

**Московский государственный технический
университет им. Н.Э. Баумана**

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Парадигмы и конструкции языков программирования»

Отчет по РК №2

Выполнил:
студент группы ИУ5-34Б:

Шимко Даниил
Подпись и дата:

Проверил:
преподаватель каф.
ИУ5
Гапанюк Ю.
Подпись и дата:

Москва, 2023 г.

Задание:

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).

Код:

```
from operator import itemgetter

class Operator:
    def __init__(self, id, name, speed, pl_id):
        self.id = id
        self.name = name
        self.speed = speed
        self.pl_id = pl_id

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

class OperatorInProgrlang:
    """
    'Операторы' для реализации
    СВЯЗИ МНОГИЕ-КО-МНОГИМ
    """
    def __init__(self, pl_id, emp_id):
        self.pl_id = pl_id
        self.operator_id = emp_id

progrlangs = [Progrlang(1, 'Java'), Progrlang(2, 'JS'), Progrlang(3, 'Golang'),
               Progrlang(11, 'Php'), Progrlang(22, 'C++'), Progrlang(33, 'C#'), ]

operators = [Operator(1, '+', 0.1818, 1), Operator(2, 'x++', 0.17839, 2),
             Operator(3, 'SDK', 0.32021, 3), Operator(4, '/=', 0.25001, 3), Operator(5, '=', 0.0911, 3),]

goslings = [OperatorInProgrlang(1, 1), OperatorInProgrlang(2, 2),
            OperatorInProgrlang(3, 3), OperatorInProgrlang(3, 4), OperatorInProgrlang(3, 5),
            OperatorInProgrlang(11, 1), OperatorInProgrlang(22, 2),
            OperatorInProgrlang(33, 3), OperatorInProgrlang(33, 4),
            OperatorInProgrlang(33, 5),]
```

```

def ans_one(one_to_many_fq):
    print("First Question")
    sorted(one_to_many_fq, key=itemgetter(0))
    i = 0
    j = 0
    ans = []
    """Sliding windows"""
    while i < len(one_to_many_fq) and one_to_many_fq[i][0].startswith('J'):
        if i == j:
            ans.append(one_to_many_fq[j][0])
            while j < len(one_to_many_fq) and one_to_many_fq[j][0] ==
one_to_many_fq[i][0]:
                ans.append(one_to_many_fq[j][1] + ' ' + str(one_to_many_fq[j]
[2]))
                j += 1
            i = j
    return ans

def ans_two(one_to_many_fq):
    print("Second Question")
    sorted(one_to_many_fq, key=itemgetter(0, 2))
    i = 0
    j = 0
    parks_maximus = []
    ans = []
    """Sliding windows"""
    while i < len(one_to_many_fq):
        curr = 0
        while j < len(one_to_many_fq) and one_to_many_fq[j][0] ==
one_to_many_fq[i][0]:
            if one_to_many_fq[j][2] > curr:
                curr = one_to_many_fq[j][2]
            j += 1
        parks_maximus.append((one_to_many_fq[i][0], curr))
        i = j
    for e in parks_maximus:
        ans.append(e)
    return ans

def ans_three(many_to_many_ans):
    print("Third Question")
    sorted(many_to_many_ans, key=itemgetter(0, 1))
    ans = dict()
    i = 0
    j = 0
    while i < len(many_to_many_ans) and j < len(many_to_many_ans):
        ans[many_to_many_ans[i][0]] = []
        while j < len(many_to_many_ans) and many_to_many_ans[j][0] ==
many_to_many_ans[i][0]:
            ans[many_to_many_ans[i][0]].append(many_to_many_ans[j][1])
            j += 1
        i = j
    return ans

def main():
    # Соединение данных один-ко-многим
    one_to_many_fq = [(pl.name, operator.name, operator.speed)
                      for pl in progrlangs
                      for operator in operators
                      if pl.id == operator.pl_id]
    # Соединение данных один-ко-многим

```

```
one_to_many_curr = [(pl.name, dia.pl_id, dia.pl_id)
                    for pl in proglangs
                    for dia in goslings
                    if pl.id == dia.pl_id]

many_to_many_ans = [(pl_name, d.name)
                    for pl_name, pl_id, operator_id in one_to_many_curr
                    for d in operators if d.id == operator_id]

print(ans_one(one_to_many_fq))
print(ans_two(one_to_many_fq))
print(ans_three(many_to_many_ans))

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