



**Министерство науки и высшего образования  
Российской Федерации  
Федеральное государственное бюджетное  
образовательное учреждение  
высшего образования  
«Московский государственный технический  
университет  
имени Н.Э. Баумана  
(национальный исследовательский**

**Отчет по лабораторной работе № 1**

Выполнила:  
студентка группы ИУ5-31Б  
Саркисян С. З.

Москва, 2022

## Задание:

Разработать программу для решения **биквадратного уравнения**.

1. Программа должна быть разработана в виде консольного приложения на языке Python.
2. Программа осуществляет ввод с клавиатуры коэффициентов  $A$ ,  $B$ ,  $C$ , вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты  $A$ ,  $B$ ,  $C$  могут быть заданы в виде параметров командной строки ( **вариант задания параметров приведен в конце файла с примером кода** ). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. **Описание работы с параметрами командной строки**.
4. Если коэффициент  $A$ ,  $B$ ,  $C$  введен или задан в командной строке некорректно, то необходимо проигнорировать некорректное значение и вводить коэффициент повторно пока коэффициент не будет введен корректно. Корректно заданный коэффициент - это коэффициент, значение которого может быть без ошибок преобразовано в действительное число.

## Текст программы:

```
# используется для сортировки
from operator import itemgetter

class Cond:
    """Дирижер"""
    def __init__(self, id, fio, sal, orch_id):
        self.id = id
        self.fio = fio
        self.sal = sal
        self.orch_id = orch_id

class Orch:
    """Оркестр"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class CondOrch:
    """
    'Дирижеры оркестра' для реализации
```

```

СВЯЗИ МНОГИЕ-КО-МНОГИМ
"""
def __init__(self, orch_id, cond_id):
    self.orch_id = orch_id
    self.cond_id = cond_id

# Оркестры
orchs = [
    Orch(1, 'филармонический'),
    Orch(2, 'народный оркестр'),
    Orch(3, 'хоровой'),

    Orch(11, 'духовой оркестр'),
    Orch(22, 'джазисты'),
    Orch(33, 'симфонический'),
]

# Дирижеры
conds = [
    Cond(1, 'Сидоров', 25000, 1),
    Cond(2, 'Петров', 35000, 2),
    Cond(3, 'Иваненко', 45000, 3),
    Cond(4, 'Сорокина', 35000, 3),
    Cond(5, 'Иванин', 25000, 3),
]

conds_orchs = [
    CondOrch(1,1),
    CondOrch(2,2),
    CondOrch(3,3),
    CondOrch(3,4),
    CondOrch(3,5),
    CondOrch(11,1),
    CondOrch(22,2),
    CondOrch(33,3),
    CondOrch(33,4),
    CondOrch(33,5),
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(c.fio, c.sal, o.name)
                    for o in orchs
                    for c in conds
                    if c.orch_id == o.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(o.name, co.orch_id, co.cond_id)
                          for o in orchs
                          for co in conds_orchs
                          if o.id == co.orch_id]

    many_to_many = [(c.fio, c.sal, orch_name)
                    for orch_name, orch_id, cond_id in many_to_many_temp
                    for c in conds if c.id == cond_id]

    print('Задание A1')
    res_11 = sorted(one_to_many, key=itemgetter(2))

```

```

print(res_11)

print('\nЗадание A2')
res_12_unsorted = []
# Перебираем все оркестры
for o in orchs:
    # Список дирижеров оркестра
    o_conds = list(filter(lambda i: i[2] == o.name, one_to_many))
    # Если отдел не пустой
    if len(o_conds) > 0:
        # Зарплаты дирижеров оркестра
        o_sals = [sal for _, sal, _ in o_conds]
        # Суммарная зарплата сотрудников отдела
        o_sals_sum = sum(o_sals)
        res_12_unsorted.append((o.name, o_sals_sum))

# Сортировка по суммарной зарплате
res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)

print('\nЗадание A3')
res_13 = {}
# Перебираем все оркестры
for o in orchs:
    if 'оркестр' in o.name:
        # Список дирижеров оркестра
        o_conds = list(filter(lambda i: i[2]==o.name, many_to_many))
        # Только ФИО дирижеров
        o_conds_names = [x for x,_,_ in o_conds]
        # Добавляем результат в словарь
        # ключ - отдел, значение - список фамилий
        res_13[o.name] = o_conds_names

print(res_13)

if __name__ == '__main__':
    main()

```

## Экранные формы с примерами выполнения программы:

