

## ПРИМЕНЕНИЕ ЯЗЫКА PYTHON ДЛЯ РАЗРАБОТКИ КОНВЕРТЕРА ВАЛЮТ

Ошурков А.Ю., Бужинская Н.В.

*Нижегородский государственный социально-педагогический институт (филиал)  
ФГАОУ ВО «Российский государственный профессионально-педагогический университет»,  
Ниžний Тагил, e-mail: nadezhda\_v\_a@mail.ru*

В связи с усложнением функций программного обеспечения наблюдается интерес к различным языкам программирования. В настоящее время одним из самых популярных языков программирования является Python. Python – это высокоуровневый язык программирования, который может использоваться для решения задач математики, создания программ для распознавания образов, построений нейронных сетей и др. В статье демонстрируется применение данного языка для создания конвертера валют. С помощью данной программы можно ознакомиться с курсами трех мировых валют, производить конвертирования из RUS в USD, EUR и CNY. В статье расписаны этапы создания конвертера на языке Python. На этапе проектирования определяются требования к данной программе, определяется набор библиотек и модулей, которые будут применяться в процессе работы. Разработка программы включает несколько этапов. Вначале создается интерфейс посредством PyQt5, затем данный интерфейс компилируется в файл с расширением .py. На втором этапе оформляется код для каждой функции, которые будут вызваны при нажатии на соответствующие кнопки. Актуальный на сегодняшний день курс валют можно получить из Интернета. Для более удобной работы создана база данных, которая хранит информацию о результатах конвертирования. Заключительным этапом работы является тестирование, в процессе которого были выявлены и исправлены ошибки.

**Ключевые слова:** программирование, программный продукт, конвертер, база данных, валюта, функция

## USING PYTHON TO DEVELOP A CURRENCY CONVERTER

Oshurkov A.Y., Buzhinskaya N.V.

*Nizhny Tagil State Socio-Pedagogical Institute (branch) of Federal State Autonomous educational institution «Russian state vocational pedagogical University», Nizhny Tagil, e-mail: nadezhda\_v\_a@mail.ru*

Due to the complexity of software functions, there is an interest in various programming languages. Currently, one of the most popular programming languages is Python. Python is a high-level programming language that can be used for solving math problems, creating programs for pattern recognition, building neural networks, and so on. This article demonstrates how to use this language to create a currency Converter. With this program, you can familiarize yourself with the exchange rates of three world currencies, make conversions from RUS to USD, EUR and CNY. This article describes the steps for creating a Python Converter. At the design stage, the requirements for this program are determined, and the set of libraries and modules that will be used in the process is determined. The program development includes several stages. First, an interface is created using PyQt5, then this interface is recompiled to a file with the .py extension. At the second stage, the code is generated for each function that will be called when the corresponding buttons are clicked. The current exchange rate can be obtained from the Internet. For more convenient operation, a database has been created that stores information about the conversion results. The final stage of the work is testing, during which errors were identified and corrected.

**Keywords:** programming, software product, Converter, database, currency, function

На современном этапе развития общества большой популярностью среди учащихся пользуются различные языки программирования, а также программные продукты, разработанные с их помощью. Одним из самых популярных языков программирования является Python. Python представляет собой высокоуровневый язык программирования, который поддерживает такие парадигмы, как императивное программирование, объектно-ориентированное, функциональное и др. [1]. Целью данного исследования является описание возможностей языка программирования Python как средства разработки программы с пользовательским интерфейсом для конвертации валют с возможностью сохранения истории действий в базе данных SQLite. Выбор данного языка для разработ-

ки нашего программного продукта объясняется не только его популярностью. В настоящее время Python активно развивается, появляются его новые версии, улучшаются и добавляются библиотеки и модули [2].

### Материалы и методы исследования

В данном исследовании описан процесс реализации программы «Конвертер валют» на языке Python. Тема для разработки была выбрана таким образом, чтобы посмотреть возможности Python с разных точек зрения – работа с базой данных, с информацией в Интернете, создание дизайна.

Процесс работы включал несколько этапов:

– проектирование программного продукта, во время которого были обозначены основные направления работы;

- разработка дизайна конвертера;
- оформление внутренней логики программы;
- обработка событий;
- создание базы данных для отображения курса валют;
- подготовка файла с расширением *.exe*;
- оформление необходимой документации;
- тестирование программы.

### Результаты исследования и их обсуждение

На этапе проектирования были определены основные требования к программе (рис. 1) [3].

Пользователь вводит необходимую сумму. Если он хочет увидеть курс, актуальный на сегодняшний день, он нажимает кнопку «Получить курс», иначе он выбирает команду «Конвертировать» и получает инте-

ресующие его значения. Далее посмотрим процесс разработки программы на языке *Python*.

Структура программы представлена на рис. 2.

В папке *Images* хранится иконка приложения. Сама база данных *db.currency\_convert.db* хранится в общей папке с проектом, *mainWindow\_converter.ui* интерфейс программы в *XVFL*, который конвертирован в файл *mainWindow\_converter.py*, файл *parse\_currency.py* содержит код для парсинга с официального сайта Google, чтобы собирать актуальную информацию о курсе. В *main.py* прописана основная логика сайта.

Запустим файл *main.py*. Появится окно программы. При вводе необходимой суммы в рублях программа переводит ее в другую денежную единицу по курсу на сегодняшний день (рис. 3).

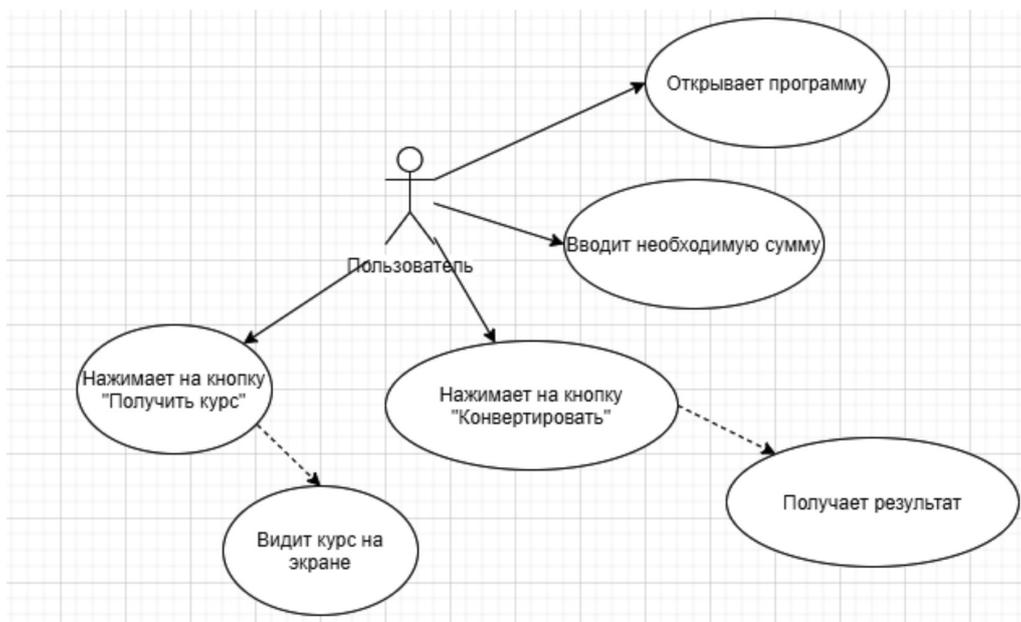


Рис. 1. Диаграмма прецедентов

Images	25.06.2020 14:10	Папка с файлами	
db.currency_convert	25.06.2020 17:31	Data Base File	12 КБ
main	25.06.2020 17:32	Python File	4 КБ
mainWindow_converter	25.06.2020 15:40	Python File	5 КБ
mainWindow_converter.ui	25.06.2020 15:34	Файл "UI"	5 КБ
parse_currency	25.06.2020 16:15	Python File	1 КБ

Рис. 2. Структура программы

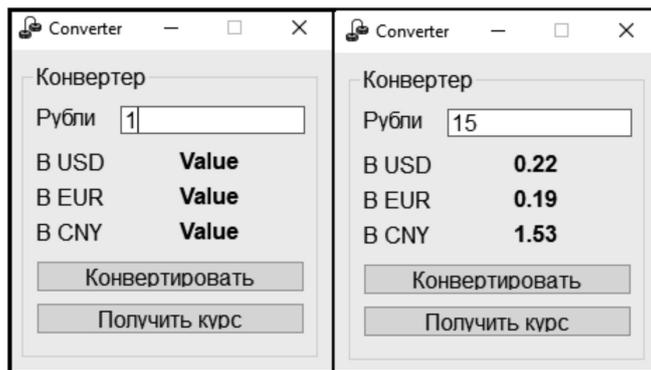


Рис. 3. Интерфейс программы

```

from mainWindow_converter import window_converter
from parse_currency import check_current
from PyQt5 import QtCore, QtGui, QtWidgets
import sys
from typing import *
from datetime import datetime
import sqlite3 as sql
from sqlite3 import Error
    
```

Рис. 4. Список подключаемых библиотек и модулей для реализации программы

Список используемых библиотек и модулей представлен на рис. 4.

Интерфейс программы создавался в *PyQt5*, с помощью *Qt Designer*. Для создания интерфейса использовались такие элементы, как *groupBox*, семь виджетов *label*, *lineEdit*, два виджета *pushButton*.

*SQLite* предназначена для работы с базой данных, которая отвечает за создание и поддержку базы данных для данного приложения. Библиотека *sys* предоставляет программисту набор функций, которые предоставляют информацию о том, как работает интерпретатор Python, взаимодействуя с операционной системой. Библиотека *typing* предназначена для создания аннотаций. Модель *datetime* предоставляет классы для обработки времени и даты разными способами. В *parse\_currency* находится функция *check\_currency* для получения информации с сайта Google.

На рисунке ниже изображена база данных *db.currency\_convert.db*, которая хранит информацию о дате и времени использования конвертера, сумму в рублях для конвертирования, курс доллара, евро и конвертированные значения по курсам (рис. 5).

Далее осуществлялось подключение к базе данных (рис. 6). Если подключение

завершилось с ошибкой, программа сигнализирует об этом с помощью исключения *Try Except*.

Затем выполняется запрос, согласно которому о переменной *currency* присваивается кортеж последней записи в базе данных. Переменные получают значения актуального на сегодняшний день курса валют (рис. 7).

На рис. 8 продемонстрировано, что функция *toFixed* на входе получает число типа *float* (переменная *numObj*) и количество знаков после запятой (переменная *digits*). На выходе получаем число типа *str* с количеством знаков после запятой, которое было указано.

В переменную *data* присваивается кортеж с данными, которые нужно внести в базу данных (рис. 9).

На рис. 10 продемонстрирован пример запроса на добавление данных из переменной *data* в базу данных.

Затем функция на вход получает ссылку на источник с актуальным на сегодняшний день курсом, затем отправляет запрос, получает HTML-файл, обрабатывает его и возвращает курс валюты. Реализация функции на языке Python представлена на рис. 11.

	id	time_use	input_currency	USD	EUR	CNY	output_USD	output_EUR	output_CNY
1	1	25/06/20 17:11:27	150	69.45	77.74	9.81	2.16	1.93	15.29
2	2	25/06/20 17:11:31	150000	69.45	77.74	9.81	2159.83	1929.51	15290.52
3	3	25/06/20 17:11:35	250000	69.45	77.74	9.81	3599.71	3215.85	25484.2
4	4	25/06/20 17:31:38	1	69.45	77.74	9.81	0.01	0.01	0.1
5	5	25/06/20 17:31:47	10	69.45	77.74	9.81	0.14	0.13	1.02
6	6	25/06/20 23:51:59	65	69.14	77.55	9.77	0.94	0.84	6.65
7	7	25/06/20 23:52:04	650	69.14	77.55	9.77	9.4	8.38	66.53
8	8	25/06/20 23:52:09	1	69.14	77.55	9.77	0.01	0.01	0.1
9	9	25/06/20 23:52:17	4000	69.14	77.55	9.77	57.85	51.58	409.42

Рис. 5. База данных db.currency\_convert.db

```
def create_connection(database: str) -> str
:   try:
        database = sql.connect(database)
        print("Connection database")
        return database
    except Error:
        print(f"Error '{Error}' connection")
```

Рис. 6. Подключение к базе данных

```
[currencys] = sqlCursor.execute("select id, USD, EUR, CNY "
                                "from history_converter_currency "
                                "order by id desc limit 1")
value_USD = currencys[-3]
value_EUR = currencys[-2]
value_CHY = currencys[-1]
```

Рис. 7. Запрос к базе данных

```
def toFixed(numObj: float, digits: int = 0) -> str:
    return f"{numObj:.{digits}f}"
```

Рис. 8. Реализация функции toFixed

```
data = [(str(datetime.strftime(datetime.now(), "%d/%m/%y %H:%M:%S")),
        input_value_RUS, value_USD,
        value_EUR, value_CHY,
        output_value_USD, output_value_EUR,
        output_value_CNY)]
```

Рис. 9. Работа с переменной data

```
sqlCursor.executemany("INSERT INTO history_converter_currency(time_use,"
                      "input_currency,"
                      "USD,"
                      "EUR,"
                      "CNY,"
                      "output_USD,"
                      "output_EUR,"
                      "output_CNY)"
                      "VALUES (?, ?, ?, ?, ?, ?, ?, ?)", data)
database.commit()
```

Рис. 10. Выполнение запроса

```
def check_current(url):
    response = rec.get(url=url, headers=HEADERS)
    HTML = response.content

    convert_html = BeautifulSoup(HTML, 'lxml')
    head = convert_html.find_all('span', {'class': 'DFlfde',
                                         'class': 'SwHCTb',
                                         'data-precision': 2})

    value = head[0].text
    value = value.replace(',', '.')
    return value
```

Рис. 11. Реализация функции

```
for i in range(len(input_value_RUS.lower())):
    if not input_value_RUS.isdigit():
        window.output_USD_2.setText("Error 01")
        window.output_EUR_2.setText("Error 01")
        window.output_CNY_2.setText("Error 01")
    return
```

Рис. 12. Проверка введенных пользователем данных

Таблица 1

Результаты тестирования программы. Тест-кейс № 1. Корректный

Номер	1
Заголовок	Перевод нужной суммы в рублях в другую денежную единицу
Предусловие	Программа запущена. Открыто главное окно программы. На ПК есть доступ в Интернет
Шаг	<b>Ожидаемый результат</b>
Ввести нужную сумму в поле «Рубли». Вводить только цифры	Ввод успешен
Нажать на кнопку «Конвертировать»	Программа выдает необходимый результат
Нажать на кнопку «Получить курс»	Программа выдает курс, актуальный на данном этапе

Таблица 2

Результаты тестирования программы. Тест-кейс № 2. Некорректный

Номер	2
Заголовок	Перевод нужной суммы в рублях в другую денежную единицу
Предусловие	Программа запущена. Открыто главное окно программы. На ПК есть доступ в Интернет
Шаг	<b>Ожидаемый результат</b>
Ввести нужную сумму в поле «Рубли». Вводить любую последовательность символов	Программа выдает сообщение об ошибке. Кнопки «Конвертировать» и «Получить курс» не активны

Кроме того, для данной функции была написана проверка, чтобы пользователь вводил только числовые значения. Если пользователь введет символ, отличный от числового значения, программа выдает ошибку. Реализация функции представлена на рис. 12.

На заключительном этапе работы было оформлено «Руководство пользователя»

и проведено тестирование программы. Тестирование направлено на решение двух важнейших задач. С одной стороны, разработчику нужно показать, что программа удовлетворяет заданным требованиям к ней, а с другой – исправить все ошибки, которые были обнаружены во время внедрения программы [4; 5]. Результаты тестирования программы представлены в табл. 1–2 [6].

### Заключение

Данная программа может использоваться в качестве демонстрационной для объяснения обучаемым основных возможностей языка Python, а именно:

- установка различных библиотек через командную строку;
- создание базы данных и запросов к ней с помощью SQLite;
- получение нужной информации с различных сайтов;
- разработка интерфейса с помощью PyQt;
- перекомпилирование файла с расширением .ui в расширение .py.

В дальнейшем планируется добавить в данную программу визуальное отобра-

жение курсов валют с помощью библиотеки Matplotlib.

### Список литературы

1. Свейгарт Эл. Учим Python, делая крутые игры. М.: Бомбора, 2018. 416 с.
2. Прохоренок Н., Дронов В. Python 3 и PyQt5. Разработка приложений. СПб.: БХВ-Петербург, 2019. 833 с.
3. Каюмова А.В. Визуальное моделирование систем в StarUML: учебное пособие. Казань: Казанский федеральный университет, 2013. 104 с.
4. Липаев В.В. Программная инженерия сложных заказных программных продуктов: учебное пособие. М.: МАКС ПРЕСС, 2014. 312 с.
5. Карпович Е.Е. Методы тестирования и отладки программного обеспечения. М.: Изд. Дом НИТУ «МИСиС», 2020. 136 с.
6. Правильно пишем тест-кейсы. Памятка начинающему специалисту по тестированию. [Электронный ресурс]. URL: <https://victorz.ru/202001101079> (дата обращения: 07.07.2020).