**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»

Рубежный контроль №2

Вариант 12

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-34Б |  | Гапанюк Ю. Е. |
| Ромашко Дарья |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2021 г.

**Постановка задачи.**

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).

**Текст программы:**

**Файл language.py**

class language:

def \_\_init\_\_(self, id, title,year\_app, tool\_id):

self.id = id

self.title = title

self.year\_app = year\_app

self.tool\_id = tool\_id

**Файл mytool.py**

class tool:

def \_\_init\_\_(self, id, name):

self.id = id

self.name = name

**Файл LangTool.py**

class LangTool:

def \_\_init\_\_(self, tool\_id, lang\_id):

self.tool\_id = tool\_id

self.lang\_id = lang\_id

**Файл data.py**

from language import language

from mytool import tool

from LangTool import LangTool

#список средств разработки

tools = [tool(1,"Visual Studio"),

tool(2,"Visual Studio Code"),

tool(3, 'Android Studio')]

#список языков программирования

languages = [ language(1, 'Python', 1991, 1),

language(2, 'C++', 1983, 2),

language(3, 'Assembly', 1949, 3),

language(4, 'Pascal', 1970, 2),

language(5, 'Java', 1995, 1)]

#список связей многие-ко-многим

lang\_tool = [

LangTool(1, 1),

LangTool(1, 2),

LangTool(1, 3),

LangTool(2, 2),

LangTool(3, 4),

LangTool(3, 5),

]

#связь один-ко-многим

one\_to\_many = [(l.title, l.year\_app, t.name)

for t in tools

for l in languages

if l.tool\_id == t.id]

#связь многие-ко-многим

many\_to\_many = [(l.title, l.year\_app, t.name)

for t in tools

for l in languages

for relation in lang\_tool

if t.id == relation.tool\_id and l.id == relation.lang\_id]

**Файл tests.py**

from data import \*

from main import task1, task2, task3

import unittest

class Tests(unittest.TestCase):

def test\_task1(self):

self.assertEqual(task1(), [('Assembly', 1949, 'Android Studio')])

def test\_task2(self):

self.assertEqual(task2(), [('Visual Studio', 1995), ('Visual Studio Code', 1983), ('Android Studio', 1949)])

def test\_task3(self):

self.assertEqual(task3(), [('Pascal', 'Android Studio'), ('Java', 'Android Studio'), ('Python', 'Visual Studio'), ('C++', 'Visual Studio'), ('Assembly', 'Visual Studio'), ('C++', 'Visual Studio Code')])

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

**Файл main.py**

from operator import itemgetter

from data import\*

def task1():

task\_1=[]

for i in range(len(one\_to\_many)):

if ((one\_to\_many[i][2])[0] == 'A'):

task\_1.append(one\_to\_many[i])

return task\_1

def task2():

res\_2\_unsorted = []

# Перебираем все среды программирования

for t in tools:

# Список языков сред программирования

t\_lang = list(filter(lambda x: x[2] == t.name, one\_to\_many))

# Если среда программирования не пуста

if len(t\_lang) > 0:

res\_2\_unsorted.append((t.name, max(t\_lang, key=lambda x: x[1])[1]))

# Сортировка по новизне

res\_2 = sorted(res\_2\_unsorted, key=itemgetter(1), reverse=True)

return(res\_2)

def task3():

res\_3 = []

# Перебираем все среды программирования

for lang, \_, tool in many\_to\_many:

res\_3.append((lang, tool))

res\_3 = sorted(res\_3, key=itemgetter(1))

return(res\_3)

def main():

print('Задание Г1')

print(task1())

print('\nЗадание Г2')

print(task2())

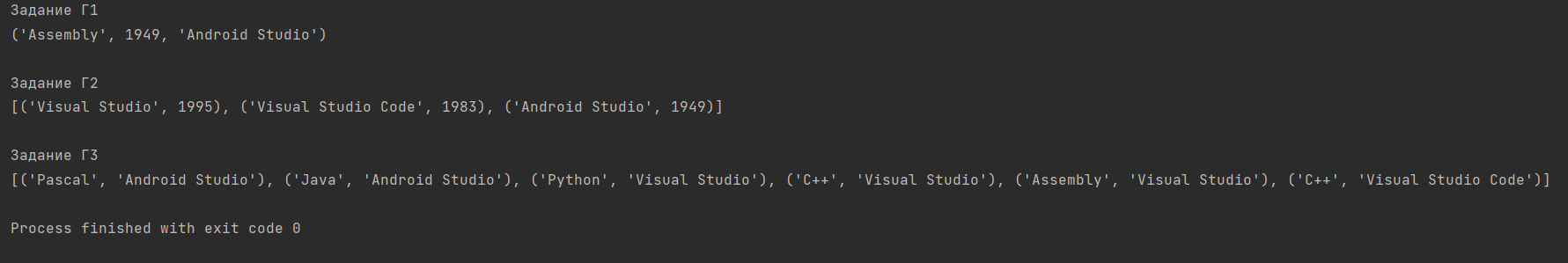
print('\nЗадание Г3')

print(task3())

if \_\_name\_\_ == '\_\_main\_\_':

main()

**Результат выполнения**

****

**Результат тестирования**

