

## Создание интерфейса для программы python3 с помощью библиотеки eel

*Радионов Сергей Владимирович*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

### Аннотация

В данной статье показано построение интерфейса для программы, написанной на языке программирования python. Интерфейс реализован с помощью библиотеки eel. Для демонстрации построения интерфейса выбран интерфейс приложения, показывающего погоду.

**Ключевые слова:** python, eel, gui, интерфейс, программа.

### Creating an interface for a python3 program using the eel library

*Radionov Sergei Vladimirovich*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

This article shows how to build an interface for a program written in the python programming language. The interface is implemented using the eel library. To demonstrate the construction of the interface, the interface of the application showing the weather was chosen.

**Keywords:** python, eel, gui, interface, program.

При проектировании ПО немаловажную роль имеет пользовательский интерфейс приложения. Его важность заключается в том, что по интерфейсу пользователи, работающие с программой, оценивают ее в целом. Графический интерфейс предоставляет пользователям возможности удобной работы с приложениями, не требуя от них специальных навыков программирования. Казалось бы, что привлекательный интерфейс программы не должен производить сегодня на пользователей весомое впечатление, чтобы они забывали при этом о функционале продукта. Но первая реакция пользователей на продукт, к сожалению, напрямую зависит именно от интерфейса. К слову, интерфейс — механизм, обеспечивающий взаимодействие пользователя с приложением. Именно поэтому рациональная конструкция интерфейса приложения, обеспечивая простоту работы с ним, имеет первостепенное значение в создании приложений. В связи с этим, интерфейс должен, прежде всего, учитывать потребности конечного пользователя приложений. Если целевая аудитория разрабатываемой программы уже известна, то разработка пользовательских интерфейсов не представляет трудностей. Грамотно сконструированный интерфейс упрощает

освоение программы. И, следовательно, неудачный — наоборот снижает эффективность работы с приложениями, вызывает у клиентов разочарование. В итоге пользователи начинают игнорировать его и вообще полностью отказаться от него.

В статье И. Кутепова рассматривается пример построения вычислителя на основе нечеткой логики и применение языка Python для написания интегрированной среды разработки [2]. И.Е. Бронштейн в своей статье описал вывод типов для программного кода на языке Python. Сначала производится обзор описанных в научной литературе алгоритмов вывода типов для языков с параметрическим полиморфизмом. Затем даётся описание нового алгоритма, являющегося модификацией одного из предыдущих: алгоритма декартова произведения. Показывается, как модуль вывода типов, использующий новый алгоритм, анализирует различные конструкции языка Python. Представляются результаты работы над прототипом [3]. В работе В.В. Найденова протестирована производительность пары аналогичных приложений, реализующих CRUD логику с помощью прослойки ORM. Сравнивается SQLAlchemy – де-факто стандартный ORM для Python с динамическим ORM для C++ собственной разработки – YB.ORM. Сравнивается производительность при использовании CPython и PyPy. Проверяется влияние отключения логов на производительность [4]. В статье И.Е. Бронштейн рассматриваются виды дефектов, которые обычно встречаются в программном коде на языке Python. Показывается, что возможные дефекты для Python не похожи на те, что часто встречаются в коде на Си/Си++ и, следовательно, необходимо исследование дефектов в крупных проектах с открытым исходным кодом. Даётся классификация найденных дефектов на основе того, нужен ли для нахождения ошибки вывод типов. Показывается, что существует небольшая доля "простых" дефектов, но для обнаружения большинства дефектов вывод типов необходим. Рассматривается вопрос, какие конструкции языка Python должны поддерживаться при выводе типов для нахождения реальных дефектов [5]. В работе Д.А. Кузнецов рассматривается структура интерфейса программы «Фармацевтическая экономическая безопасность» для фармацевтических организаций. Описывается функциональное предназначение и возможности пунктов основного меню прикладной программы [6]. Ю.А. Котов, А.В. Шаповалов в своей статье рассмотрели интерфейс программной реализации экспертной системы для восстановления простой замены букв текста. Описаны базовые элементы интерфейса, включающие выбор функциональных методов и базовых операций (замена, сдвиг, перебор). Не менее значимы иностранные исследования в данной сфере [8-9].

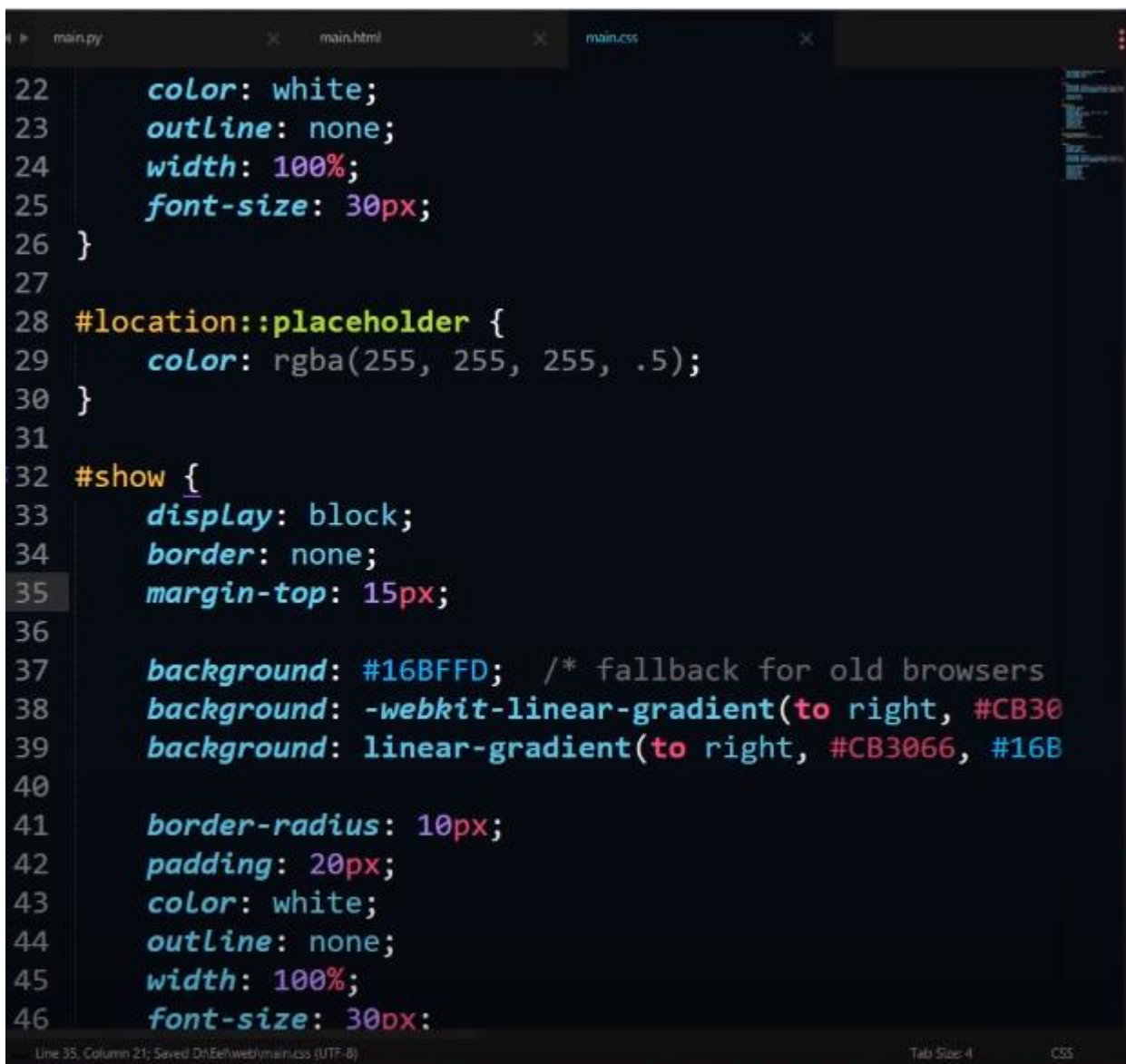
Разработку приложения следует начать с загрузки необходимых библиотек командой “pip install ruowm eel”. В папке проекта создаем папку web в ней создаем main.html, где описываем интерфейс в html верстке (Рис.1).

```
main.py  main.html  main.css
7      <script src="eel.js"></script>
8      <link rel="icon" type="image/png" href="/
      favicon.png">
9
10     <link rel="stylesheet" href="main.css">
11     <link href="https://fonts.googleapis.com/
      css2?family=Roboto:wght@300;400;500&display=swap"
      rel="stylesheet">
12 </head>
13 <body>
14
15     <input id="location" type="text" placeholder="
      Введите название страны и города ..." required=""
      value="Нью-Йорк, США">
16
17     <button id="show">Узнать погоду</button>
18
19     <div class="info">
20
21     </div>
22
23 </body>
24 </html>
```

Line 17, Column 36 Saved D:\Echweb\main.html (UTF-8) Tab Size: 4 HTML

Рис.1. Html файл

Далее необходимо придать интерфейсу красивый вид с помощью CSS. Описывает все необходимое в файле style.css (Рис.2).



```
22     color: white;
23     outline: none;
24     width: 100%;
25     font-size: 30px;
26 }
27
28 #location::placeholder {
29     color: rgba(255, 255, 255, .5);
30 }
31
32 #show {
33     display: block;
34     border: none;
35     margin-top: 15px;
36
37     background: #16BFFD; /* fallback for old browsers
38     background: -webkit-linear-gradient(to right, #CB30
39     background: linear-gradient(to right, #CB3066, #16B
40
41     border-radius: 10px;
42     padding: 20px;
43     color: white;
44     outline: none;
45     width: 100%;
46     font-size: 30px;
```

Рис.2. Содержание файла CSS

Теперь необходимо написать логику программы на языке python. Для этого создадим скрипт main.py, и напишем функцию получения температуры по заданному месту (Рис.3).

```
main.py main.html x
1 import eel
2 import pyowm
3
4 owm = pyowm.OWM("6d00d1d4e704068d70191bad2673e0cc")
5
6 @eel.expose
7 def get_weather(place):
8     mgr = owm.weather_manager()
9
10    observation = mgr.weather_at_place(city)
11    w = observation.weather
12
13    temp = w.temperature('celsius')['temp']
14
15    return "В городе " + city + " сейчас " + str(temp) + " градусов!"
16
17
18 #eel.init("web")
19 #eel.start("main.html", size=(700, 700))
```

Рис.3. Скрипт python

Для того чтобы вызвать метод `get_weather` из интерфейса необходимо в html файле написать обработчик нажатия кнопки (Рис.4).

```
<button id="show">Узнать погоду</button>
<div id="info"></div>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script type="text/javascript">
    // тут пишем JS код для вызова Python функций
    async function display_weather() {
        let place = document.getElementById('location').value;
        // вызываем функцию get_weather из Python кода
        let res = await eel.get_weather(place());
        document.getElementById('info').innerHTML = res;
    }
    jQuery('#show').on('click', function() {
        // при клике на кнопку, отображаем погоду
        display_weather();
    });
</script>
```

Рис.4. Обработчик нажатия кнопки

В результате получилось простое и красивое приложение отображения температуры по заданному месту (Рис.5).

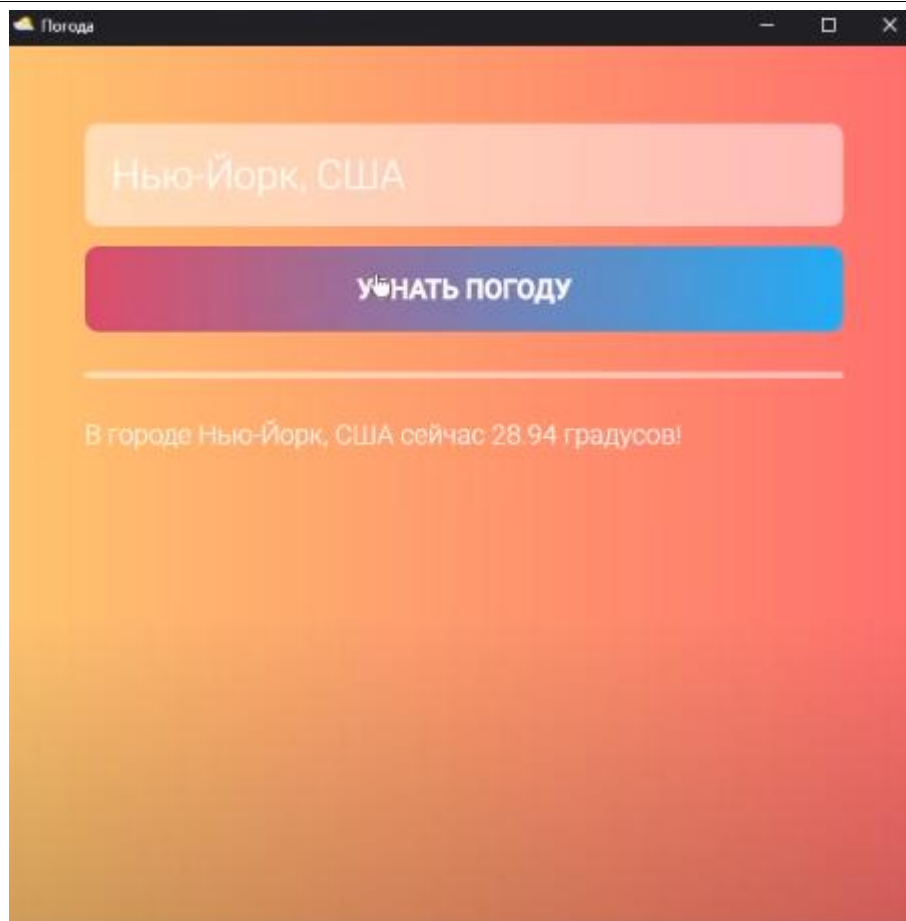


Рис.5. Внешний вид приложения

Таким образом, было рассмотрено создание интерфейса с помощью библиотеки eel языка программирования python3.

### **Библиографический список**

1. Лутц М. Изучаем python. М., 2009.
2. Кутепов И. Применение языка python при проектировании нечеткого контроллера // Компоненты и технологии. 2013. № 8 (145). С. 148-154.
3. Бронштейн И.Е. Вывод типов для языка python // Труды Института системного программирования РАН. 2013. Т. 24. С. 161-190.
4. Найденов В.В. Тестирование производительности orm в языках python и c++ // RSDN Magazine. 2014. № 1. С. 05-08.
5. Бронштейн И.Е. Исследование дефектов в коде программ на языке python // Программирование. 2013. Т. 39. № 6. С. 25-32.
6. Кузнецов Д.А. Интерфейс программы "фармацевтическая экономическая безопасность" // Российский медико-биологический вестник им. академика И.П. Павлова. 2009. № 2. С. 162-165.
7. Котов Ю.А., Шаповалов А.В. Интерфейс программы восстановления простой замены букв текста // Современные тенденции развития науки и технологий. 2016. № 4-4. С. 57-59.
8. Smith A. W. et al. Application program interface that enables communication

for a network software platform : пат. 7117504 США. 2006.

9. Parikh V., Moore R., Cheng H. Application program interface for a graphics system : пат. 6456290 США. 2002.