Группа ИУ5-35Б
Студент Рябов Матвей

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)

Вариант №22 «Библиотека – Язык программирования»

Вариант запросов Г.

1. «Язык программирования» и «Библиотека» связаны соотношением один-ко-многим. Выведите список всех языков программирования, у которых название начинается с буквы «J», и список их библиотек.
2. «Язык программирования» и «Библиотека» связаны соотношением один-ко-многим. Выведите список языков программирования с максимальным количеством скачиваний библиотеки данного языка, отсортированный по максимальному количеству скачиваний.
3. «Язык программирования» и «Библиотека» связаны соотношением многие-ко-многим. Выведите список всех связанных библиотек и языков программирования, отсортированный по языкам программирования, сортировка по библиотекам произвольная.

main.py

```python
from operator import itemgetter

class Library:
    def __init__(self, id, name, downloads, language_id):
        self.id = id
        self.name = name
        self.downloads = downloads
        self.language_id = language_id
```

```python
class Language:
    def __init__(self, id, name):
        self.id = id
        self.name = name

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

languages = [
    Language(1, "Python"),
    Language(2, "C/C++"),
    Language(3, "Javascript"),
    Language(4, "Java")
]

libraries = [
    Library(1, "NumPy", 16_000_000, 1),
    Library(2, "Pandas", 13_000_000, 1),
    Library(3, "SFML", 20_000, 2),
    Library(4, "React", 6_500_000, 3),
    Library(5, "Guava", 4_000_000, 4)
]

libraries_languages = [
    LibraryLanguage(1, 1),
    LibraryLanguage(1, 2),
    LibraryLanguage(2, 1),
    LibraryLanguage(2, 2),
    LibraryLanguage(3, 2),
    LibraryLanguage(4, 3),
    LibraryLanguage(5, 4)
]

def build_one_to_many(libraries, languages):
    return [
        (lib.name, lib.downloads, lang.name)
        for lib in libraries
        for lang in languages
        if lib.language_id == lang.id
    ]

def build_many_to_many(libraries, languages, libraries_languages):
    many_to_many_temp = [
        (lang.name, lib_lang.language_id, lib_lang.library_id)
        for lang in languages
        for lib_lang in libraries_languages
        if lang.id == lib_lang.language_id
    ]
```

```
    return [
        (lib.name, lib.downloads, lang_name)
        for lang_name, _, lib_id in many_to_many_temp
        for lib in libraries
        if lib.id == lib_id
    ]

def first_task(one_to_many):
    result = {}

    for lang in [lang_obj.name for lang_obj in languages]:
        libs = [lib_name for lib_name, _, target_lang in one_to_many if
target_lang == lang]
        if lang.startswith("J"):
            result[lang] = libs

    return result

def second_task(one_to_many):
    result = []

    for lang in [lang_obj.name for lang_obj in languages]:
        libs = [(name, downloads) for name, downloads, target_lang in one_to_many
if target_lang == lang]
        if libs:
            max_downloads = max(downloads for _, downloads in libs)
            result.append((lang, max_downloads))
        else:
            result.append((lang, 0))

    return sorted(result, key=itemgetter(1), reverse=True)

def third_task(many_to_many):
    return sorted(many_to_many, key=itemgetter(2))

def main():
    one_to_many = build_one_to_many(libraries, languages)
    many_to_many = build_many_to_many(libraries, languages, libraries_languages)

    print("Задание Г1")
    print(first_task(one_to_many))
    print("")

    print("Задание Г2")
    print(second_task(one_to_many))
    print("")

    print("Задание Г3")
    print(third_task(many_to_many))
```

```
if __name__ == "__main__":
    main()
```

test_main.py

```
import unittest
from main import *

class TestLibraryFunctions(unittest.TestCase):
    def setUp(self):
        self.one_to_many = build_one_to_many(libraries, languages)
        self.many_to_many = build_many_to_many(libraries, languages,
libraries_languages)

    def test_first_task(self):
        result = first_task(self.one_to_many)
        self.assertIn("Java", result)
        self.assertIn("Javascript", result)
        self.assertEqual(len(result["Java"]), 1)
        self.assertEqual(result["Java"][0], "Guava")

    def test_second_task(self):
        result = second_task(self.one_to_many)
        self.assertEqual(result[0][0], "Python")
        self.assertEqual(result[0][1], 16000000)

    def test_third_task(self):
        result = third_task(self.many_to_many)
        self.assertEqual(result[0][2], "C/C++")
        self.assertEqual(result[-1][2], "Python")

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

Результат выполнения

```
mmmmm@LAPTOPIC:~/projects/pcpl_2025/rk_2$ python3 main.py
Задание Г1
{'Javascript': ['React'], 'Java': ['Guava']}

Задание Г2
[('Python', 16000000), ('Javascript', 6500000), ('Java', 4000000), ('C/C++', 20000)]

Задание Г3
[('NumPy', 16000000, 'C/C++'), ('Pandas', 13000000, 'C/C++'), ('SFML', 20000, 'C/C++'), ('Guava', 4000000, 'Java'), ('React', 6500000, 'Javascri
pt'), ('NumPy', 16000000, 'Python'), ('Pandas', 13000000, 'Python')]
mmmmm@LAPTOPIC:~/projects/pcpl_2025/rk_2$
```

```
mmmmm@LAPTOPIC:~/projects/pcpl_2025/rk_2$ python3 test_main.py
...
----------------------------------------------------------------------
Ran 3 tests in 0.000s

OK
mmmmm@LAPTOPIC:~/projects/pcpl_2025/rk_2$
```