



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных  
технологий

### **Отчет по практической работе №3**

по дисциплине «Тестирование и верификация ПО»

**Выполнил:**

Студент группы ИКБО-25-22

Ракитин В.А.

**Проверил:**

Ассистент

Овчинникова М.А.

2024 г.

## СОДЕРЖАНИЕ

1 Разработка программного модуля методом Test-Driven Development.....	3
2 Разработка программного модуля методом Behavior Driven Development.....	5
3 Заключение .....	8

# 1 Разработка программного модуля методом Test-Driven Development

## Создание теста, описывающего ожидание поведения функции

Вариант 23. Программа для напоминания о днях рождения друзей.

### Листинг 1 – юнит-тесты

```
import unittest
from main import birthdayCheck

class MyTestCase(unittest.TestCase):
    def testCheckBirthdays(self):
        self.assertEqual(birthdayCheck("Тоха"), "19.01.2004")
        self.assertEqual(birthdayCheck("Саня"), "23.01.2003")
        self.assertEqual(birthdayCheck("Даша"), "15.07.2004")
        self.assertEqual(birthdayCheck("Петя"), "14.11.2004")
        self.assertEqual(birthdayCheck("Миха"), "15.10.2003")
        self.assertEqual(birthdayCheck("Андрей"), "Он тебе не
друг")
        self.assertEqual(birthdayCheck("Артем"), "Он тебе не
друг")

if __name__ == '__main__':
    unittest.main()
```

test\_something проверяет все возможные вводы в консоль.

### Запуск юнит теста.

Запуск юнит теста представлен на рисунке 1. Выдает ошибку т.к.

функции еще не существует

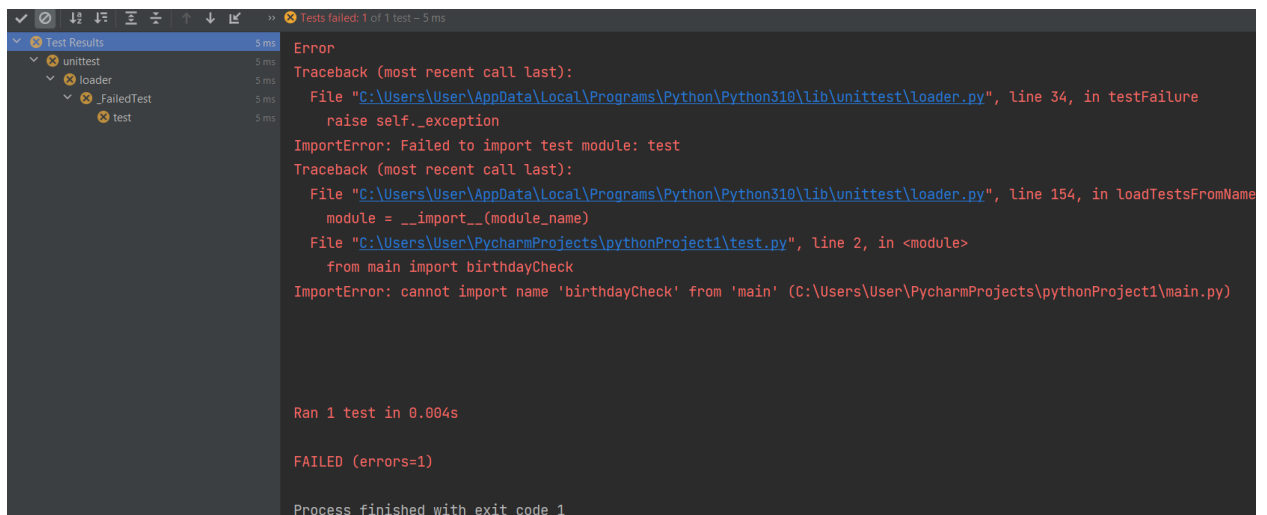


Рисунок 1 – Запуск юнит теста

### Реализация функции

### Листинг 2 – Основная программа

```
def birthdayCheck(name):
    birthdays = {
```

```
        "Тоха": "19.01.2004",  
        "Саня": "23.01.2003",  
        "Даша": "15.07.2004",  
        "Петя": "14.11.2004",  
        "Миха": "15.10.2003"  
    }  
    return birthdays.get(name, "Он тебе не друг")
```

## Повторный запуск тестов

Ниже на рисунке 2 представлена работа юнит теста.

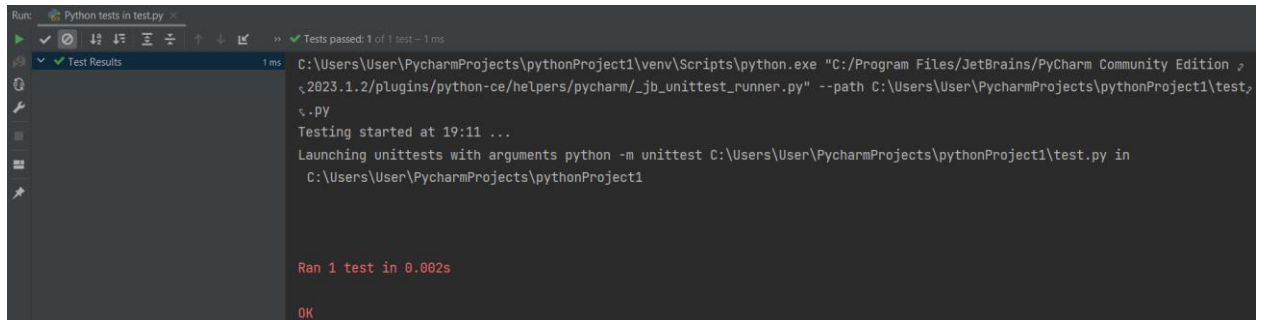


Рисунок 2 – Работа юнит теста

## 2 Разработка программного модуля методом Behavior Driven Development

### Создание описания сценариев

Ниже представлен листинг описания сценариев, на котором описан минимальный сценарий о днях рождения друзей.

#### Листинг 3 – Описание сценариев

Feature: Напоминание о днях рождениях друзей

```
Scenario: Проверка "Тоха"  
  Given я ввожу "Тоха"  
  When вызов функции  
  Then я вижу "19.01.2004"
```

```
Scenario: Проверка "Саня"  
  Given я ввожу "Саня"  
  When вызов функции  
  Then я вижу "23.01.2003"
```

```
Scenario: Проверка "Даша"  
  Given я ввожу "Даша"  
  When вызов функции  
  Then я вижу "15.07.2004"
```

```
Scenario: Проверка "Петя"  
  Given я ввожу "Петя"  
  When вызов функции  
  Then я вижу "14.11.2004"
```

```
Scenario: Проверка "Миха"  
  Given я ввожу "Миха"  
  When вызов функции  
  Then я вижу "15.10.2003"
```

```
Scenario: Проверка "Андрей"  
  Given я ввожу "Андрей"  
  When вызов функции  
  Then я вижу "Он тебе не друг"
```

```
Scenario: Проверка "Артем"  
  Given я ввожу "Артем"  
  When вызов функции  
  Then я вижу "Он тебе не друг"
```

### Реализация функции и автоматизация сценариев

## Листинг 4 – Автоматизация сценариев и функция напоминание о днях рождениях

```
from behave import given, when, then

class BirthdayReminder:
    def __init__(self):
        self.friends_birthdays = {
            "Тоха": "19.01.2004",
            "Саня": "23.01.2003",
            "Даша": "15.07.2004",
            "Петя": "14.11.2004",
            "Миха": "15.10.2003"
        }

    def get_birthday(self, name):
        return self.friends_birthdays.get(name, "Он тебе не друг")

@given('вызов функции')
def step_given_function(context):
    context.reminder = BirthdayReminder()

@when('я ввожу "{name}"')
def step_when_input_name(context, name):
    context.birthday = context.reminder.get_birthday(name)

@then('я вижу "{expected_birthday}"')
def step_then_see_birthday(context, expected_birthday):
    assert context.birthday == expected_birthday
```

### Запуск тестов

На рисунках 3 и 4 представлено работа тестов.

```
C:\Users\User\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:\Users\User\PycharmProjects\pythonProject1\venv\Lib\site-packages\behave feature\test
.feature
```

Feature: Напоминание о днях рождениях друзей # feature/test.feature:1

```
Scenario: Проверка "Тоха" # feature/test.feature:3
  Given вызов функции # feature/steps/birthday_reminder.py:16
  When я ввожу "Тоха" # feature/steps/birthday_reminder.py:20
  Then я вижу "19.01.2004" # feature/steps/birthday_reminder.py:24
```

```
Scenario: Проверка "Саня" # feature/test.feature:8
  Given вызов функции # feature/steps/birthday_reminder.py:16
  When я ввожу "Саня" # feature/steps/birthday_reminder.py:20
  Then я вижу "23.01.2003" # feature/steps/birthday_reminder.py:24
```

```
Scenario: Проверка "Даша" # feature/test.feature:13
  Given вызов функции # feature/steps/birthday_reminder.py:16
  When я ввожу "Даша" # feature/steps/birthday_reminder.py:20
  Then я вижу "15.07.2004" # feature/steps/birthday_reminder.py:24
```

```
Scenario: Проверка "Петя" # feature/test.feature:18
  Given вызов функции # feature/steps/birthday_reminder.py:16
  When я ввожу "Петя" # feature/steps/birthday_reminder.py:20
  Then я вижу "14.11.2004" # feature/steps/birthday_reminder.py:24
```

### Рисунок 3 – Работа тестов

```
Scenario: Проверка "Миха" # feature/test.feature:23
  Given вызов функции      # feature/steps/birthday_reminder.py:16
  When я ввожу "Миха"      # feature/steps/birthday_reminder.py:20
  Then я вижу "15.10.2003" # feature/steps/birthday_reminder.py:24

Scenario: Проверка "Андрей" # feature/test.feature:28
  Given вызов функции      # feature/steps/birthday_reminder.py:16
  When я ввожу "Андрей"    # feature/steps/birthday_reminder.py:20
  Then я вижу "Он тебе не друг" # feature/steps/birthday_reminder.py:24

Scenario: Проверка "Артем" # feature/test.feature:33
  Given вызов функции      # feature/steps/birthday_reminder.py:16
  When я ввожу "Артем"     # feature/steps/birthday_reminder.py:20
  Then я вижу "Он тебе не друг" # feature/steps/birthday_reminder.py:24

1 feature passed, 0 failed, 0 skipped
7 scenarios passed, 0 failed, 0 skipped
21 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m0.010s

Process finished with exit code 0
```

## Рисунок 4 – Работа тестов

### **3 Заключение**

В ходе практической работы мы изучили методологию Test-Driven Development (Разработка, управляемая тестами) и методологию Behavior Driven Development (Разработка, ориентированная на поведение).