Лабораторная работа №3

Фикстуры

```python
# Фикстуры для базы данных
@pytest.fixture  2 usages
def db_connection():
    """Фикстура для создания временной базы данных в памяти"""
    connection = sqlite3.connect(":memory:")
    yield connection
    connection.close()


@pytest.fixture
def auth_manager(db_connection):
    """Фикстура для создания экземпляра AuthManager"""
    return AuthManager(db_connection)


@pytest.fixture  19 usages
def populated_auth_manager(auth_manager):
    """Фикстура с предзаполненными данными пользователей"""
    # Добавляем тестовых пользователей
    test_users = [
        ("user1", "pass1", "USA", 1000.0),
        ("user2", "pass2", "USA", 2000.0),
        ("user3", "pass3", "Canada", 1500.0),
        ("user4", "pass4", "UK", 3000.0),
        ("user5", "pass5", "Germany", 2500.0)
    ]
```

Базовые тесты:

1) Тест создания таблиц

```
    for username, password, country, balance in test_users:
        auth_manager.register_user(username, password, country, balance)

    return auth_manager


# Базовые тесты (3 штуки)
def test_create_tables(auth_manager):
    """Тест создания таблиц"""
    cursor = auth_manager.connection.cursor()
    cursor.execute("SELECT name FROM sqlite_master WHERE type='table' AND name='users'")
    result = cursor.fetchone()
    assert result is not None
    assert result[0] == 'users'
```

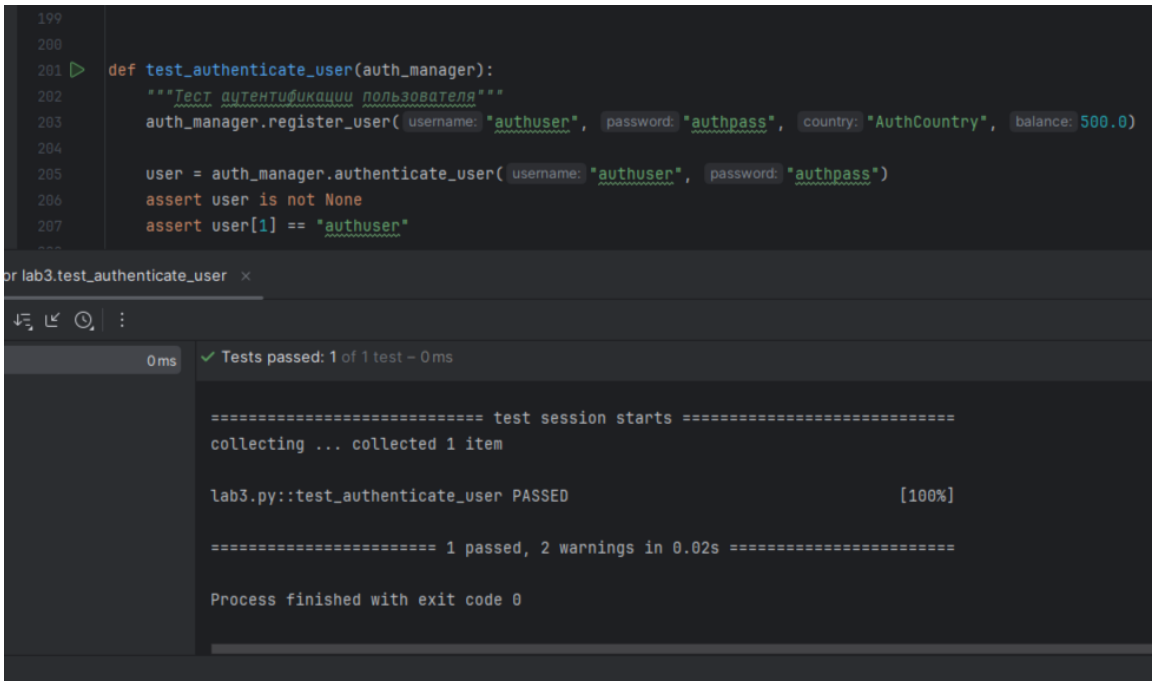
```
lab3.test_create_tables
0 ms
Tests passed: 1 of 1 test – 0 ms
Launching pytest with arguments lab3.py::test_create_tables --no-header --no-summary -q in C:\Users\...

============================= test session starts =============================
collecting ... collected 1 item

lab3.py::test_create_tables PASSED                                       [100%]

======================== 1 passed, 2 warnings in 0.02s ========================

Process finished with exit code 0
```

2) Тест регистрации пользователей

```
def test_register_user(auth_manager):
    """Тест регистрации пользователя"""
    auth_manager.register_user( username: "testuser",  password: "testpass",  country: "TestCountry",  balance: 1000.0)

    cursor = auth_manager.connection.cursor()
    cursor.execute("SELECT * FROM users WHERE username = 'testuser'")
    user = cursor.fetchone()

    assert user is not None
    assert user[1] == "testuser"
    assert user[2] == "testpass"
    assert user[3] == "TestCountry"
    assert user[4] == 1000.0


def test_authenticate_user(auth_manager):
    """Тест аутентификации пользователя"""
    auth_manager.register_user( username: "authuser",  password: "authpass",  country: "AuthCountry",  balance: 500.0)
```

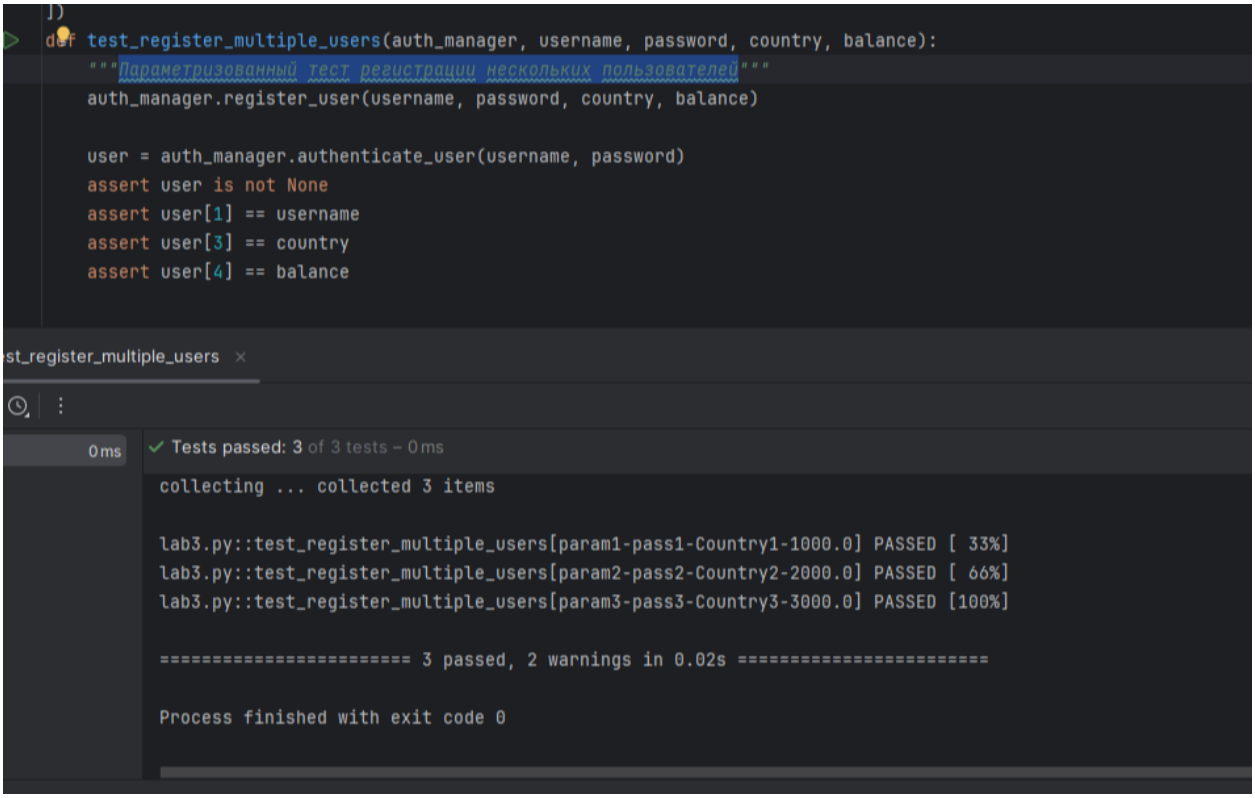
```
est_register_user
0 ms
Tests passed: 1 of 1 test – 0 ms

============================= test session starts =============================
collecting ... collected 1 item

lab3.py::test_register_user PASSED                                       [100%]

======================== 1 passed, 2 warnings in 0.02s ========================

Process finished with exit code 0
```

3) Тест аутентификации пользователей

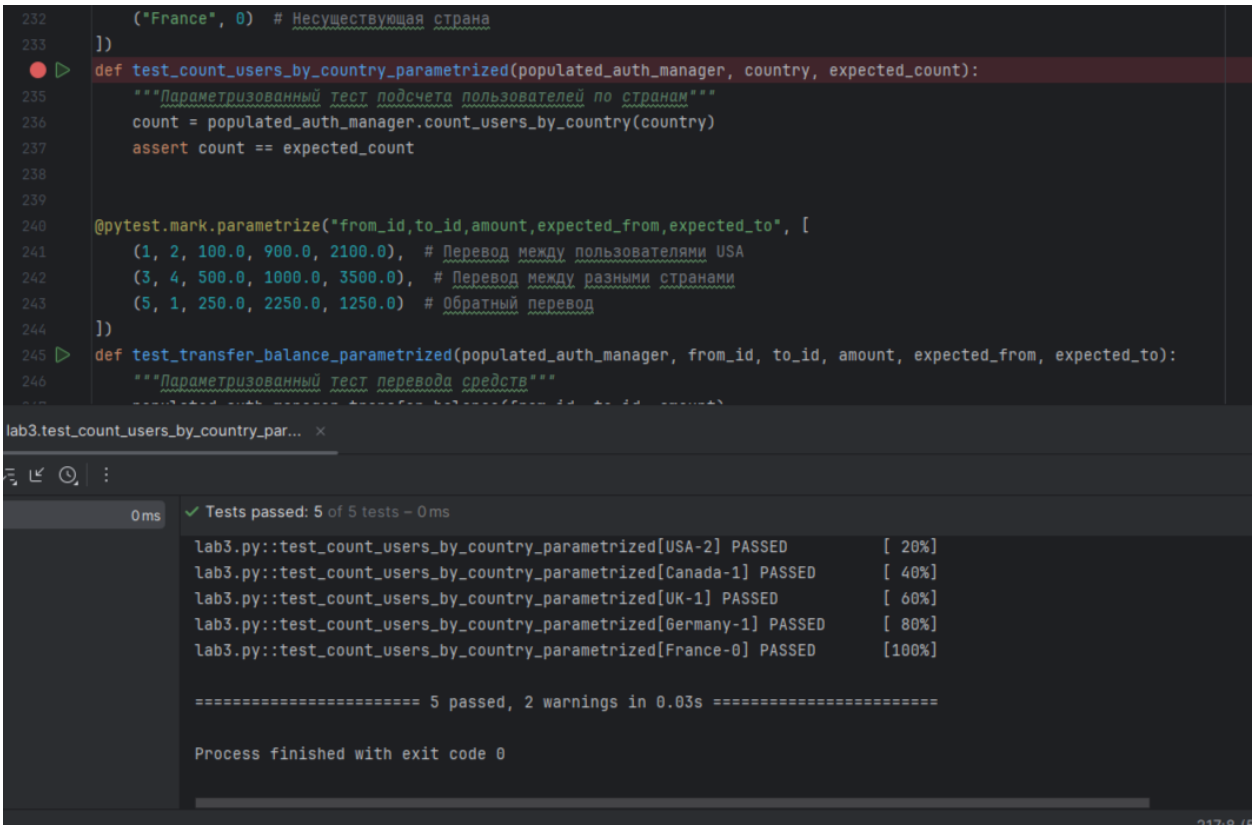


Параметрические тесты

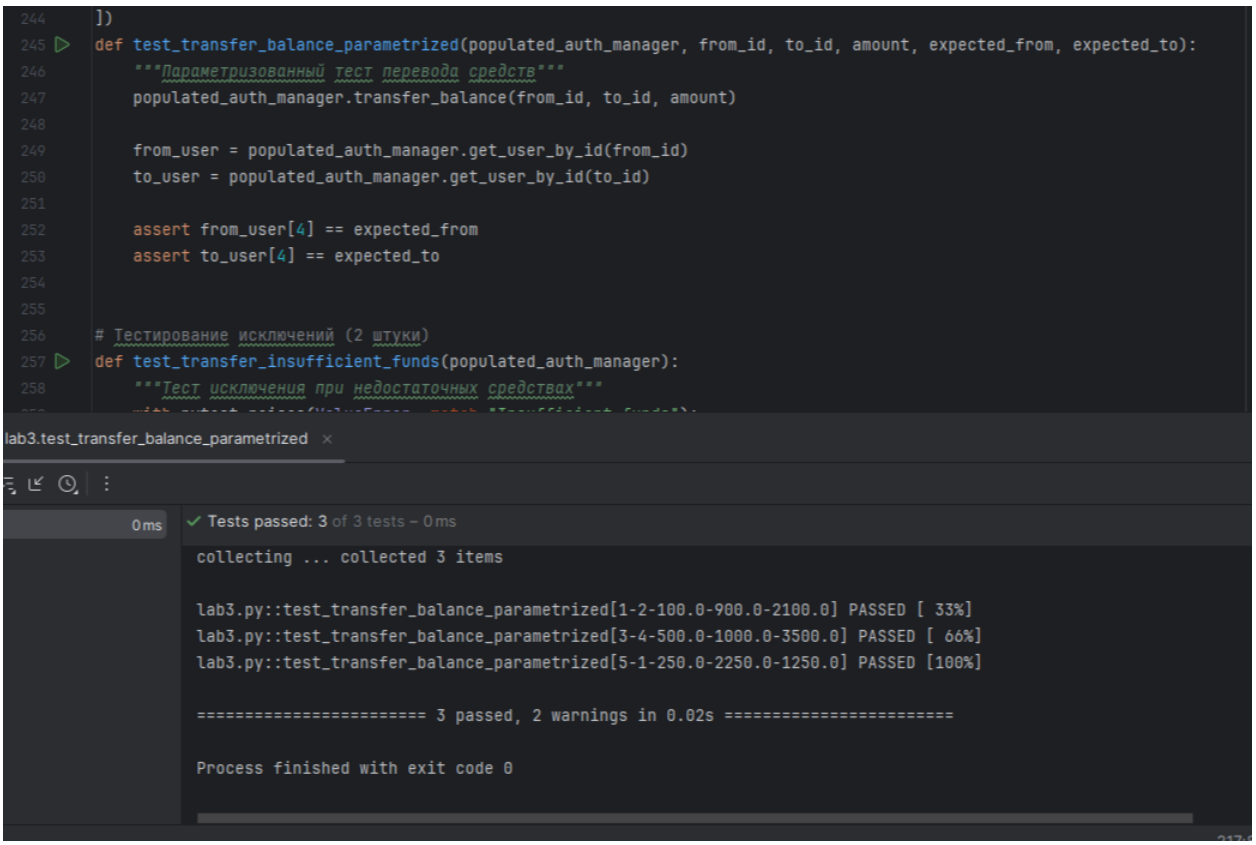
1) Тест регистрации нескольких пользователей



2) Тест подсчета пользователей по странам

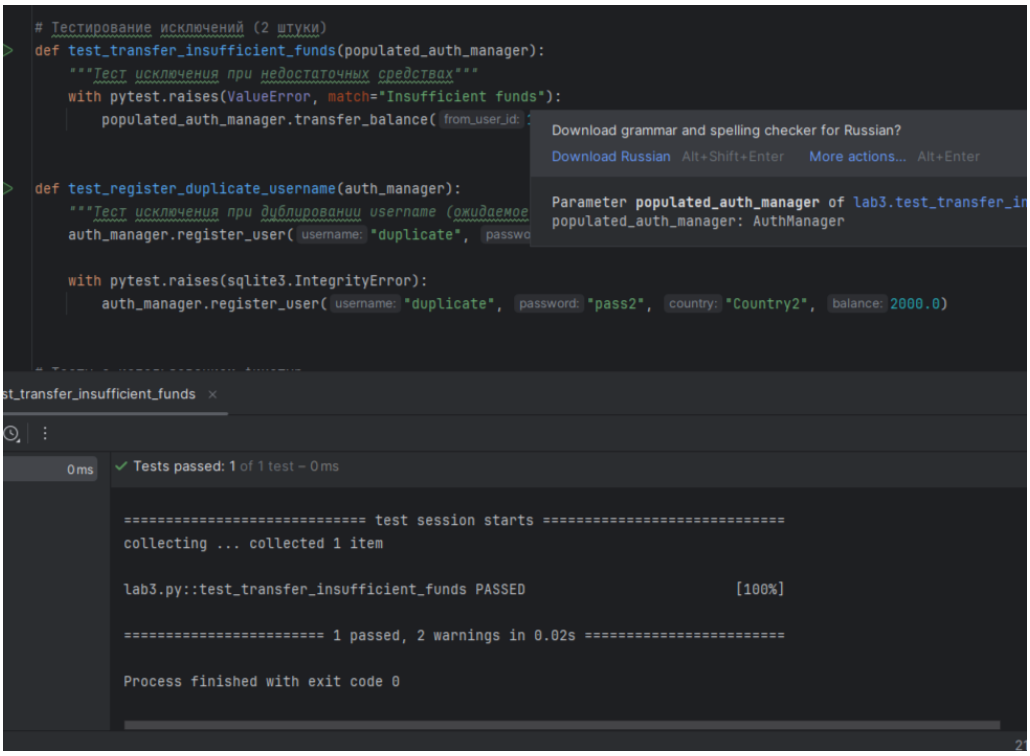


3) Тест перевода средств

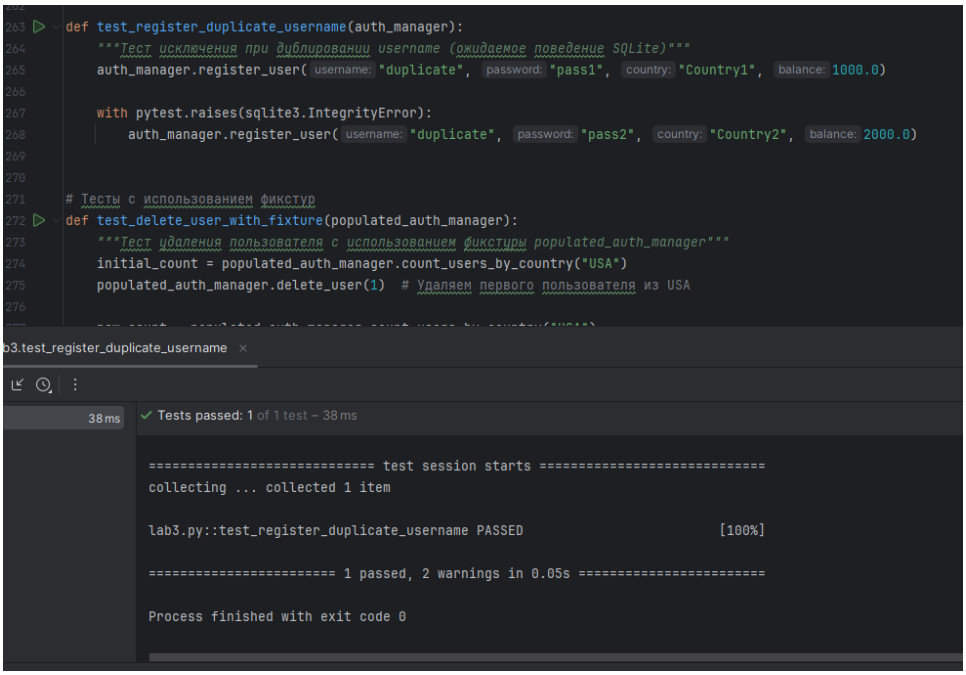


Тестирование исключений

1) Тест исключения при недостаточных средствах



2) Тест исключения при дублировании username



Тесты с использованием фикстур

1) Тест удаления пользователя с ипользованием фикстуры populated_auth_manager

```python
def test_delete_user_with_fixture(populated_auth_manager):
    """Тест удаления пользователя с использованием фикстуры populated_auth_manager"""
    initial_count = populated_auth_manager.count_users_by_country("USA")
    populated_auth_manager.delete_user(1)  # Удаляем первого пользователя из USA

    new_count = populated_auth_manager.count_users_by_country("USA")
    assert new_count == initial_count - 1

    # Проверяем, что пользователь действительно удален
    user = populated_auth_manager.get_user_by_id(1)
    assert user is None


def test_get_user_by_id_with_fixture(populated_auth_manager):
```

```
lab3.test_delete_user_with_fixture
38 ms
Tests passed: 1 of 1 test – 38 ms

============================= test session starts ==============================
collecting ... collected 1 item

lab3.py::test_delete_user_with_fixture PASSED                            [100%]

======================== 1 passed, 2 warnings in 0.06s =========================

Process finished with exit code 0
```

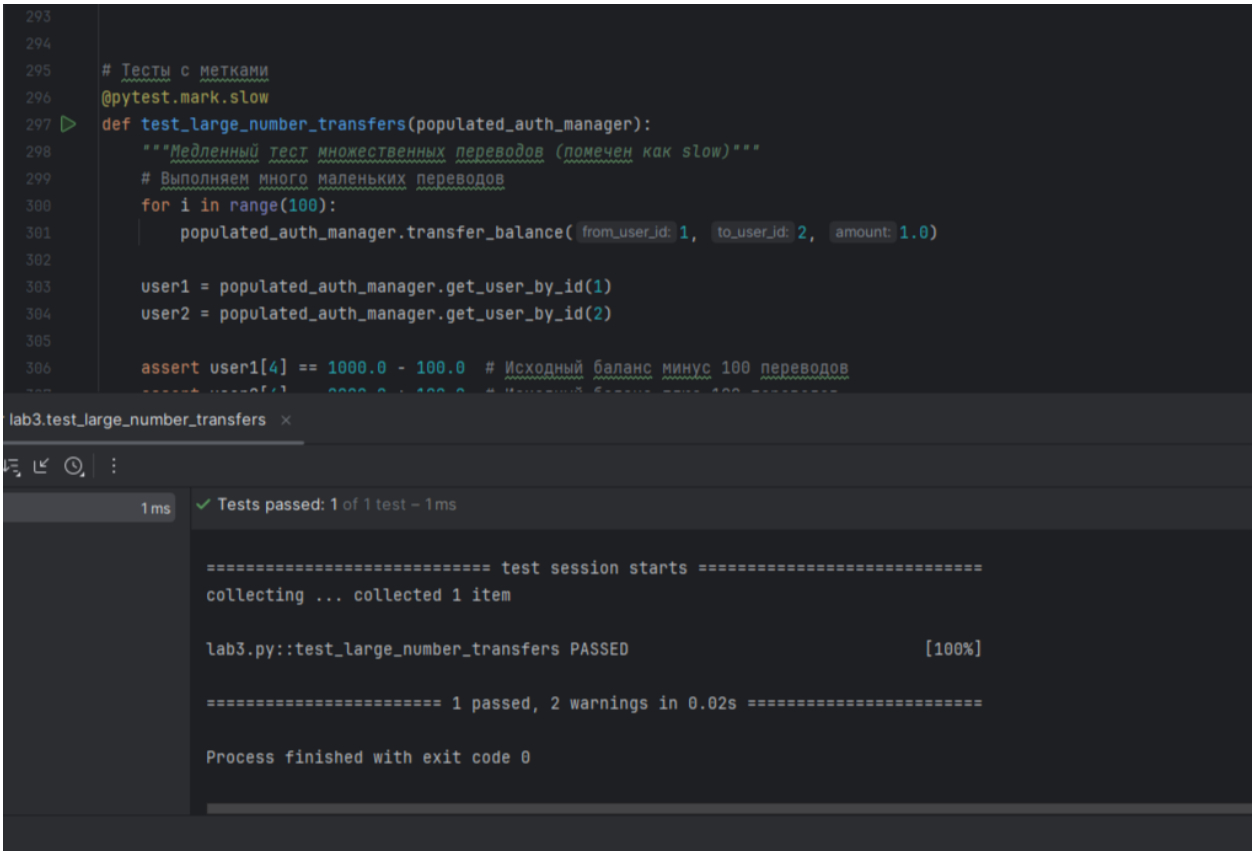
2) Тест получения пользователя по ID с использованием фикстуры

```python
def test_get_user_by_id_with_fixture(populated_auth_manager):
    """Тест получения пользователя по ID с использованием фикстуры"""
    user = populated_auth_manager.get_user_by_id(2)  # Второй пользователь

    assert user is not None
    assert user[0] == 2  # ID
    assert user[1] == "user2"  # username
    assert user[3] == "USA"  # country


# Тесты с метками
@pytest.mark.slow
def test_large_number_transfers(populated_auth_manager):
```

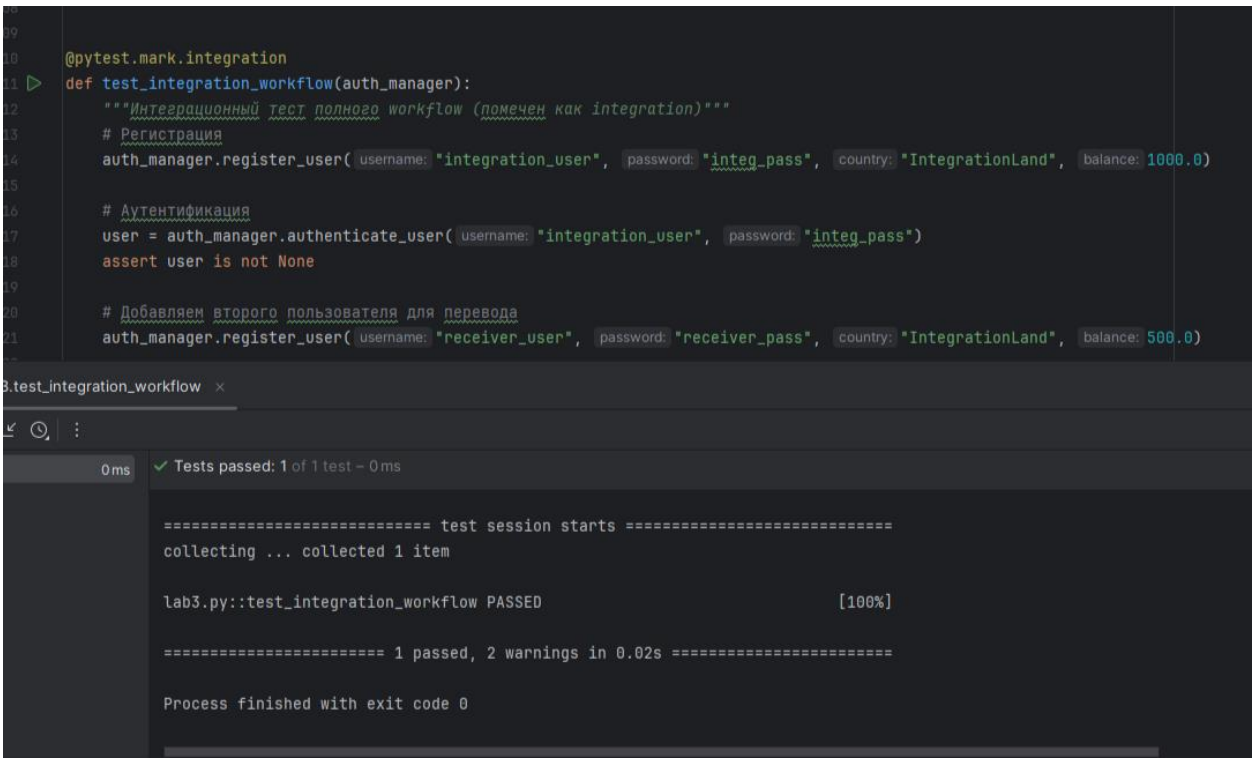
```
or lab3.test_get_user_by_id_with_fixture
0 ms
Tests passed: 1 of 1 test – 0 ms

============================= test session starts ==============================
collecting ... collected 1 item

lab3.py::test_get_user_by_id_with_fixture PASSED                         [100%]

======================== 1 passed, 2 warnings in 0.02s =========================

Process finished with exit code 0
```

Тесты с метками

1) Медленный тест множественных переводов



2) Интеграционный тест полного workflow



Тест демонстрации SQL-инъекции при аутентификации