

Московский государственный технический Университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет рубежному контролю №2

Выполнил:

студент группы ИУ5-31Б

Тянутов А.Д.

Проверил:

Гапанюк Е.Ю.

Москва 2022г.

Условия рубежного контроля №2 по курсу БКИТ

Рубежный контроль представляет собой разработку тестов на языке Python.

1. Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования
2. Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

Измененный код РК1

В рамках РК2 был изменен только файл main:

```
1. from rk_library import rk_name_lib
2. from prog_lang import rk_name_pl
3.
4. class Combined_table:
5.     def __init__(self, id_pl : rk_name_pl, id_lib :
      rk_name_lib):
6.         self.id_lib = id_lib.get_id()
7.         self.id_pl = id_pl.get_id()
8.     def __lt__(self, other):
9.         return self.id_pl < other.id_pl
10.    def __repr__(self) -> str:
11.        return "{}->{}".format(self.id_pl,
      self.id_lib)
12.
13.    def task_1(fct: list[Combined_table]):
14.        print('Before sorting\n', fct)
15.        fct.sort(reverse=True)
16.        print('After sorting\n', fct)
17.        return fct
```

Файл тестирования:

```
import unittest
from unittest import main
from rk_library import rk_name_lib
from prog_lang import rk_name_pl
from main import task_1, Combined_table

class test_class(unittest.TestCase):
    def __init__(self, methodName: str = ...) -> None:
        super().__init__(methodName)
```

```

        self.lst_rk_l = []
        self.lst_rk_pl = []
        rk_l1 = rk_name_pl('FirstPl', 100)
        self.lst_rk_pl.append(rk_l1)
        rk_l2 = rk_name_pl('SecondPl', 200)
        self.lst_rk_pl.append(rk_l2)
        rk_l3 = rk_name_pl('ThirdPl', 300)
        self.lst_rk_pl.append(rk_l3)
        lib_1 = rk_name_lib('FirstLib', [rk_l1])
        self.lst_rk_l.append(lib_1)
        lib_2 = rk_name_lib('SecondLib', [rk_l1, rk_l2])
        self.lst_rk_l.append(lib_2)
        lib_3 = rk_name_lib('ThirdLib', [rk_l1])
        self.lst_rk_l.append(lib_3)
        self.libs_langs = [
            Combined_table(rk_l1, lib_1),
            Combined_table(rk_l1, lib_2),
            Combined_table(rk_l2, lib_2),
            Combined_table(rk_l1, lib_3)
        ]

    def test_default_case(self):
        self.assertEqual(self.lst_rk_pl[0].id, 0)
        self.assertEqual(self.lst_rk_pl[0].name,
'FirstPl')
    def test_prog_lang(self):
        self.assertEqual(self.lst_rk_pl[1].id, 4)
    def test_prog_lang_has_name(self):
        self.assertEqual(self.lst_rk_pl[1].name,
'SecondPl')
    def test_table(self):
        self.assertEqual(type(self.lst_rk_pl[0]),
rk_name_pl)

if __name__ == '__main__':
    main()

```

Результат тестирования:

```

PS D:\Projects\BKIT\rk2> python .\test_for_rk1.py
....
-----
Ran 4 tests in 0.001s

OK
PS D:\Projects\BKIT\rk2> |

```