

# Разработка библиотеки функций на языке Python, реализующей автоматизированное построение динамических графических пользовательских интерфейсов

Василян Артур Размикович, студент группы РК6-73Б

[var19r219@student.bmstu.ru](mailto:var19r219@student.bmstu.ru)

Научный руководитель: Соколов Александр Павлович

Место проведения: МГТУ им. Н. Э. Баумана

Кафедра: Системы автоматизированного проектирования (РК-6)



Россия, Москва, 2023 г.

# Содержание

- Введение.
- Подходы к разработке пользовательского интерфейса.
- Постановка задачи.
  - Цель и задачи.
  - Базовый принцип генерации GUI.
- Программная реализация.
  - Разработка тестового web-приложения.
  - Особенности генерации GUI на основе aINI.
  - Пример генерации GUI на основе aINI.
- Запуск web-приложения на тестовом сервере.
- Заключение.

# Введение

- **Интерфейс<sup>1</sup>** — это совокупность средств, методов и правил взаимодействия, управления, контроля и т.д. между элементами системы;
- **Пользовательский интерфейс<sup>1</sup>** — это разновидность интерфейсов, в котором одна сторона представлена человеком-пользователем, другая — машиной-устройством.
- **Графический пользовательский интерфейс<sup>1</sup> (GUI)** — это разновидность пользовательского интерфейса, в котором элементы интерфейса, представленные пользователю на дисплее, исполнены в виде графических изображений.

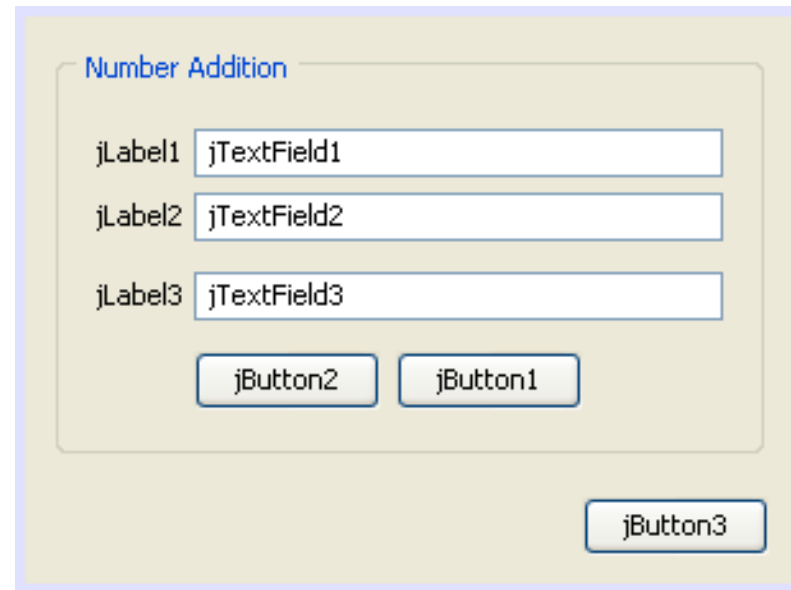


Рис. 1. Пример GUI

# Подходы к разработке пользовательского интерфейса

На основе пользовательских целей<sup>1</sup>.

Методы взаимодействия человека и ЭВМ:

1. **Ограничительный.** Пользователь должен иметь необходимые знания для того, чтобы самому планировать ход выполнения своего задания из предоставляемых ему операций.
2. **Направляющий.** Каждая из целей соответствует определенному пользовательскому заданию, которое может выполнить ЭВМ, взаимодействуя с пользователем.

Методический подход<sup>2</sup>.

- Мониторинг действий оператора.
- Применения типовой системы показателей качества. Обеспечивается оценка качества пользовательского интерфейса.
- Автоматизация программирования и документирования пользовательского интерфейса (возможность автоматизированного документирования интерфейса программы).
- Интерпретация сценария воздействия пользователя в стандартные программные процедуры.

Так же существует метод **построение пользовательского интерфейса с использованием интерактивного машинного обучения**<sup>3</sup>.

---

[1] Санковский Ю.Е. Метод построения оконного интерфейса пользователя на основе моделирования пользовательских целей. 1998

[2] Казаков Г.В., Корянов В.В., Чемирисов В.В., Уваров А.В. Методический подход к созданию универсального пользовательского интерфейса. 2020.

[3] Юркин В.А., Сараджишвили С.Э. Построение пользовательского интерфейса с использованием интерактивного машинного обучения. 2020.

# Постановка задачи

## Цель и задачи

Цель: разработать библиотеку, обеспечивающую автоматизацию построения динамических пользовательских интерфейсов.

## Задачи:

- Рассмотреть существующие подходы разработки GUI.
- Разработать тестовое web-приложение.
- Запустить разработанное web-приложение на тестовом сервере.
- Изучение существующего web-приложения comwps.\*
- Доработка библиотеки русomsdk в части возможности генерации GUI.\*
- Интеграция разработки в состав web-приложения comwps и тестирование.\*

---

[\*] в рамках ВКР

# Постановка задачи

## Базовый принцип генерации GUI

- На рис.2 слева представлены предметно-ориентированные языки (DSL), с помощью которых можно описать элементы интерфейса для дальнейшей генерации.
- В рамках проекта был выбран aINI из-за несложного синтаксиса.

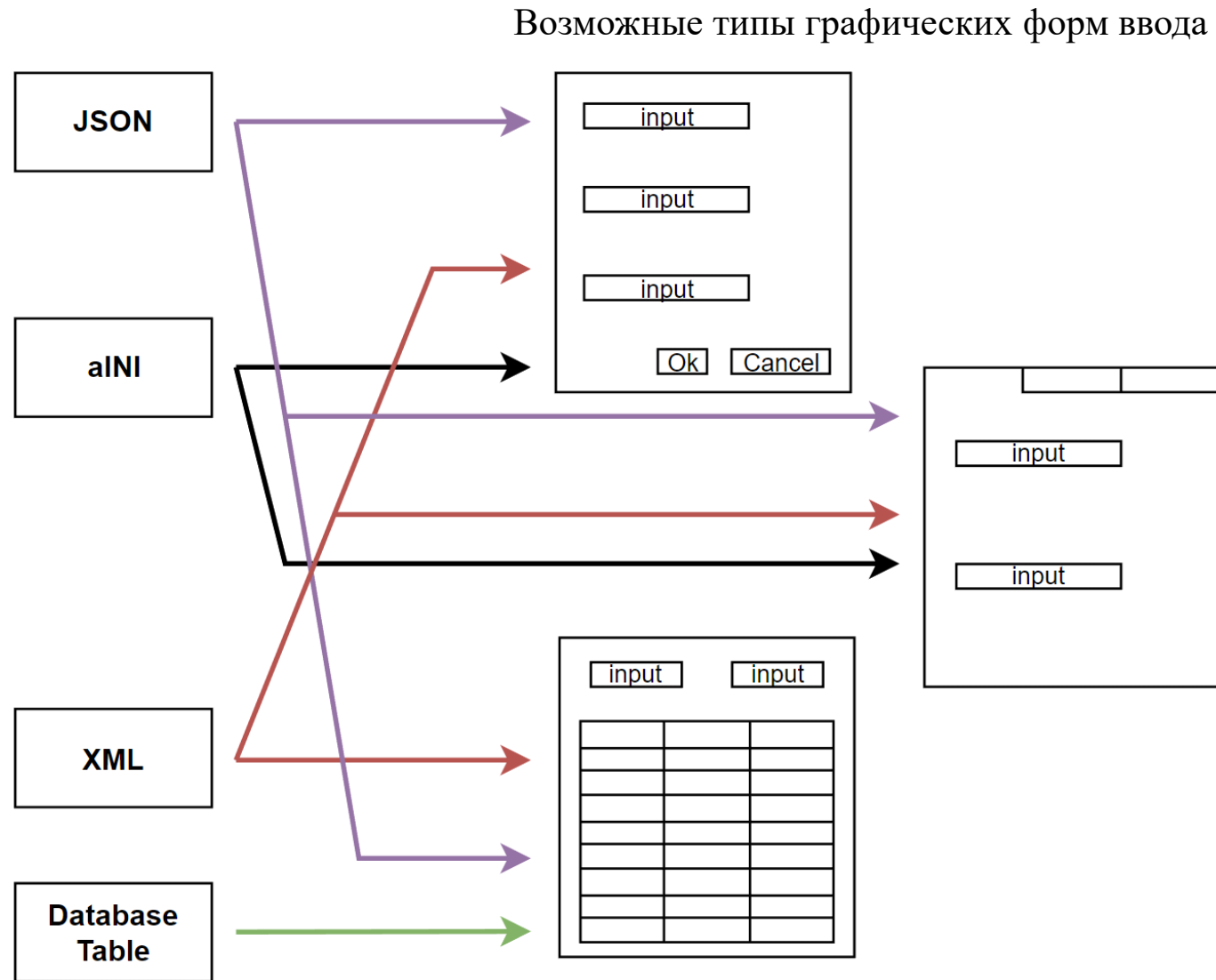


Рис. 2. Схема базового принципа генерации GUI на основе DSL

# Программная реализация

## Разработка тестового web-приложения

- Использовались: Django<sup>1</sup>, Docker<sup>2</sup>, Nginx<sup>3</sup>.
- Контейнеризация — это способ упаковки приложения и всех его зависимостей в один образ, который запускается в изолированной среде, не влияющей на основную операционную систему.
- Docker-образ — шаблон для создания Docker-контейнеров.

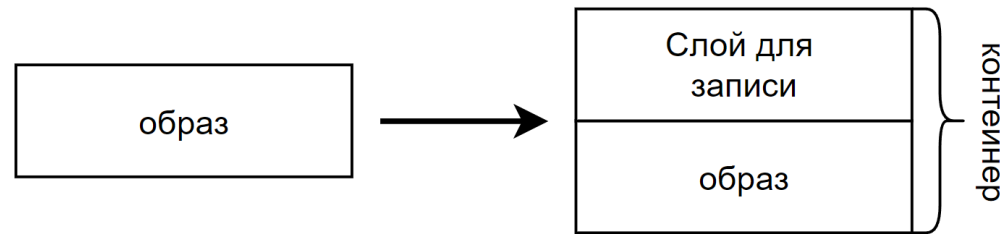


Рис. 2. Создание контейнера на основе образа<sup>4</sup>

---

[1] Django — это Python web-фреймворк, с помощью которого можно вести разработку web-приложения.

[2] Docker — программное обеспечение, применяемое для разработки web-приложений в средах с поддержкой контейнеризации.

[3] Nginx — веб-сервер и почтовый прокси-сервер.

# Программная реализация

## Особенности генерации GUI на основе aINI

- `urlpatterns` — список всех URL, которые обрабатываются web-приложением.
- `gui` — функция-представления, которая сопоставлена с адресом URL с помощью функции `path()` и которая обрабатывает запрос по этому адресу.

```
...
urlpatterns = [
    path("", gui, name="gui"),
    path("image_upload/", image_upload,
name="upload"),
    path("admin/", admin.site.urls),
]
...
```

Листинг 1. Содержимое файла `urls.py`

- В функции `gui` используется HTML файл (`title.html`), который был сгенерирован с помощью преобразователя данных в формате aINI в HTML-код.

```
def gui(request):
    return render(request, "title.html")
```

Листинг 2. Функция-представления `gui` в `views.py`

- В разработке программы для генерации HTML файла на основе aINI использовалась библиотека `ryparsing`<sup>1</sup>.



# Программная реализация

## Пример генерации GUI на основе aINI

```
[sec1]//Пункт 1
X=VX//Параметр X
Y=YX//Параметр Y
box=[1] {0|1}//Флажок 1
box=[1] {0|1}//Флажок 2
[sec1]//Пункт 2
box=[0] {0|1}//Флажок 3
```

Рис. 3. Входные данные в формате aINI

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>title</title>
</head>
<body>
<h1>Пункт 1</h1>
<p><b>Параметр X</b><br>
<input type="text" value="VX"></p>
<p><b>Параметр Y</b><br>
<input type="text" value="YX"></p>
<p><<input type="checkbox" name="Флажок 1" checked>Флажок 1</p>
<p><<input type="checkbox" name="Флажок 2" checked>Флажок 2</p>
<h1>Пункт 2</h1>
<p><<input type="checkbox" name="Флажок 3">Флажок 3</p>
</body>
</html>
```

Рис. 4. Результат работы разработанного преобразователя

# Запуск web-приложения на тестовом сервере

```
avasilyan@sandbox: ~  
Windows PowerShell  
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.  
  
Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)  
  
PS C:\Users\нк> ssh avasilyan@195.19.40.68  
Enter passphrase for key 'C:\Users\нк/.ssh/id_rsa':  
Linux sandbox 5.10.0-15-amd64 #1 SMP Debian 5.10.120-1 (2022-06-09) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sun Dec 11 19:04:02 2022 from 79.139.180.171  
avasilyan@sandbox: $
```

Рис. 5. Вход на сервер

## Пункт 1

Параметр X

VX

Параметр Y

YX

< ☒ Флажок 1

< ☒ Флажок 2

## Пункт 2

< ☐ Флажок 3

Рис. 6. Работа web-приложения на тестовом сервере

# Заключение

- Были рассмотрены существующие подходы разработки GUI;
- была разработана программа для преобразования данных формата aINI в HTML-код;
- в рамках Django было разработано web-приложение, использующее сгенерированный интерфейс;
- автоматизированное построение GUI на основе данных в формате с простым синтаксисом (например, aINI) позволяет создавать графические интерфейсы пользователям без знаний программирования.