



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «ГУИМЦ»
КАФЕДРА ИУ5 «Системы обработки информации и управления»

Дисциплина «Базовые компоненты интернет-технологий»
ОТЧЕТ

Рубежный контроль №2

Студент: Соловьева А.М., группа ИУ5Ц-53Б
Преподаватель: Гапанюк Ю.Е.

2021 г.

СОДЕРЖАНИЕ

1. Описание задания:.....	3
2. Листинг программы	3
3. Результат выполнения программы:	6

1. Описание задания:

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).

2. Листинг программы

main.py

```
from operator import itemgetter

class Chapter:
    """Глава"""

    def __init__(self, title, id, pages, book_id):
        self.title = title
        self.id = id
        self.pages = pages
        self.book_id = book_id

class Book:
    """Книга"""

    def __init__(self, title, id):
        self.title = title
        self.id = id

class ChapterBook:
    """
    'Главы книги' для реализации
    СВЯЗИ МНОГИЕ-КО-МНОГИМ
    """

    def __init__(self, chap_id, books_id):
        self.chap_id = chap_id
        self.books_id = books_id

# Книги
books = [
    Book('Аэропорт', 1),
    Book('Вино из одуванчиков', 2),
    Book('451 градус по Фаренгейту', 3),
    Book('1984', 4),
    Book('Алые паруса', 5),
    Book('Повелитель мух', 6),
]

# Главы
chapters = [
    Chapter('Глава 1', 1, 20, 1),
```

```

Chapter('Глава 2', 2, 12, 2),
Chapter('Глава 3', 3, 37, 3),
Chapter('Глава 4', 4, 24, 4),
Chapter('Глава 5', 5, 18, 5),
Chapter('Глава 6', 6, 40, 6),
]
# Главы книги, для связи многие-ко-многим
chapters_books = [
    ChapterBook(1, 1),
    ChapterBook(2, 5),
    ChapterBook(3, 2),
    ChapterBook(4, 4),
    ChapterBook(5, 3),
    ChapterBook(6, 6),

    ChapterBook(11, 5),
    ChapterBook(12, 4),
    ChapterBook(13, 3),
    ChapterBook(13, 2),
    ChapterBook(13, 1),
    ChapterBook(13, 3),
]

def task_1(one_to_many):
    res1 = []
    for title, _, book in one_to_many:
        if title[0] == "Г":
            res1.append((title, book))
    return res1

def task_2(one_to_many):
    res2 = []
    for b in books:
        b_chapters = list(filter(lambda x: x[2] == b.title, one_to_many))
        if len(b_chapters) > 0:
            b_pages = [pages for _, pages, _ in b_chapters]
            b_pages_min = min(b_pages)
            res2.append((b.title, b_pages_min))
    res2_sorted = sorted(res2, key=itemgetter(1))
    return res2_sorted

def task_3(many_to_many):
    res3 = []
    for title, _, book in many_to_many:
        res3.append((title, book))
    res3_sorted = sorted(res3, key=itemgetter(0))
    return res3_sorted

def main():
    """Основная функция"""
    # Соединение данных один-ко-многим
    one_to_many = [(c.title, c.pages, b.title)
                   for b in books
                   for c in chapters
                   if b.id == c.book_id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(b.title, cb.chap_id, cb.books_id)

```

```

        for b in books
        for cb in chapters_books
        if b.id == cb.books_id]

many_to_many = [(c.title, c.pages, b_title)
                 for b_title, chapters_id, b_id in many_to_many_temp
                 for c in chapters if c.id == chapters_id]

print('\nЗадание B1 \n', task_1(one_to_many))
print('\nЗадание B2 \n', task_2(one_to_many))
print('\nЗадание B3 \n', taks_3(many_to_many))

if __name__ == '__main__':
    main()

```

test.py

```

import unittest
import sys, os

sys.path.append(os.getcwd())
from main import *

class TddTest(unittest.TestCase):
    def test_1(self):
        one_to_many = [(c.title, c.pages, b.title)
                       for b in books
                       for c in chapters
                       if b.id == c.book_id]
        self.assertEqual(task_1(one_to_many), [('Глава 1', 'Аэропорт'),
        ('Глава 2', 'Вино из одуванчиков'),
        ('Глава 3', '451 градус по
Фаренгейту'), ('Глава 4', '1984'),
        ('Глава 5', 'Алые паруса'),
        ('Глава 6', 'Повелитель мух')])

    def test_2(self):
        one_to_many = [(c.title, c.pages, b.title)
                       for b in books
                       for c in chapters
                       if b.id == c.book_id]
        self.assertEqual(task_2(one_to_many), [('Вино из одуванчиков', 12),
        ('Алые паруса', 18),
        ('Аэропорт', 20), ('1984',
        24), ('451 градус по Фаренгейту', 37),
        ('Повелитель мух', 40)])

    def test_3(self):
        many_to_many_temp = [(b.title, cb.chap_id, cb.books_id)
                              for b in books
                              for cb in chapters_books
                              if b.id == cb.books_id]
        many_to_many = [(c.title, c.pages, b_title)
                       for b_title, chapters_id, b_id in many_to_many_temp
                       for c in chapters if c.id == chapters_id]
        self.assertEqual(taks_3(many_to_many), [('Глава 1', 'Аэропорт'),
        ('Глава 2', 'Алые паруса'),
        ('Глава 3', 'Вино из
одуванчиков'), ('Глава 4', '1984'),

```

```

('Глава 5', '451 градус по
Фаренгейту'), ('Глава 6', 'Повелитель мух']])

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

```

3. Результат выполнения программы:

```

C:\Users\sashu\PycharmProjects\pythonProject3\venv\Scripts\python.exe C:/Users/sashu/PycharmProjects/pythonProject3/main.py

Задание В1
[('Глава 1', 'Аэропорт'), ('Глава 2', 'Вино из одуванчиков'), ('Глава 3', '451 градус по Фаренгейту'), ('Глава 4', '1984'), ('Глава 5', 'Алые паруса'), ('Глава 6', 'Повелитель мух')]

Задание В2
[('Вино из одуванчиков', 12), ('Алые паруса', 18), ('Аэропорт', 20), ('1984', 24), ('451 градус по Фаренгейту', 37), ('Повелитель мух', 40)]

Задание В3
[('Глава 1', 'Аэропорт'), ('Глава 2', 'Алые паруса'), ('Глава 3', 'Вино из одуванчиков'), ('Глава 4', '1984'), ('Глава 5', '451 градус по Фаренгейту'), ('Глава 6', 'Повелитель мух')]

Process finished with exit code 0

```

```

C:/Users/sashu/PycharmProjects/pythonProject3/test.py
Testing started at 22:01 ...

Ran 3 tests in 0.002s

OK
Launching unittests with arguments python -m unittest C:/Users/sashu/PycharmProjects/pythonProject3/test.py in
C:\Users\sashu\PycharmProjects\pythonProject3

Process finished with exit code 0

```