**РК2 БКИТ Валиков РТ5-31Б**

Текст измененной программы РК1 допустимой для тестирования:

from operator import itemgetter

class file:

    """файл"""

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

        self.id = id

        self.name = name

        self.size = size

        self.catId = catId

class cat:

    """каталог"""

    def \_\_init\_\_(self, id, name):

        self.id = id

        self.name = name

class fileCat:

    """

    'файлы каталога' для реализации

    связи многие-ко-многим

    """

    def \_\_init\_\_(self, catId, fileId):

        self.catId = catId

        self.fileId = fileId

cats = [cat(1, "Программирование"),

        cat(2, "Проектирование"),

        cat(3, "Разработка")]

files = [file(1,"Студенческий",10000,1),

         file(2,"Преподавательский",5000,2),

         file(3,"Учебный",30000,1)]

filesCats = [fileCat(1,1),

             fileCat(2,2),

             fileCat(1,3),

             fileCat(3,3)]

catsId = [c.id for c in cats]

oneToMany = [(f.name, f.size, cats[catsId.index(f.catId)].name) for f in files]

filesId = [f.id for f in files]

manyToMany = [(files[filesId.index(fc.fileId)].name,

                files[filesId.index(fc.fileId)].size,

                cats[catsId.index(fc.catId)].name)

                for fc in filesCats]

def task1(oneToMany):

    word1 = "г"

    catsE1 = [c.name for c in cats if word1 in c.name]

    filesE1 = [otm[0] for otm in oneToMany for c in catsE1 if otm[2] == c]

    return catsE1, filesE1

def task2(oneToMany):

    return sorted([[c.name, round(sum([otm[1] for otm in oneToMany if otm[2] == c.name])/(lambda x: 1 if x==0 else x)(len([otm[1] for otm in oneToMany if otm[2] == c.name])))] for c in cats], key=itemgetter(1),reverse=True)

def task3(manyToMany):

    char3 = "g"

    return [[f.name,[mtm[2] for mtm in manyToMany if mtm[0]==f.name]] for f in files if f.name[0] == char3]

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

    print(task1(oneToMany))

    print(task2(oneToMany))

    print(task3(manyToMany))

Текст программы тестирования:

import unittest

import RK1

class testRK1(unittest.TestCase):

    def setUp(self):

        self.test1 = (['Программирование'], ['Студенческий', 'Учебный'])

        self.test2 = [['Программирование', 20000], ['Проектирование', 5000], ['Разработка', 0]]

        self.test3 = []

    def test1\_rk(self):

        self.assertEqual(RK1.task1(RK1.oneToMany), self.test1)

    def test2\_rk(self):

        self.assertEqual(RK1.task2(RK1.oneToMany), self.test2)

    def test3\_rk(self):

        self.assertEqual(RK1.task3(RK1.manyToMany), self.test3)

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

    unittest.main()

Результаты:

