

**Московский государственный
технический университет им. Н.Э.
Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по РК №2
Вариант запросов: В
Вариант предметной области: 22

Выполнил:
студент группы ИУ5-33Б
Смирнов Артём

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2023 г.

Вариант запросов В. Предметная область 22.

- 1) «Язык программирования» и «Библиотека» связаны соотношением один-ко-многим. Выведите список всех библиотек, которые начинаются с буквы «А», и названия их языков программирования.
- 2) «Язык программирования» и «Библиотека» связаны соотношением один-ко-многим. Выведите список языков программирования с минимальный размером библиотеки в каждом языке программирования, отсортированный по минимальному размеру.
- 3) «Язык программирования» и «Библиотека» связаны соотношением многие-ко-многим. Выведите список всех связанных библиотек и языков программирования, отсортированный по библиотекам, сортировка по языкам программирования произвольная.

Задание

- 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).

Файл main.py

```
# используется для сортировки
from operator import itemgetter

class Library:
    def __init__(self, id, name, size, language_id):
        self.id = id
        self.name = name
        self.size = size # in Mb
        self.language_id = language_id

class ProgrammingLanguage:
    def __init__(self, id, name):
        self.id = id
        self.name = name

class LanguageLibrary:
    def __init__(self, language_id, library_id):
        self.language_id = language_id
        self.library_id = library_id
```

```

# Языки программирования
languages = [
    ProgrammingLanguage(1, "Python"),
    ProgrammingLanguage(2, "JavaScript"),
    ProgrammingLanguage(3, "Java")
]

# Библиотеки
libraries = [
    Library(1, "NumPy", 512, 1),
    Library(2, "Pandas", 1024, 1),
    Library(3, "React", 2048, 2),
    Library(4, "Node.js", 256, 2),
    Library(5, "Spring", 4096, 3),
    Library(6, "Angular.js", 128, 2)
]

language_library = [
    LanguageLibrary(1, 1),
    LanguageLibrary(1, 2),
    LanguageLibrary(2, 3),
    LanguageLibrary(2, 4),
    LanguageLibrary(3, 5),
    LanguageLibrary(2, 6)
]

def join_one_to_many(languages, libraries):
    return [(library.name, library.size, language.name)
            for language in languages
            for library in libraries
            if library.language_id == language.id]

def join_many_to_many(languages, language_library, libraries):
    many_to_many_temp = [(language.name, LangLibs.language_id, LangLibs.library_id)
                          for language in languages
                          for LangLibs in language_library
                          if language.id == LangLibs.language_id]

    return [(library.name, library.size, lang_name)
            for lang_name, lang_id, lib_id in many_to_many_temp
            for library in libraries
            if library.id == lib_id]

def get_a1_result(data):
    res_11 = []
    for i in range(len(data)):
        if data[i][0][0] == 'A':
            res_11.append((data[i][0], data[i][2]))
    return res_11

```

```

def get_a2_result(data):
    res_12 = {}
    for i in range(len(data)):
        if data[i][2] not in res_12:
            res_12[data[i][2]] = data[i][1]
        elif data[i][1] < res_12[data[i][2]]:
            res_12[data[i][2]] = data[i][1]
    return sorted(res_12.items(), key=lambda x: x[1])

def get_a3_result(data):
    result = sorted(data, key=itemgetter(0))
    return result

# Остальной код оставляем как есть, добавим всего лишь вызовы новых функций внутри
main()
def main():
    one_to_many = join_one_to_many(languages, libraries)
    many_to_many = join_many_to_many(languages, language_library, libraries)

    print('\n')
    print('Задание A1\n')
    res_11 = get_a1_result(one_to_many)
    for i in range(len(res_11)):
        print(*res_11[i])

    print('\nЗадание A2\n')
    res_12 = get_a2_result(one_to_many)
    for i in range(len(res_12)):
        print(*res_12[i])

    print('\nЗадание A3\n')
    res_13 = get_a3_result(many_to_many)
    for i in range(len(res_13)):
        print(*res_13[i])
    print('\n')

if __name__ == "main":
    main()

```

Файл TDD_tests.py

```

import unittest
from main import *

class TestJoinOperations(unittest.TestCase):
    def test_join_one_to_many(self):
        result = join_one_to_many(languages, libraries)
        self.assertTrue(len(result) > 0) # Проверка, что результат не пустой

    def test_join_many_to_many(self):
        result = join_many_to_many(languages, language_library, libraries)
        self.assertTrue(len(result) > 0) # Проверка, что результат не пустой

```

```

class TestTaskResults(unittest.TestCase):
    def test_get_a1_result(self):
        data = [("NumPy", 100, "Python"), ("Angular", 200, "JavaScript"), ("React",
300, "JavaScript")]
        result = get_a1_result(data)
        self.assertEqual(result, [("Angular", "JavaScript")]) # Проверка корректности
результата

    def test_get_a1_empty_result(self):
        data = [] # пустой список
        result = get_a1_result(data)
        self.assertEqual(result, []) # ожидаемый результат - пустой список

    def test_get_a2_result(self):
        data = [("NumPy", 100, "Python"), ("Angular", 200, "JavaScript"), ("React",
300, "JavaScript"), ("Pandas", 1024, "Python")]
        result = get_a2_result(data)
        self.assertEqual(result, [("Python", 100), ("JavaScript", 200)]) # Проверка
корректности результата

    def test_get_a3_result(self):
        data = [
            ("Library A", 100, "Language 1"),
            ("Library B", 200, "Language 2"),
            ("Library C", 150, "Language 1")
        ]
        result = sorted(data, key=itemgetter(0)) # ожидаемый результат -
отсортированный по имени библиотеки
        self.assertEqual(result, get_a3_result(data))

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

```

Файл test_runner.py

```

import TDD_tests
import unittest

suite = unittest.TestLoader().loadTestsFromModule(TDD_tests)
results = unittest.TextTestRunner(verbosity=2).run(suite)

```

Результат выполнения

```
PS E:\Python 3rd Semestr\git_copy\3rd-semester-codes\RK2> & "C:/Program Files/Python312/python.exe" "e:/Python 3rd Semestr/git_copy/3rd-semester-codes/RK2/test_runner.py"

Задание A1

Angular.js JavaScript

Задание A2

JavaScript 128
Python 512
Java 4096

Задание A3

Angular.js 128 JavaScript
Node.js 256 JavaScript
NumPy 512 Python
Pandas 1024 Python
React 2048 JavaScript
Spring 4096 Java

test_join_many_to_many (TDD_tests.TestJoinOperations.test_join_many_to_many) ... ok
test_join_one_to_many (TDD_tests.TestJoinOperations.test_join_one_to_many) ... ok
test_get_a1_empty_result (TDD_tests.TestTaskResults.test_get_a1_empty_result) ... ok
test_get_a1_result (TDD_tests.TestTaskResults.test_get_a1_result) ... ok
test_get_a2_result (TDD_tests.TestTaskResults.test_get_a2_result) ... ok
test_get_a3_result (TDD_tests.TestTaskResults.test_get_a3_result) ... ok

-----
Ran 6 tests in 0.001s

OK
PS E:\Python 3rd Semestr\git_copy\3rd-semester-codes\RK2>
```