

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”

Факультет	Программной Инженерии и Компьютерной Техники
Направление подготовки (специальность)	Нейротехнологии и программирование
Дисциплина	Программирование на Python

Лабораторная работа 9
ОТЧЕТ

Выполнил студент:	Иголкин Владислав Андреевич (504623)
Группа:	P3124
Преподаватель:	Жуков Николай Николаевич (261087)

г. Санкт-Петербург

2025 г.

1. Цель работы:

- Реализовать **CRUD** (Create, Read, Update, Delete) для сущностей бизнес-логики приложения.
- Освоить работу с **SQLite** в памяти (:memory:) через модуль sqlite3.
- Понять принципы **первичных и внешних ключей** и их роль в связях между таблицами.
- Выделить **контроллеры** для работы с БД и для рендеринга страниц в отдельные модули.
- Использовать архитектуру **MVC** и соблюдать разделение ответственности.
- Отображать пользователям таблицу с валютами, на которые они подписаны.
- Реализовать полноценный **роутер**, который обрабатывает GET-запросы и выполняет сохранение/обновление данных и рендеринг страниц.
- Научиться тестировать функционал на примере сущностей currency и user с использованием **unittest.mock**.

2. Описание моделей, их свойств и связей

Модель Currency:

Поле	Тип	Описание
id	INTEGER	Уникальный идентификатор
name	TEXT	Название валюты
code	TEXT	Символьный код валюты (USD, EUR, GBP...)
rate	REAL	Курс валюты относительно базовой
last_update	TEXT	Дата и время последнего обновления курса

Связей с другими таблицами нет, так как приложение содержит только одну сущность валют.

2. Описание моделей, их свойств и связей

Модели:

Currency
Поле Тип Описание
id INTEGER Уникальный идентификатор
num_code TEXT Цифровой код валюты
char_code TEXT Символьный код валюты (USD...)
name TEXT Название валюты
value FLOAT Курс валюты
nominal INTEGER Номинал валюты
User
Поле Тип Описание
id INTEGER Уникальный идентификатор
name TEXT Имя пользователя

Связь между моделями: user_currency — связь «многие ко многим» между пользователями и валютами.

3. Структура проекта с назначением файлов

```
project/
  └── myapp.py          # Основной файл приложения, маршруты, запуск сервера
  └── models/
```

```

    └── __init__.py
    ├── currency.py          # Модель Currency
    ├── user.py               # Модель User
    ├── author.py             # Модель Author
    └── controllers/
        └── __init__.py
            ├── databasecontroller.py   # CRUD для Currency и User
            ├── currencycontroller.py   # Контроллер для валют
            ├── usercontroller.py       # Контроллер для пользователей
            └── cbr_api.py              # Подключение к API ЦБ
    └── templates/
        ├── index.html
        ├── currencies.html
        ├── author.html
        ├── users.html
        └── user.html
    └── static/
        └── style.css
    └── tests/
        └── test_myapp.py          # Тесты с unittest и unittest.mock

```

4. Реализация CRUD с примерами SQL-запросов

4.1 Создание валюты (Create)

```

data = {
    "num_code": "840",
    "char_code": "USD",
    "name": "Доллар США",
    "value": 90.0,
    "nominal": 1
}
db_controller._create([data])

```

SQL-запрос:

```

INSERT INTO currency(num_code, char_code, name, value, nominal)
VALUES ('840', 'USD', 'Доллар США', 90.0, 1);

```

4.2 Чтение валют (Read)

```
currencies = db_controller._read()
```

SQL-запрос:

```
SELECT * FROM currency;
```

4.3 Обновление курса валюты (Update)

```
db_controller._update({"USD": 91.5})
```

SQL-запрос:

```
UPDATE currency SET value = 91.5 WHERE char_code = 'USD';
```

4.4 Удаление валюты (Delete)
db_controller._delete(1) # по id

SQL-запрос:

```
DELETE FROM currency WHERE id = 1;
```

4.5 CRUD для пользователей

```
user_id = db_controller.create_user("Игорь")
users = db_controller.read_users()
db_controller.assign_currency_to_user(user_id, 1)
user_currencies = db_controller.get_user_currencies(user_id)
```

5. Примеры работы приложения (главная страница, таблица валют, обновление и удаление).

The image shows two screenshots of a web application named "CurrenciesListApp".

Top Screenshot (Main Currency List):

The title bar says "CurrenciesListApp". The URL is "localhost:8080". The page header includes "Автор: Иголкин Владислав 504623" and "Группа: Р3124". Navigation links are "Главная", "Об авторе", and "Пользователи". The main content is a table titled "Валюты" with the following data:

ID	NumCode	CharCode	Name	Value	Nominal	Действия
1	840	USD	USD	76.0382	1	<button>Удалить</button>
2	978	EUR	EUR	88.7898	1	<button>Удалить</button>
3	826	GBP	GBP	102.1041	1	<button>Удалить</button>
4	036	AUD	AUD	51.744	1	<button>Удалить</button>

Bottom Screenshot (Author Profile):

The title bar says "Об авторе". The URL is "localhost:8080/author". The page title is "Информация об авторе". It displays the author's information: "Имя: Иголкин Владислав 504623" and "Группа: Р3124". A blue button at the bottom left says "На главную".

Пользователи

localhost:8080/users

Список пользователей

Главная

ID	Имя	Действия
1	Ярослав	Просмотр
2	Виктория	Просмотр
3	Антон	Просмотр

Пользователь

localhost:8080/user?id=1

Информация о пользователе

[Назад к списку пользователей](#) [Главная](#)

ID	1
Имя	Ярослав

Пользователь

localhost:8080/user?id=2

Информация о пользователе

[Назад к списку пользователей](#) [Главная](#)

ID	2
Имя	Виктория

Пользователь

localhost:8080/user?id=3

Информация о пользователе

[Назад к списку пользователей](#) [Главная](#)

ID	3
Имя	Антон

Список валют

localhost:8080/currency/delete?id=4

Все валюты

ID	NumCode	CharCode	Name	Value	Nominal	Действия
1	840	USD	USD	76.0382	1	Удалить
2	978	EUR	EUR	88.7898	1	Удалить
3	826	GBP	GBP	102.1041	1	Удалить

Список валют

localhost:8080/currency/delete?id=3

Все валюты

ID	NumCode	CharCode	Name	Value	Nominal	Действия
1	840	USD	USD	76.0382	1	Удалить
2	978	EUR	EUR	88.7898	1	Удалить

Список валют

localhost:8080/currency/delete?id=2

Все валюты

ID	NumCode	CharCode	Name	Value	Nominal	Действия
1	840	USD	USD	76.0382	1	Удалить

6. Примеры тестов с unittest.mock и результаты их выполнения.

```

import sys
import os
sys.path.append(os.path.dirname(os.path.abspath(__file__)))
import unittest
from unittest.mock import MagicMock
from .controllers.currencycontroller import CurrencyController
from .controllers.usercontroller import UserController


class TestCurrencyController(unittest.TestCase):
    def setUp(self):
        self.mock_db = MagicMock()
        self.controller = CurrencyController(self.mock_db)

    def test_list_currencies(self):
        self.mock_db._read.return_value = [{"id": 1, "char_code": "USD", "value": 90}]
        result = self.controller.list_currencies()
        self.assertEqual(result[0]['char_code'], "USD")
        self.mock_db._read.assert_called_once()

    def test_update_currency(self):
        self.controller.update_currency("USD", 100.5)
        self.mock_db._update.assert_called_once_with({"USD": 100.5})

    def test_delete_currency(self):
        self.controller.delete_currency(1)
        self.mock_db._delete.assert_called_once_with(1)

class TestUserController(unittest.TestCase):

```

```

def setUp(self):
    self.controller = UserController()
    self.controller.add_user("Антон")
    self.controller.add_user("Мария")

def test_list_users(self):
    users = self.controller.list_users()
    self.assertEqual(len(users), 2)
    self.assertEqual(users[0].name, "Антон")
    self.assertEqual(users[1].name, "Мария")

def test_get_user(self):
    user = self.controller.get_user(1)
    self.assertEqual(user.name, "Антон")
    self.assertIsNone(self.controller.get_user(99))

def test_add_user(self):
    self.controller.add_user("Игорь")
    users = self.controller.list_users()
    self.assertEqual(len(users), 3)
    self.assertEqual(users[2].name, "Игорь")

if __name__ == "__main__":
    unittest.main()

```

```

run Python tests in test_myapp.py
Run Configuration: Python tests in test_myapp.py | C:\Users\solik\PycharmProjects\lab9\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm 2021.1.1\helpers\pydev\pydevconsole.py" -w C:\Users\solik\PycharmProjects\lab9\test_myapp.py
Test Results 2ms
✓ 6 tests passed 6 tests total, 2ms
C:\Users\solik\PycharmProjects\lab9\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm 2021.1.1\helpers\pydev\pydevconsole.py" -w C:\Users\solik\PycharmProjects\lab9\test_myapp.py
Testing started at 22:49 ...
Launching unitests with arguments python -m unittest C:\Users\solik\PycharmProjects\lab9\test_myapp.py
Ran 6 tests in 0.003s
OK
Process finished with exit code 0

```

7. Выводы

- В работе реализован полный CRUD для валют и пользователей.
- Использование SQLite показало работу с базой через parameterized queries для защиты от SQL-инъекций.
- Архитектура MVC позволяет разделять логику (controllers), данные (models) и представление (templates).
- Тесты с unittest.mock позволяют проверять контроллеры без подключения к реальной базе.
- Jinja2 обеспечивает гибкий рендеринг шаблонов HTML с динамическими данными.