

Конвертер валют

Автор: Киселев Тимофей Денисович

Проект представляет собой настольное приложение для конвертации валют на основе актуальных (или последних доступных) курсов Центрального банка Российской Федерации. Приложение позволяет не только мгновенно переводить суммы между любыми парами поддерживаемых валют, но и просматривать историю курсов по конкретной валюте, а также экспортировать полный список курсов в формат Excel (.xlsx). Особое внимание уделено удобству пользователя: реализованы автодополнение кодов валют с отображением их полных названий, поддержка арифметических выражений во вводе суммы, сохранение последних настроек между запусками и работа в условиях отсутствия интернета за счёт локального кэширования данных.

Приложение состоит из трёх основных окон: главное окно конвертера (класс `kalc`), окно со списком всех курсов (`Wn2`) и окно просмотра истории курсов по валюте (`Wn3`). В главном окне пользователь вводит сумму (возможно, в виде арифметического выражения), выбирает исходную и целевую валюту и получает результат конвертации. Окно `Wn2` отображает таблицу всех доступных валют с курсами в рублях и позволяет экспортировать данные в Excel. Окно `Wn3` предоставляет историю изменения курса выбранной валюты с использованием кастомного виджета `wd1` для наглядного отображения даты и значения.

Ключевые технические решения включают безопасное вычисление арифметических выражений с помощью модуля `ast` (функция `safe_eval`), что предотвращает выполнение произвольного кода при вводе. Все курсы валют сохраняются в локальной SQLite-базе данных, что позволяет приложению работать автономно — при отсутствии интернета загружаются последние сохранённые курсы. Для улучшения пользовательского опыта реализовано автодополнение с отображением полных названий валют на русском языке. Весь интерфейс оформлен в едином стиле через

строку CSS-подобных правил, обеспечивающих визуальную целостность и приятную «зелёную» тематику.

Приложение написано на Python с использованием фреймворка PyQt6. Необходимые сторонние библиотеки: requests (для загрузки курсов с сайта ЦБ РФ), xlswriter (для экспорта в Excel) и sqlite3 (входит в стандартную библиотеку Python). Для запуска достаточно установить зависимости командой: `pip install PyQt6 requests xlswriter`.

Интерфейс приложения включает три ключевых экрана: главное окно конвертера с полями ввода и кнопкой обмена валютами, таблицу всех курсов с возможностью экспорта и окно истории курсов с поиском по коду валюты. Проект ориентирован на пользователей, которым нужен надёжный, автономный и удобный инструмент для работы с валютными курсами без зависимости от онлайн-сервисов.

```
def xlsxxx(self, arg="Curss", path="/Users/denis/Downloads/"):
    os.makedirs(path, exist_ok=True)
    workbook = xlswriter.Workbook(f"{path}{arg}.xlsx")
    worksheet = workbook.add_worksheet()
    worksheet.write(0, 0, "Валюта")
    worksheet.write(0, 1, "Курс (RUB)")
    keys = sorted(valcurss.keys())
    for row, key in enumerate(keys, start=1):
        worksheet.write(row, 0, key)
        worksheet.write(row, 1, valcurss[key])
    workbook.close()

def show_save_file_dialog(self):
    url, _ = QFileDialog.getSaveFileName(
        parent=self,
        caption="Сохранить файл",
        filter="Excel Files (*.xlsx)",
        initialFilter="Excel Files (*.xlsx)"
    )
    if url.isEmpty():
        return
    file_path = url.toLocalFile()
    if not file_path.endswith('.xlsx'):
        file_path += '.xlsx'
    dir_path = os.path.dirname(file_path) + os.sep
    base_name = os.path.splitext(os.path.basename(file_path))[0]
    self.xlsxxx(arg=base_name, path=dir_path)
```

```

def get_currency_rates():
    url = "https://www.cbr.ru/scripts/XML_daily.asp"
    response = requests.get(url)
    response.raise_for_status()

    root = ET.fromstring(response.content)
    currency_rates = {}

    conn = sqlite3.connect('curs_database.db')
    cursor = conn.cursor()
    current_time = datetime.now().strftime("%Y-%m-%d %H:%M")

    for valute in root.findall('Valute'):
        char_code = valute.find('CharCode').text
        nominal = int(valute.find('Nominal').text)
        value = float(valute.find('Value').text.replace(',', '.'))
        rate = value / nominal
        currency_rates[char_code] = rate
        cursor.execute("INSERT INTO curss (title, curs, date) VALUES (?, ?, ?)",
            | | | | (char_code, rate, current_time))

    currency_rates["RUB"] = 1.0
    conn.commit()
    conn.close()
    return currency_rates

```