions]
explanatory_note_handlers =
[create_ask_and_save_handlers(explanatory_note_code_blocks, question, key) for question, key
  in explanatory_note_questions]

```

```

def next_step_handler(message, step=0):
    global last_chat_id
    last_chat_id = message.chat.id
    chat_steps[message.chat.id] = step + 1
    document_type = current_document_type[message.chat.id]
    if document_type == "title_page":
        if step < len(title_page_handlers):
            ask_handler, _ = title_page_handlers[step]
            ask_handler(message)
        elif step == len(title_page_handlers):
            create_document(message, title_page_code_blocks[message.chat.id],
'title_page_maket.docx',
                          'title_page')
        elif document_type == "technical_specifications":
            if step < len(tech_spec_handlers):
                ask_handler, _ = tech_spec_handlers[step]
                ask_handler(message)
            elif step == len(tech_spec_handlers):
                create_document(message, tech_spec_code_blocks[message.chat.id],
'tech_spec_maket.docx',
                          'technical_specifications')
        elif document_type == "explanatory_note":
            if step < len(explanatory_note_handlers):
                ask_handler, _ = explanatory_note_handlers[step]
                ask_handler(message)
            elif step == len(explanatory_note_handlers):
                create_document(message, explanatory_note_code_blocks[message.chat.id],
'explanatory_note_maket.docx',
                          'explanatory_note')

```

```

def start(message):
    keyboard = types.InlineKeyboardMarkup()
    button1 = types.InlineKeyboardButton("Титульный лист",
callback_data="title_page")

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

```

        button2 = types.InlineKeyboardButton("Техническое задание",
callback_data="technical_specifications")
        button3 = types.InlineKeyboardButton("Пояснительная записка",
callback_data="explanatory_note")

        keyboard.add(button1)
        keyboard.add(button2)
        keyboard.add(button3)

        bot.send_message(message.chat.id, "Выберите документ, который хотите создать.",
reply_markup=keyboard)

```

```

@bot.message_handler(
    func=lambda message: message.text.startswith("/start") and message.chat.id in
current_document_type)
def handle_start_after_document_creation(message):
    handle_start_after_creation(message)

def handle_start_after_creation(message):
    reset_document_creation(message.chat.id)
    bot.send_message(message.chat.id, "Приступаем к созданию нового документа.")
    start(message)

# def handle_start_after_error(message):
#     reset_document_creation(message.chat.id)
#     bot.send_message(message.chat.id, "Возникла ошибка, приступите к созданию
нового документа.")
#     start(message)

def reset_document_creation(chat_id):
    if chat_id in current_document_type:
        del current_document_type[chat_id]
    if chat_id in chat_steps:
        del chat_steps[chat_id]
    if chat_id in title_page_code_blocks:
        del title_page_code_blocks[chat_id]
    if chat_id in tech_spec_code_blocks:
        del tech_spec_code_blocks[chat_id]
    if chat_id in explanatory_note_code_blocks:
        del explanatory_note_code_blocks[chat_id]

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и


```

@bot.message_handler(func=lambda message: message.text == "/start" and
message.chat.id in current_document_type)
def handle_start_after_document_creation(message):
    handle_start_after_creation(message)

def update_document_choice_to_inactive(chat_id, message_id,
selected_document_type):
    keyboard = types.InlineKeyboardMarkup()

    button1 = types.InlineKeyboardButton("Титульный лист" + ("✅" if
selected_document_type == "title_page" else "❌"),
callback_data="ignore")
    button2 = types.InlineKeyboardButton(
"Техническое задание" + ("✅" if selected_document_type ==
"technical_specifications" else "❌"),
callback_data="ignore")
    button3 = types.InlineKeyboardButton(
"Пояснительная записка" + ("✅" if selected_document_type ==
"explanatory_note" else "❌"),
callback_data="ignore")

    keyboard.add(button1)
    keyboard.add(button2)
    keyboard.add(button3)

    bot.edit_message_text("Выберите документ, который хотите создать:",
chat_id=chat_id, message_id=message_id,
reply_markup=keyboard)

def send_restart_choice(message):
    keyboard = types.InlineKeyboardMarkup()
    button1 = types.InlineKeyboardButton("Создать новый документ",
callback_data="restart")
    button2 = types.InlineKeyboardButton("Закончить", callback_data="finish")

    keyboard.add(button1)
    keyboard.add(button2)

    sent_message = bot.send_message(message.chat.id, "Что вы хотите сделать
дальше?", reply_markup=keyboard)
    return sent_message.message_id

def update_buttons_to_inactive(chat_id, message_id, selected_button):

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

```

keyboard = types.InlineKeyboardMarkup()
button1 = types.InlineKeyboardButton(
    "Создать новый документ" + ("✅" if selected_button == "new_document" else "
❌"), callback_data="ignore")
button2 = types.InlineKeyboardButton("Закончить" + ("✅" if selected_button ==
"finish" else "❌"),
    callback_data="ignore")

button1.url = None
button2.url = None

keyboard.add(button1)
keyboard.add(button2)

bot.edit_message_text("Что вы хотите сделать дальше?", chat_id=chat_id,
message_id=message_id,
    reply_markup=keyboard)

@bot.callback_query_handler(func=lambda call: call.data == "finish")
def finish_handler(call):
    update_buttons_to_inactive(call.message.chat.id, call.message.message_id, "finish")
    bot.send_message(call.message.chat.id,
        "Благодарим Вас за использование нашего бота!\n\nС уважением,
<b>Mentors Joy</b>.",
        parse_mode='HTML')

def ask_for_conversion(message, filename):
    keyboard = types.InlineKeyboardMarkup()
    file_id = store_file_id(filename)
    button1 = types.InlineKeyboardButton("Конвертировать",
callback_data=f'convert_to_pdf,{file_id}')
    button2 = types.InlineKeyboardButton("Не нужно",
callback_data=f'no_conversion,{file_id}')
    keyboard.add(button1, button2)

    bot.send_message(message.chat.id,
        '<b>Хотите конвертировать файл в формат PDF?</b>\n\n'
        'Конвертация создает повышенную нагрузку на наш сервер, поэтому
если нет острой необходимости, '
        'откажитесь от нее, пожалуйста. Если же конвертация необходима,
нажмите на кнопку один раз и '
        'подождите 10-15 секунд.',
        parse_mode='HTML', reply_markup=keyboard)

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

```
@bot.callback_query_handler(func=lambda call: call.data.startswith('convert_to_pdf'))
def convert_to_pdf_callback(call):
    _, file_id = call.data.split(',', 1)
    filename = get_filename_by_file_id(file_id)

    pdf_filename = os.path.splitext(filename)[0] + '.pdf'
    # os.chmod("/Users/shapovv/PycharmProjects/Radost/output_files", 0o777)
    os.chmod(filename, 0o777)
    convert(filename, pdf_filename)

    with open(pdf_filename, 'rb') as file:
        bot.send_document(call.message.chat.id, file)

# Изменяем кнопки
update_conversion_buttons(call, '✅ Конвертировать', '❌ Не нужно', True)
send_restart_choice(call.message)
```

```
@bot.callback_query_handler(func=lambda call: call.data.startswith('no_conversion'))
def no_conversion_callback(call):
    _, file_id = call.data.split(',', 1)
    filename = get_filename_by_file_id(file_id)

    # Изменяем кнопки
    update_conversion_buttons(call, '❌ Конвертировать', '✅ Не нужно', True)
    send_restart_choice(call.message)
```

```
def update_conversion_buttons(call, conversion_text, no_conversion_text,
disable_buttons):
    markup = telebot.types.InlineKeyboardMarkup()

    conversion_button = telebot.types.InlineKeyboardButton(
        text=conversion_text,
        callback_data='dummy_data',
        disabled=disable_buttons
    )
    no_conversion_button = telebot.types.InlineKeyboardButton(
        text=no_conversion_text,
        callback_data='dummy_data',
        disabled=disable_buttons
    )

    markup.add(conversion_button, no_conversion_button)
    bot.edit_message_text(chat_id=call.message.chat.id,
        message_id=call.message.message_id,
        text=call.message.text,
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

```
reply_markup=markup)
```

```
@bot.callback_query_handler(func=lambda call: call.data == 'dummy_data')
def dummy_callback(call):
    pass
```

```
def create_document(message, code_blocks, template_name, output_name):
    template = DocxTemplate(template_name)
    template.render(code_blocks)
    second_name = code_blocks['second_name']
    year = code_blocks['year']
    filename = os.path.join(OUTPUT_FOLDER,
f'{output_name}_{second_name}_{year}.docx')
    template.save(filename)
    os.chmod(filename, 0o777)
```

```
with open(filename, 'rb') as file:
    bot.send_document(message.chat.id, file)
```

```
ask_for_conversion(message, filename)
```

```
# Удалить .docx и .pdf файлы после отправки?
# os.remove(filename)
# os.remove(pdf_filename)
```

```
@bot.callback_query_handler(func=lambda call: call.data == "restart")
def restart_handler(call):
    update_buttons_to_inactive(call.message.chat.id, call.message.message_id,
"new_document")
    start(call.message)
```

```
def handle_stop(message):
    global last_chat_id
    last_chat_id = message.chat.id
    bot.send_message(message.chat.id, 'Процесс создания документа прерван.')
    document_type = current_document_type[message.chat.id]
    if document_type == "technical_specifications":
        create_document(message, tech_spec_code_blocks[message.chat.id],
'tech_spec_maket.docx',
'technical_specifications')
    elif document_type == "explanatory_note":
        create_document(message, explanatory_note_code_blocks[message.chat.id],
'explanatory_note_maket.docx',
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

```

'explanatory_note')
elif document_type == "title_page":
    create_document(message, title_page_code_blocks[message.chat.id],
'title_page_maket.docx',
'title_page')

@bot.callback_query_handler(func=lambda call: True)
def document_template_handler(call):
    global last_chat_id
    last_chat_id = call.message.chat.id
    document_type = call.data

    if document_type == "ignore":
        return # Завершить обработчик, если callback_data равно "ignore"

    update_document_choice_to_inactive(call.message.chat.id, call.message.message_id,
document_type)

    current_document_type[call.message.chat.id] = document_type
    if document_type == "technical_specifications":
        bot.send_message(call.message.chat.id,
            "<b>Приступим к созданию вашего технического задания.</b>\n"
            "Сейчас мы поэтапно пройдем по всем пунктам, начиная с
титульного листа и заканчивая списком литературы. "
            "Следуйте дальнейшим указаниям и подсказкам. \n\nВ любом месте
вы можете остановиться и отправить ответ позднее, "
            "а если захотите закончить и получить не до конца заполненный
документ, введите /stop. \n\n"
            "Если вы хотите пропустить любой шаг, необходимо отправить в
ответ любой символ, тогда вы сможете "
            "заполнить этот раздел самостоятельно в итоговом документе.",
            parse_mode='HTML')
        chat_steps[call.message.chat.id] = 0
        next_step_handler(call.message, step=0)
    elif document_type == "explanatory_note":
        bot.send_message(call.message.chat.id,
            "<b>Приступим к созданию вашей пояснительной
записки.</b>\nСейчас мы поэтапно пройдем по всем пунктам, "
            "начиная с титульного листа и заканчивая списком литературы.
Следуйте дальнейшим указаниям и подсказкам. "
            "\n\nВ любом месте вы можете остановиться и отправить ответ
позднее, а если захотите закончить и получить "
            "не до конца заполненный документ, введите /stop. \n\nЕсли вы
хотите пропустить любой шаг, необходимо отправить "
            "в ответ любой символ, тогда вы сможете заполнить этот раздел
самостоятельно в итоговом документе.",

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

```

        parse_mode='HTML')
    chat_steps[call.message.chat.id] = 0
    next_step_handler(call.message, step=0)
elif document_type == "title_page":
    bot.send_message(call.message.chat.id,
        "<b>Приступим к созданию вашего титульного листа.</b>\nСейчас
мы поэтапно пройдем по всем пунктам, "
        "следуйте дальнейшим указаниям и подсказкам. \n\nВ любом месте
вы можете остановиться и отправить ответ "
        "позднее, а если захотите закончить и получить не до конца
заполненный документ, введите /stop. \n\nЕсли вы "
        "хотите пропустить любой шаг, необходимо отправить в ответ
любой символ, тогда вы сможете заполнить этот раздел "
        "самостоятельно в итоговом документе.",
        parse_mode='HTML')
    chat_steps[call.message.chat.id] = 0
    next_step_handler(call.message, step=0)

```

```

@bot.message_handler(commands=['start'])
def handle_start(message):
    global last_chat_id
    last_chat_id = message.chat.id
    create_code_blocks(message.chat.id)
    bot.send_message(message.chat.id, 'Добро пожаловать в бота <b>"Mentors
Joy"</b>!\n\n'
        'Я помогу вам создать профессиональную техническую
документацию, '
        'включая Титульный лист, Техническое задание и
Пояснительную записку. '
        'Все документы будут оформлены согласно ГОСТам, что
обеспечит точность и качество '
        'вашей работы.', parse_mode='HTML')
    start(message)

```

```

@bot.message_handler(commands=['stop'])
def handle_stop_command(message):
    handle_stop(message)

```

```

@bot.message_handler(commands=['raise_error'])
def handle_raise_error_command(message):
    global last_chat_id
    last_chat_id = message.chat.id
    raise ValueError("Тестовое исключение")

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

```

def send_error_message(error_message, chat_id=None, user_id=None):
    global last_chat_id
    if chat_id is None and last_chat_id is not None:
        chat_id = last_chat_id

    if chat_id is not None:
        try:
            bot.send_message(chat_id, error_message)
            bot.send_message(chat_id, "Чтобы начать создание документа заново введите
сначала /stop, а затем /start.")

        except Exception as e:
            logging.error(f'Failed to send error message: {e}')
    else:
        if user_id is None:
            user_id_str = "unknown"
        else:
            user_id_str = str(user_id)
        logging.error(f'Cannot send error message, chat_id is missing (user_id:
{user_id_str}): {error_message}')

if __name__ == '__main__':
    while True:
        try:
            bot.polling(none_stop=True)
        except Exception as e:
            user_id = None
            if last_chat_id is not None:
                try:
                    user_id = bot.get_chat(last_chat_id).id
                except Exception:
                    pass
            logging.error(f'Error: {e}')
            send_error_message(f'Произошла ошибка: {e}', user_id=user_id)
            time.sleep(10)

```

1.3 Tech_spec_data.py

```

tech_spec_questions = [
    (
        '<b>Сначала заполним титульный лист.</b>\n\nВведите полное название ВУЗа,
например "Национальный исследовательский университет Высшая Школа Экономики":',

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

'university'),
 ('Введите факультет, например "Факультет компьютерных наук":', 'faculty'),
 ('Введите департамент, например "Департамент программной инженерии":',
 'department'),
 ('Введите название работы, например «Генератор документации "Радость
 Научника"»: ', 'project_name'),
 ('Введите название работы на английском языке, например «Documentation
 Constructor “Mentors Joy”»: ',
 'project_name_eng'),
 (
 'Введите номер работы:\n\nПодсказка: в соответствии с ГОСТ 19.103-77
 "Обозначения программ и программных документов", номер имеет вид
 RU.12345678.123456-01, где RU - код страны, 12345678 - код организации-разработчик,
 123456-01 - регистрационный номер, который присваивается в соответствии с ОКП.:',
 'number'),
 ('Введите ФИО исполнителя в формате Фамилия И. О.:', 'student_name'),
 ('Введите номер группы, например "БПИ217":', 'group_number'),
 ('Введите год:', 'year'),
 ('Введите ФИО научного руководителя в формате Фамилия И. О.:',
 'supervisor_name'),
 ('Введите ФИО академического руководителя в формате Фамилия И. О.:',
 'akad_name'),
 (
 'Переходим к заполнению раздела "Введение".\n\nНапишите краткую
 характеристику области применения программы, например "«Генератор документации
 “Радость Научника”» — прикладная программа, разрабатываемая с целью облегчения
 формирования и оформления документации.':',
 'brief_description'),
 (
 'Переходим к разделу "Основания для разработки".\n\nВведите
 основания для разработки, это может быть, например, указ ректора или учебный план:',
 'grounds_for_development'),
 ('Переходим к разделу "Назначение разработки".\n\nВведите
 функциональное назначение:', 'functional_purpose'),
 ('Введите эксплуатационное назначение:', 'operational_purpose'),
 ('Переходим к разделу "Требования к программе".\n\nВведите требования
 к составу выполняемых функций:',
 'requirements_functions_performed'),
 ('Введите требования к организации входных данных:', 'organization_input_data'),
 ('Введите требования организация выходных данных:', 'organization_output_data'),
 ('Введите требования к временным характеристикам:', 'requirements_time'),
 ('Введите требования к интерфейсу:', 'interface_requirements'),
 ('Введите требования к контролю входной информации:',
 'control_input_information'),
 ('Введите требования к контролю выходной информации:',
 'control_output_information'),
 ('Введите требования к восстановлению после отказа:', 'recovery_time'),

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

('Введите требования к климатическим условиям эксплуатации:',
'climatic_conditions'),
('Введите требования к видам обслуживания:', 'types_services'),
('Введите требования к численности и квалификации персонала:',
'number_and_qual_personnel'),
('Введите требования к составу и параметрам технических средств:',
'parameters_technical_means'),
('Введите требования к исходным кодам и языкам программирования:',
'programming_languages'),
('Введите требования к программным средствам, используемым программой:',
'software'),
('Введите требования к защите информации и программ:', 'protection_information'),
('Введите требования к маркировке и упаковке:', 'labeling_and_packaging'),
('Введите требования к транспортированию и хранению:',
'transportation_and_storage'),
('Введите специальные требования:', 'special_requirements'),
(
 'Переходим к разделу "Требования к документации".\n\nВведите
специальные требования к программной документации:',
 'special_requirements_for_documentation'),
(
 'Следующий раздел "Техникоэкономические показатели".\n\nОпишите
ориентировочную экономическую эффективность:',
 'economic_efficiency'),
('Опишите предполагаемую потребность:', 'estimated_need'),
(
 'Заполним сравнительную таблицу с аналогами.\n\nПо умолчанию она
имеет вид таблицы с тремя аналогами и пятью сравнительными характеристиками. В
итоговом файле вы сможете изменить их количество.\n\nВведите название первого
аналога:',
 'analogue_1'),
('Введите название второго аналога:', 'analogue_2'),
('Введите название третьего аналога:', 'analogue_3'),
('Введите первую сравнительную характеристику:', 'characteristic_1'),
('Введите вторую сравнительную характеристику:', 'characteristic_2'),
('Введите третью сравнительную характеристику:', 'characteristic_3'),
('Введите четвертую сравнительную характеристику:', 'characteristic_4'),
('Введите пятую сравнительную характеристику:', 'characteristic_5'),
('Следующий раздел "Порядок контроля и приемки".\n\nВведите виды
испытаний:', 'types_of_tests'),
('Введите общие требования к приемке работы:',
'general_requirements_acceptance_of_work'),
(
 'Заключительный раздел "Список использованных источников".\n\nПо
умолчанию в него включены 10 основных ГОСТ-ов, по которым оформляется техническое
задание. Чтобы добавить свои источники, введите их через перевод строки в одном
сообщении:',

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

```
'sources')
]
```

1.4 Explanatory_note_data.py

```
explanatory_note_questions = [
    (
        '<b>Чтобы в дальнейшем корректно сохранить документ, ответьте на два вопроса:</b>\n\n1) Введите год выполнения работы, например "2023":',
        'year'),
    ('2) Введите ваше ФИО на английском языке в формате "Ivanov_A_S":',
    'second_name'),
    (
        '<b>Отлично! Теперь заполним титульный лист.</b>\n\nВведите полное название ВУЗа, например "Национальный исследовательский университет Высшая Школа Экономики":',
        'university'),
    ('Введите факультет, например "Факультет компьютерных наук":', 'faculty'),
    ('Введите департамент, например "Департамент программной инженерии":',
    'department'),
    ('Введите название работы, например «Генератор документации "Радость Научника"»:', 'project_name'),
    ('Введите название работы на английском языке, например «Documentation Constructor "Mentors Joy"»:',
    'project_name_eng'),
    (
        'Введите номер работы:\n\nПодсказка: в соответствии с ГОСТ 19.103-77 "Обозначения программ и программных документов", номер имеет вид <b>A.B.XXXXX-XX</b>, где:\nA - Код страны; \nB - код организации-разработчика; \nXXXXX - регистрационный номер, который присваивается в соответствии с ОКП; \nXX - Номер редакции (в диапазоне 1-99).\n\nПодробнее можно прочитать по ссылке - https://www.swrit.ru/doc/espd/19.103-77.pdf:',
        'number'),
    (
        'Введите должность руководителя, согласовавшего документ, например "Научный руководитель, приглашенный преподаватель департамента программной инженерии":',
        'supervisor_post'),
    ('Введите ФИО научного руководителя в формате Фамилия И. О.:',
    'supervisor_name'),
    (
        'Введите должность руководителя, утвердившего документ, например "Академический руководитель образовательной программы «Программная инженерия», кандидат технических наук":',
        'akad_post'),
    ('Введите ФИО академического руководителя в формате Фамилия И. О.:',
    'akad_name'),
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

(
 ('Введите ФИО исполнителя в формате Фамилия И. О.:', 'student_name'),
 ('Введите ваш пол в формате "студент" или "студентка":', 'gender'),
 ('Введите номер группы, например "БПИ217":', 'group_number'),
 (
 'Теперь заполнима дату, когда будет подписана пояснительная записка.',
 ('Введите месяц в родительном падеже, например "Мая":', 'month'),
 ('Введите число, например "11":', 'day'),
 (
 'Переходим к заполнению раздела "Введение".',
 ('Введите основания для разработки, это может быть, например, указ ректора или учебный план:', 'grounds_for_development'),
 ('Переходим к разделу "Назначение и область применения".',
 ('Введите функциональное назначение:', 'functional_purpose'),
 ('Введите эксплуатационное назначение:', 'operational_purpose'),
 (
 'Напишите краткую характеристику области применения программы, например "«Генератор документации “Радость Научника”» — прикладная программа, разрабатываемая с целью облегчения формирования и оформления документации."',
 ('Введите краткую характеристику:', 'brief_description'),
 ('Переходим к разделу "Технические характеристики".',
 ('Опишите поставленные задачи на разработку программы:', 'task_statement'),
 ('Опишите архитектуру программы:', 'description_architecture'),
 ('Обоснуйте выбор архитектуры программы:', 'justification_of_architecture'),
 ('Опишите алгоритм работы программы:', 'description_algorithm'),
 ('Обоснуйте выбор алгоритма работы программы:', 'rationale_algorithm'),
 ('Опишите и обоснуйте выбор способа организации входных и выходных данных:', 'input_output_data'),
 ('Опишите состав технических и программных средств:', 'composition_of_technical_means'),
 ('Обоснуйте выбор технических и программных средств:', 'justification_of_technical_means'),
 (
 'Переходим к разделу "Ожидаемые технико-экономические показатели".',
 ('Опишите ориентировочную экономическую эффективность:', 'economic_efficiency'),
 ('Опишите предполагаемую потребность:', 'perceived_need'),
 ('Опишите экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами:', 'advantages_over_analogues'),
 (
 'Переходим к разделу "Список использованных источников".',
 ('По умолчанию в него включены 12 основных ГОСТ-ов, по которым оформляется пояснительная записка. Чтобы добавить свои источники, введите их через перевод строки

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

в одном сообщении\n\nПодсказка: чтобы оформить список литературы по ГОСТу рекомендуем воспользоваться сайтом - <https://open-resource.ru/spisok-literatury/>;',

'sources'),

(

'Наконец заполним одно из приложений в качестве примера.\n\nНе забудьте дозаполнить таблицы, которые генерируются по умолчанию в приложениях. Вы сможете сделать это после сохранения документа в формате .docx.\n\nВведите любой технический термин из текста вашей пояснительной записки:',

'term_1'),

('Введите определение введенного выше термина:', 'definition_1')

]

1.5 Title_page_data.py

```
title_page_questions = [
    (
        '<b>Чтобы в дальнейшем корректно сохранить документ, ответьте на два вопроса:</b>\n\n1) Введите год выполнения работы, например "2023":',
        'year'),
    ('2) Введите ваше ФИО на английском языке в формате "Ivanov_A_S":',
    'second_name'),
    ('Введите полное название ВУЗа, например "Национальный исследовательский университет Высшая Школа Экономики":',
    'university'),
    ('Введите факультет, например "Факультет компьютерных наук":', 'faculty'),
    ('Введите департамент, например "Департамент программной инженерии":',
    'department'),
    ('Введите тип работы, например "Техническое задание":', 'project_type'),
    ('Введите название работы, например «Генератор документации "Радость Научника"»:', 'project_name'),
    (
        'Введите должность руководителя, согласовавшего документ, например "Научный руководитель, приглашенный преподаватель департамента программной инженерии":',
        'supervisor_post'),
    ('Введите ФИО руководителя, согласовавшего документ, в формате Фамилия И. О.:', 'supervisor_name'),
    (
        'Введите должность руководителя, утвердившего документ, например "Академический руководитель образовательной программы «Программная инженерия», кандидат технических наук":',
        'akad_post'),
    ('Введите ФИО руководителя, утвердившего документ, в формате Фамилия И. О.:',
    'akad_name'),
    (
        'Введите номер работы:\n\nПодсказка: в соответствии с ГОСТ 19.103-77 "Обозначения программ и программных документов", номер имеет вид <b>A.B.XXXXX-
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

XX, где: \nA - Код страны; \nB - код организации-разработчика; \nXXXXX - регистрационный номер, который присваивается в соответствии с ОКП; \nXX - Номер редакции (в диапазоне 1-99).\n\nПодробнее можно прочитать по ссылке - <https://www.swrit.ru/doc/espd/19.103-77.pdf>;

'number'),

('Введите ФИО исполнителя в формате Фамилия И. О.:', 'student_name'),

('Введите номер группы, например "БПИ217":', 'group_number')

]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.11-01 12				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

Лист регистрации изменений

[illegible]