

Разработка библиотеки функций на языке Python, реализующей автоматизированное построение динамических графических пользовательских интерфейсов в рамках CMS Django

Подготовил студент: Василян А. Р.

Группа: РК6-73Б

Научный руководитель: Соколов А. П.

Постановка задач

- Рассмотреть существующие подходы разработки GUI;
- Разработать тестовое web-приложение;
- Запустить разработанное web-приложение на тестовом сервере.

Введение

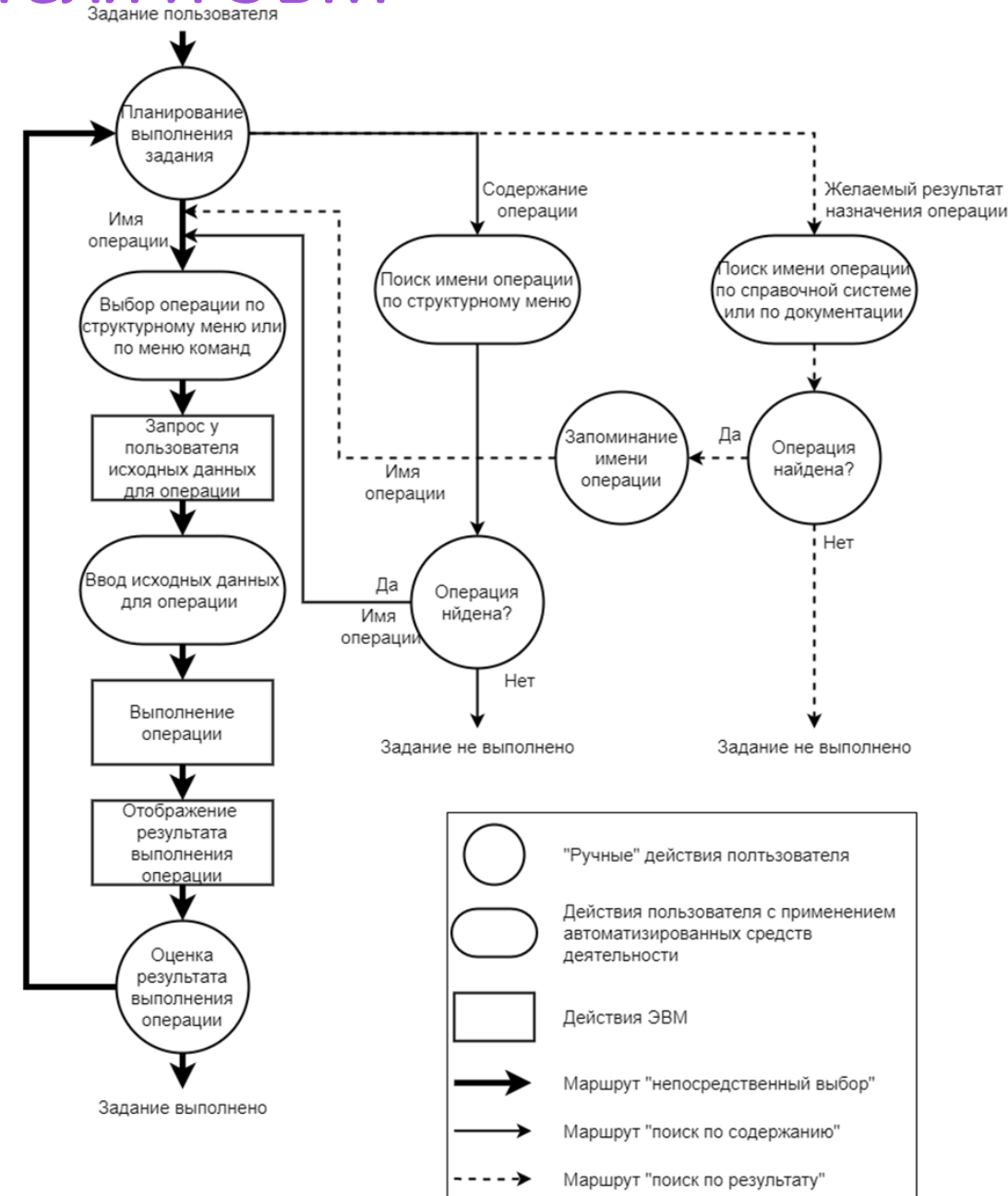
- **Интерфейс** — это совокупность средств методов и правил взаимодействия управления контроля и т.д. между элементами системы;
- **Пользовательский интерфейс** — это разновидность интерфейсов, в котором одна сторона представлена человеком-пользователем, другая — машиной-устройством.
- **Графический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

Методы взаимодействия пользователя и ЭВМ

1. Ограничительный

Пользователю предоставляется набор операций. Операция имеет название, исходные данные и результаты. Пользователь решает, какую из операций необходимо выбрать, и задает для нее исходные данные. После чего ЭВМ выполняет указанную операцию, активируя соответствующие функции приложения, и выдаёт результаты операции пользователю. И этот процесс продолжается до тех пор, пока не будет достигнут желаемый результат.

Следовательно, пользователь должен сам планировать ход выполнения своего задания из предоставляемых ему операций.



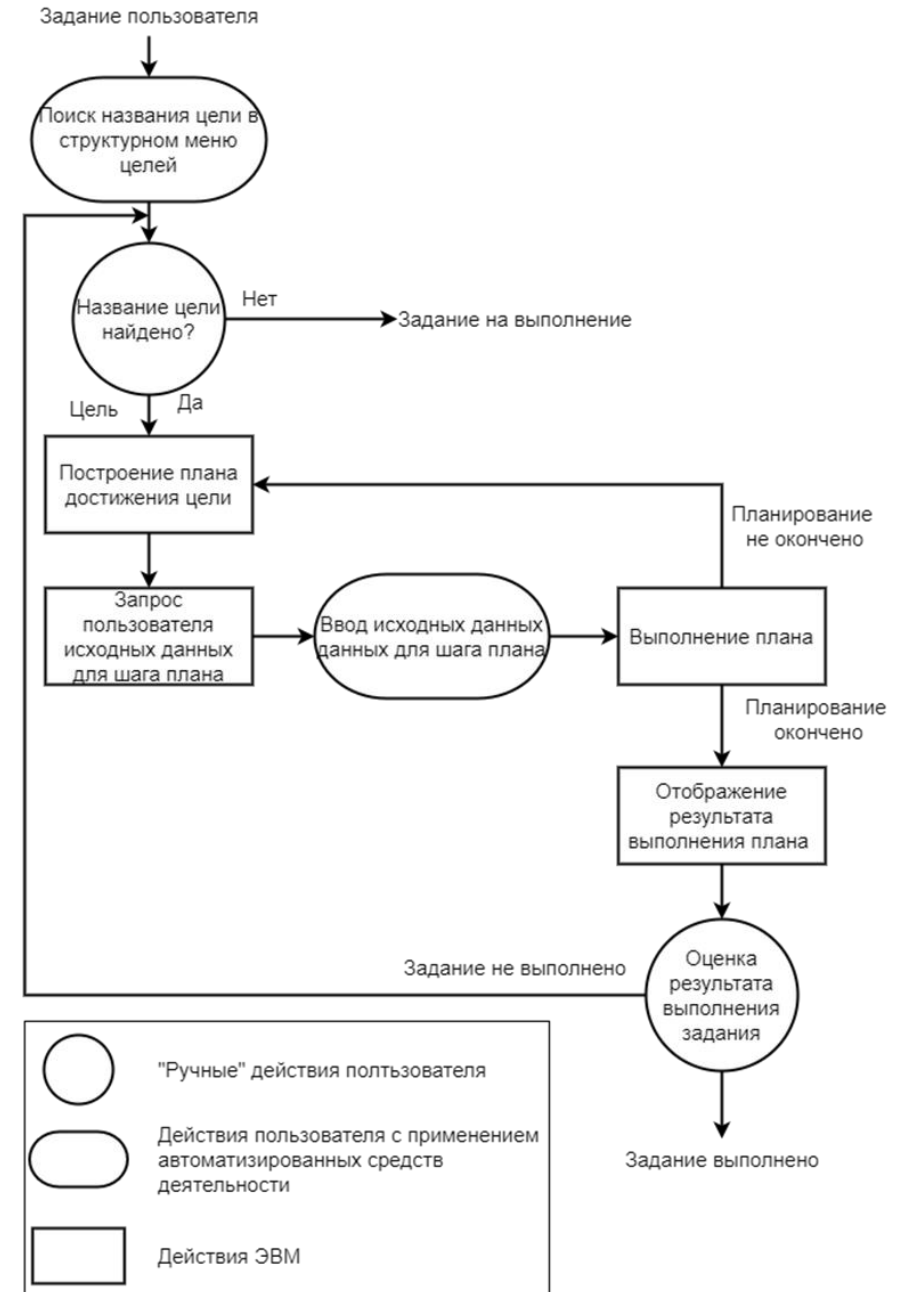
Методы взаимодействия пользователя и ЭВМ

2. Направляющий

Основой направляющего метода является DT-модель (модель диалоговой транзакции).

Направляющий метод взаимодействия “пользователь-ЭВМ” состоит из следующих основных этапов:

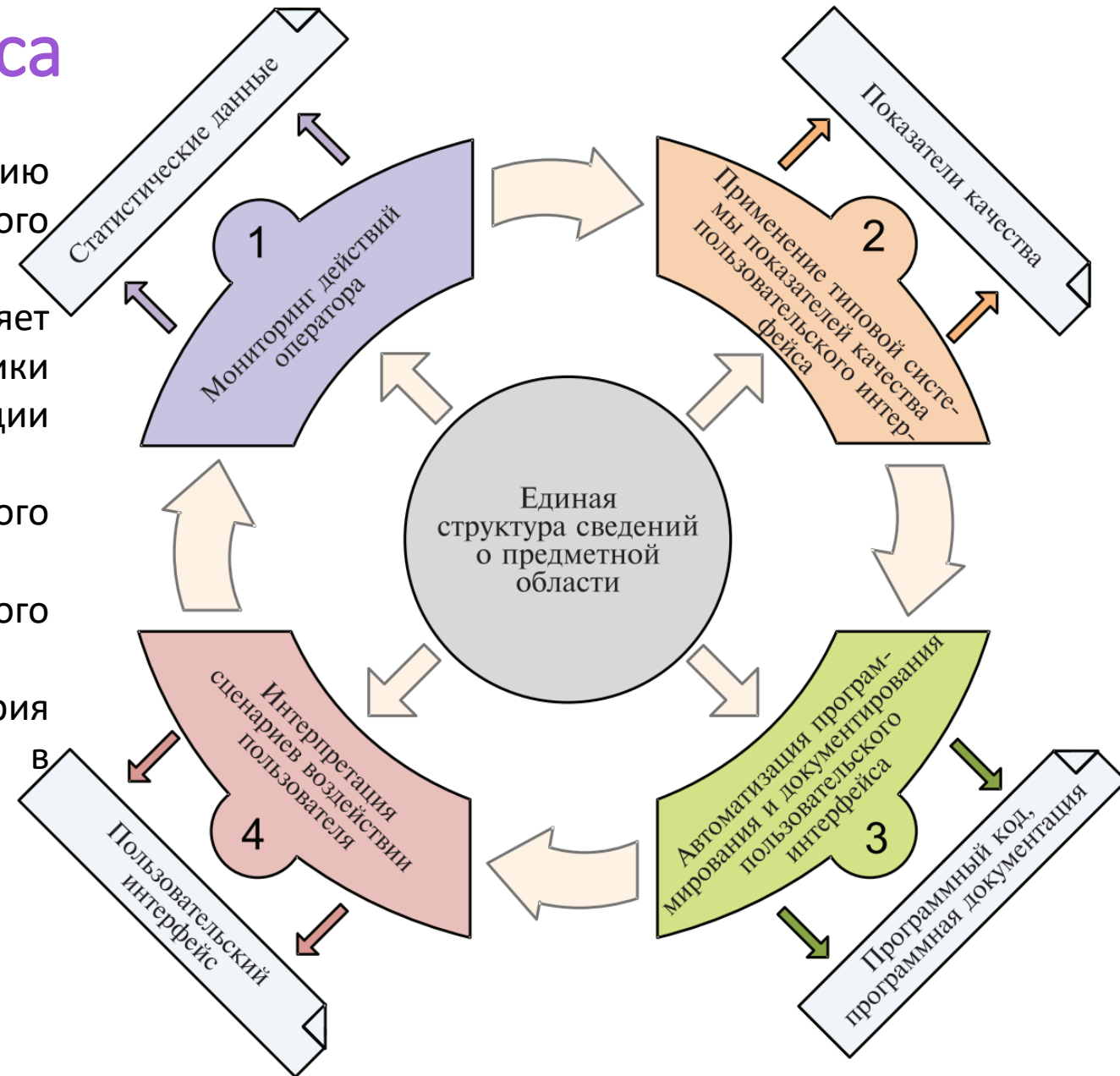
- информирование пользователя о множестве допустимых заданий, которые может выполнять ЭВМ в рамках данного приложения;
- выбор пользователем задания по меню заданий и передача его ЭВМ на выполнение;
- планирование процесса взаимодействия при выполнении задания;
- ввод пользователем данных, необходимых ЭВМ для выполнения задания;
- передача пользователю результатов выполнения задания и их оценка пользователем.



Методический подход к созданию универсального пользовательского интерфейса

Составные элементы подхода к созданию универсального средства построения пользовательского интерфейса программных средств:

1. Мониторинг действий оператора позволяет осуществлять сбор и накопление статистики деятельности оператора во время эксплуатации программных средств;
2. Обеспечивается оценка качества пользовательского интерфейса;
3. Подразумевает возможность автоматизированного документирования интерфейса программы;
4. Отображение некоторого абстрактного сценария осуществляет механизм его интерпретации в стандартные программные процедуры.



Построение пользовательского интерфейса с использованием интерактивного машинного обучения

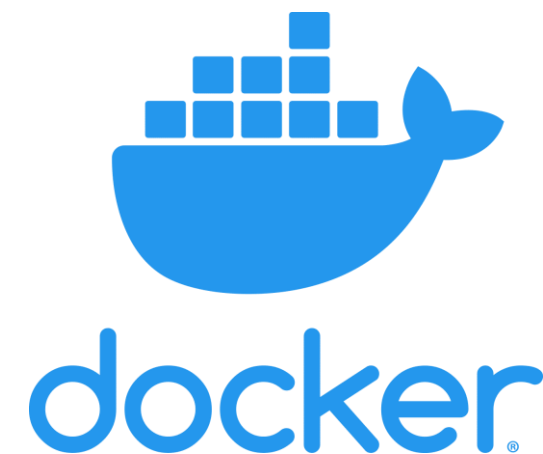
1. Сбор входных данных;
2. На основании собранных данных проводится обучение;
3. По результатам обучения строится последовательность действий для достижения необходимого результата;
4. На основании полученных результатов вносятся корректировки в существующий интерфейс, после чего обучение продолжается.

Разработка тестового web-приложения

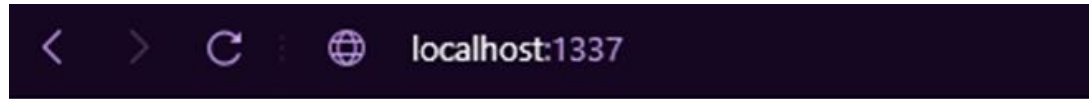
В рамках проекта использовались:

- **Django** — это высокоуровневый Python web-фреймворк для бэкенда, который позволяет быстро создавать безопасные и поддерживаемые web-сайты.
- **Docker** — программное обеспечение, применяемое для разработки, тестирования, доставки и запуска web-приложений в средах с поддержкой контейнеризации.

django



Разработка тестового web-приложения



Выберите файл Файл не выбран submit



Выберите файл Файл не выбран submit

File uploaded at: [/media/intro-1638468413_4A6gK2S.jpg](#)



```
PS D:\PyCharm\django-on-docker\app> docker-compose exec db psql --username=hello_django --dbname=hello_django_dev
psql (13.0)
Type "help" for help.

hello_django_dev=# \l

               List of databases
   Name      | Owner   | Encoding | Collate |  Ctype  |           Access privileges
-----+-----+-----+-----+-----+-----
hello_django_dev | hello_django | UTF8    | en_US.utf8 | en_US.utf8 | 
postgres      | hello_django | UTF8    | en_US.utf8 | en_US.utf8 | 
template0     | hello_django | UTF8    | en_US.utf8 | en_US.utf8 | =c/hello_django
template1     | hello_django | UTF8    | en_US.utf8 | en_US.utf8 | =c/hello_django
(4 rows)
```

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\nk> ssh avasilyan@195.19.40.68  
Enter passphrase for key 'C:\Users\nk/.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:~$
```

< > ↺ ⚠ Не защищено 195.19.40.68:8084

Выберите файл

Файл не выбран

submit

Дальнейшая работа

- Изучение существующего web-приложения comwps (проект по разработке web-клиента для доступа к подсистемам PBC GCD и другим программным системам);
- Доработка библиотеки русomsdk (SDK для программных реализаций сложных вычислительных методов в рамках графоориентированной технологии GBSE) в части возможности генерации GUI;
- Интеграция разработки в состав web-приложения comwps и тестирование работоспособности созданных программных средств.