Текст программы

bindings.py:

```
def get_one_to_many_bindings(files, filedirs):
    return [(f.name, f.size, fd.path) for f in files for fd in filedirs if f.filedir_id == fd.id]


def get_many_to_many_bindings(files, filedirs, filefiledirs):
    many_to_many_temp = [(fd.path, ffd.file_directory_id, ffd.file_id)
                         for fd in filedirs
                         for ffd in filefiledirs
                         if fd.id == ffd.file_directory_id]

    many_to_many = [(f.name, fd_path)
                    for fd_path, fd_id, f_id in many_to_many_temp
                    for f in files if f.id == f_id]

    return many_to_many
```

data.py:

```
from entities import File, FileDirectory, FileFileDirectory
from datetime import date


filedirs = [FileDirectory(1, '/root/PyCharm Projects/lab1'),
            FileDirectory(2, '/root/PyCharm Projects/lab2'),
            FileDirectory(3, '/root/PyCharm Projects/lab3'),
            FileDirectory(4, '/root/Desktop/')]

files = [File(1, 'lab2.py', date(2023, 9, 23), date(2023, 9, 23), 'rasulov1337', 1024, 2),
         File(2, 'lab1.py', date(2023, 10, 19), date(2023, 10, 19), 'rasulov1337', 10, 1),
         File(3, 'readme.txt', date(2023, 10, 1), date(2023, 10, 15), 'root', 10, 3),
         File(4, '.gitignore', date(2023, 10, 24), date(2023, 10, 24), 'adam', 1, 1),
         File(5, 'main.py', date(2023, 10, 19), date(2023, 10, 19), 'rasulov1337', 1, 2),
         File(6, 'wallpaper.png', date(2023, 10, 20), date(2023, 10, 20), 'root', 1024 * 7, 4)]

filefiledirs = [FileFileDirectory(2, 1),
                FileFileDirectory(1, 2),
                FileFileDirectory(3, 3),
                FileFileDirectory(4, 1),
                FileFileDirectory(4, 2),
                FileFileDirectory(4, 3),
                FileFileDirectory(5, 1),
                FileFileDirectory(6, 4)]
```

entities.py:

```
class File:
    def __init__(self, id, name, creation_date, modification_date, author, size, filedir_id):
        self.id = id
        self.name = name
        self.creation_date = creation_date
        self.modification_date = modification_date
        self.author = author
        self.size = size
        self.filedir_id = filedir_id

    def __repr__(self):
        return self.name


class FileDirectory:
    def __init__(self, id, path):
        self.id = id
        self.path = path

    def __repr__(self):
        return self.path


class FileFileDirectory:
    def __init__(self, file_id, file_directory_id):
        self.file_id = file_id
        self.file_directory_id = file_directory_id
```

main.py:

```
from data import files, filedirs, filefiledirs
from bindings import get_many_to_many_bindings, get_one_to_many_bindings


def get_files_that_ends_with(ending: str, one_to_many_bindings):
    return list(filter(lambda x: x[0].endswith(ending), one_to_many_bindings))


def get_filedirs_avg_file_size(one_to_many_bindings):
    res = []
    counted_fds = set()
    for filename, file_size, fd_path in one_to_many_bindings:
        if fd_path not in counted_fds:
            counted_fds.add(fd_path)
        else:
            continue

        file_sizes = [i[1] for i in one_to_many_bindings if i[2] == fd_path]
        res.append((fd_path, sum(file_sizes) / len(file_sizes)))
    return res


def get_filedirs_starts_with(beginning, many_to_many_bindings):
    filedirs = list(set(filedir_path for _, filedir_path in many_to_many_bindings))
    dict_ = {i: [] for i in filedirs}
    for filename, filedir_path in many_to_many_bindings:
        dict_[filedir_path].append(filename)

    res_3 = [(i, dict_[i]) for i in dict_]
    return list(filter(lambda x: x[0].startswith(beginning), res_3))


def taskD1(one_to_many_bindings):
    print("Task #D1")
    for i in get_files_that_ends_with('.py', one_to_many_bindings):
        print(i[0], i[2])


def taskD2(one_to_many_bindings):
    print("\nTask #D2")
    res = get_filedirs_avg_file_size(one_to_many_bindings)
    for i in sorted(res, key=lambda x: x[1]):
        print(i)


def taskD3(many_to_many_bindings):
    print("\nTask #D3")
    for i in get_filedirs_starts_with('/root/PyCharm Projects/', many_to_many_bindings):
        print(i)


def main():
    one_to_many = get_one_to_many_bindings(files, filedirs)
    many_to_many = get_many_to_many_bindings(files, filedirs, filefiledirs)
    taskD1(one_to_many)
    taskD2(one_to_many)
    taskD3(many_to_many)


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

test.py:

```
import unittest
import data
from bindings import get_one_to_many_bindings,
get_many_to_many_bindings
from main import get_filedirs_starts_with, get_filedirs_avg_file_size,
get_files_that_ends_with


class Test(unittest.TestCase):
    def setUp(self):
        self.data = data

    def test_one_to_many(self):
        desired = [('lab2.py', 1024, '/root/PyCharm Projects/lab2'),
                   ('lab1.py', 10, '/root/PyCharm Projects/lab1'),
                   ('readme.txt', 10, '/root/PyCharm Projects/lab3'),
                   ('.gitignore', 1, '/root/PyCharm Projects/lab1'),
                   ('main.py', 1, '/root/PyCharm Projects/lab2'),
                   ('wallpaper.png', 7168, '/root/Desktop/')]
        actual = get_one_to_many_bindings(self.data.files, self.data.filedirs)
        self.assertCountEqual(desired, actual)

    def test_many_to_many(self):
        actual = get_many_to_many_bindings(self.data.files, self.data.filedirs,
self.data.filefiledirs)
        desired = [('lab1.py', '/root/PyCharm Projects/lab1'),
                    ('.gitignore', '/root/PyCharm Projects/lab1'),
                    ('main.py', '/root/PyCharm Projects/lab1'),
                    ('lab2.py', '/root/PyCharm Projects/lab2'),
                    ('.gitignore', '/root/PyCharm Projects/lab2'),
                    ('readme.txt', '/root/PyCharm Projects/lab3'),
                ('.gitignore', '/root/PyCharm Projects/lab3'),
                ('wallpaper.png', '/root/Desktop/')]
        self.assertCountEqual(desired, actual)


    def test_get_files_that_ends_with(self):
        actual = get_files_that_ends_with('.py',
get_one_to_many_bindings(self.data.files, self.data.filedirs))
        desired = [('lab2.py', 1024, '/root/PyCharm Projects/lab2'), ('lab1.py', 10,
'/root/PyCharm Projects/lab1'), ('main.py', 1, '/root/PyCharm Projects/lab2')]
        self.assertCountEqual(actual, desired)

    def test_get_filedirs_avg_file_size(self):
        actual =
get_filedirs_avg_file_size(get_one_to_many_bindings(self.data.files,
self.data.filedirs))
        desired = [('/root/PyCharm Projects/lab2', 512.5), ('/root/PyCharm
Projects/lab1', 5.5), ('/root/PyCharm Projects/lab3', 10.0), ('/root/Desktop/',
7168.0)]
        self.assertCountEqual(actual, desired)

    def test_get_filedirs_starts_with(self):
        actual = get_filedirs_starts_with('/root/PyCharm Projects/',
                                          get_many_to_many_bindings(self.data.files,
self.data.filedirs, self.data.filefiledirs))
        desired = [('/root/PyCharm Projects/lab3', ['readme.txt', '.gitignore']),
('/root/PyCharm Projects/lab2', ['lab2.py', '.gitignore']), ('/root/PyCharm
Projects/lab1', ['lab1.py', '.gitignore', 'main.py'])]
        self.assertCountEqual(actual, desired)
```

Результаты выполнения

```
Testing started at 3:11 PM ...
Launching unittests with arguments python -m unittest
/home/arsen/PycharmProjects/pikyap/rk2/test.py in
/home/arsen/PycharmProjects/pikyap/rk2


Ran 5 tests in 0.001s

OK
Process finished with exit code 0
```