

## Код программы

- **RK1.py**

```
from operator import itemgetter

class House:
    def __init__(self, id, address, houseCreatingYear,
street_id):
        self.id = id
        self.address = address
        self.street_id = street_id
        self.houseCreatingYear = houseCreatingYear

class Street:
    def __init__(self, id, name):
        self.id = id
        self.name = name

class HousesOnStreet:
    def __init__(self, house_id, street_id):
        self.house_id = house_id
        self.street_id = street_id

streets = [
    Street(1, 'Улица Луговая'),
    Street(2, "Улица Гениальная"),
    Street(3, "Улица Кринжовая"),
    Street(4, 'Улица Герасимовская'),
    Street(5, 'Улица Маслянная'),
]

houses = [
    House(1, 'Улица Луговая, 3', 2000, 1),
    House(2, 'Улица Гениальная, 45', 2005, 2),
    House(3, 'Улица Кринжовая, 228', 2028, 3),
    House(4, 'Улица Кринжовая, 322', 1998, 3),
    House(5, 'Улица Луговая, 10', 1905, 1),
    House(6, 'Улица Герасимова, 205', 1998, 4),
    House(7, 'Улица Герасимова 303', 2017, 4),
    House(8, 'Улица Маслянная 76', 2023, 4),
]

houses_on_streets = [
    HousesOnStreet(1, 1),
    HousesOnStreet(2, 2),
    HousesOnStreet(3, 3),
    HousesOnStreet(4, 3),
    HousesOnStreet(5, 1),
    HousesOnStreet(6, 4),
    HousesOnStreet(7, 4),
    HousesOnStreet(8, 5),
]
```

```

def main():
    one_to_many = [(h.address, h.houseCreatingYear, s.name)
                    for s in streets
                    for h in houses
                    if h.street_id == s.id]
    many_to_many_temp = [(s.name, hos.street_id,
                           hos.house_id)
                          for s in streets
                          for hos in houses_on_streets
                          if s.id == hos.street_id]
    many_to_many = [(h.address, h.houseCreatingYear,
                      street_name)
                     for street_name, street_id, house_id in
many_to_many_temp
                     for h in houses if h.id == house_id]

    print('Задание E1')
    print(list(filter(lambda i: i[2].find('Луговая') != -1,
one_to_many)))

    print('Задание E2')
    res_12_unsorted = []
    for s in streets:
        street_house = list(filter(lambda i: i[2] ==
s.name, one_to_many))
        if len(street_house) > 0:
            s_year = [houseCreatingYear for _,
houseCreatingYear, _ in street_house]
            s_year_sum = round(sum(s_year) /
len(street_house), 2)
            res_12_unsorted.append((s.name, s_year_sum))

    res_12 = sorted(res_12_unsorted, key=itemgetter(1),
reverse=True)
    print(res_12)

    print('Задание E3')
    print(list(filter(lambda i: i[0].find('Г') != -1,
many_to_many)))

if __name__ == '__main__':
    main()

```

- **test.py**

```

import unittest
from RK1 import streets, houses, houses_on_streets, main

class TestRK1Functions(unittest.TestCase):
    def test_task_E1(self):
        expected_result = [

```

```

        ('Улица Луговая, 3', 2000, 'Улица Луговая'),
        ('Улица Луговая, 10', 1905, 'Улица Луговая')
    ]
    one_to_many = [(h.address, h.houseCreatingYear,
s.name)
                    for s in streets
                    for h in houses
                    if h.street_id == s.id]
    filtered_result = list(filter(lambda i:
i[2].find('Луговая') != -1, one_to_many))
    self.assertEqual(filtered_result, expected_result)

def test_task_E2(self):
    expected_result = [
        ('Улица Кринжовая', 2123.0),
        ('Улица Герасимовская', 2005.0),
        ('Улица Маслянная', 2023.0)
    ]
    one_to_many = [(h.address, h.houseCreatingYear,
s.name)
                    for s in streets
                    for h in houses
                    if h.street_id == s.id]
    res_12_unsorted = []
    for s in streets:
        street_house = list(filter(lambda i: i[2] ==
s.name, one_to_many))
        if len(street_house) > 0:
            s_year = [houseCreatingYear for _,
houseCreatingYear, _ in street_house]
            s_year_sum = round(sum(s_year) /
len(street_house), 2)
            res_12_unsorted.append((s.name,
s_year_sum))

    res_12 = sorted(res_12_unsorted, key=lambda x:
x[1], reverse=True)
    self.assertEqual(res_12, expected_result)

def test_task_E3(self):
    expected_result = [
        ('Улица Гениальная, 45', 2005, 'Улица
Гениальная'),
        ('Улица Герасимова, 205', 1998, 'Улица
Герасимовская'),
        ('Улица Герасимова 303', 2017, 'Улица
Герасимовская'),
        ('Улица Маслянная 76', 2023, 'Улица Маслянная')
    ]
    many_to_many_temp = [(s.name, hos.street_id,
hos.house_id)

```

```

        for s in streets
        for hos in houses_on_streets
        if s.id == hos.street_id]
    many_to_many = [(h.address, h.houseCreatingYear,
street_name)
                    for street_name, street_id,
house_id in many_to_many_temp
                    for h in houses if h.id ==
house_id]
    filtered_result = list(filter(lambda i:
i[0].find('Г') != -1, many_to_many))
    self.assertEqual(filtered_result, expected_result)

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

```

## Результат работы

Testing started at 2:21 ...

Launching unittests with arguments python -m unittest

test.TestRK1Functions in C:\Users\user\Desktop\Study\PARADIGM LABS  
3 SEM\RK2

```

[('Улица Кринжовая', 2123.0),
('Улица Герасимовская', 2005.0),
('Улица Маслянная', 2023.0)] != [('Улица Кринжовая', 2013.0),
('Улица Герасимовская', 2012.67),
('Улица Гениальная', 2005.0),
('Улица Луговая', 1952.5)]

```

<Click to see difference>

Traceback (most recent call last):

File "C:\Users\user\Desktop\Study\PARADIGM LABS 3  
SEM\RK2\test.py", line 38, in test\_task\_E2

self.assertEqual(res\_12, expected\_result)

AssertionError: Lists differ: [('Улица Кринжовая', 2013.0), ('Улица  
Герасимовская', 2012.67),[52 chars]2.5)] != [('Улица Кринжовая',  
2123.0), ('Улица Герасимовская', 2005.0), [23 chars]3.0)]

First differing element 0:

('Улица Кринжовая', 2013.0)

('Улица Кринжовая', 2123.0)

First list contains 1 additional elements.

First extra element 3:

('Улица Луговая', 1952.5)

- [('Улица Кринжовая', 2013.0),  
? -

+ [('Улица Кринжовая', 2123.0),  
? +

- ('Улица Герасимовская', 2012.67),  
? ^^ ^^

+ ('Улица Герасимовская', 2005.0),  
? ^^ ^

+ ('Улица Маслянная', 2023.0)]

- ('Улица Гениальная', 2005.0),

- ('Улица Луговая', 1952.5)]

[('Улица Гениальная', 45', 2005, 'Улица Гениальная'),  
(('Улица Герасимова', 205', 1998, 'Улица Герасимовская'),  
(('Улица Герасимова 303', 2017, 'Улица Герасимовская'),  
(('Улица Маслянная 76', 2023, 'Улица Маслянная'))] != [('Улица  
Гениальная', 45', 2005, 'Улица Гениальная'),  
(('Улица Герасимова', 205', 1998, 'Улица Герасимовская'),  
(('Улица Герасимова 303', 2017, 'Улица Герасимовская'))]

<Click to see difference>

Traceback (most recent call last):

File "C:\Users\user\Desktop\Study\PARADIGM LABS 3  
SEM\RK2\test.py", line 56, in test\_task\_E3

self.assertEqual(filtered\_result, expected\_result)

AssertionError: Lists differ: [('Ул[144 chars]расимовская')] != [('Ул[144  
chars]расимовская'), ('Улица Маслянная 76', 2023, 'Улица Маслянная')]

Second list contains 1 additional elements.

First extra element 3:

('Улица Маслянная 76', 2023, 'Улица Маслянная')

[('Улица Гениальная, 45', 2005, 'Улица Гениальная'),  
('Улица Герасимова, 205', 1998, 'Улица Герасимовская'),  
- ('Улица Герасимова 303', 2017, 'Улица Герасимовская')]  
? ^

+ ('Улица Герасимова 303', 2017, 'Улица Герасимовская'),  
? ^

+ ('Улица Маслянная 76', 2023, 'Улица Маслянная')]

Ran 3 tests in 0.015s

FAILED (failures=2)

Process finished with exit code 1