

```
# используется для сортировки
from operator import itemgetter

class Brauser:
    """Браузер"""

    def __init__(self, id, name, share, brause_id):
        self.id = id
        self.name = name
        self.market_share = share
        self.brause_id = brause_id

class Dep:
    """Компьютер"""

    def __init__(self, id, model):
        self.id = id
        self.model = model

class Brauser_Dep:
    """
    'Браузеры компьютера' для реализации связи многие-ко-многим
    """

    def __init__(self, brause_id, emp_id):
        self.brause_id = brause_id
        self.emp_id = emp_id

# Компьютеры
deps = [
    Dep(1, 'MacBook Air'),
    Dep(2, 'MacBook Pro'),
    Dep(3, 'Xiaomi RedmiBook'),

    Dep(11, 'Asus ViviBook Pro'),
    Dep(22, 'Lenovo ThinkPad'),
    Dep(33, 'Honor 2020'),
]

# Браузеры
brausers = [
    Brauser(1, 'Google', 112.52, 11),
    Brauser(2, 'Opera', 21.1, 11),
    Brauser(3, 'Safari', 78.3, 33),
    Brauser(4, 'Yandex', 53.8, 33),
    Brauser(5, 'Firefox', 11.23, 2),
]
```

```

brausers_deps = [
    Brauser_Dep(1, 1),
    Brauser_Dep(1, 2),
    Brauser_Dep(1, 3),
    Brauser_Dep(3, 4),
    Brauser_Dep(2, 5),

    Brauser_Dep(11, 1),
    Brauser_Dep(22, 2),
    Brauser_Dep(33, 3),
    Brauser_Dep(33, 4),
    Brauser_Dep(33, 5),
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(e.name, e.market_share, d.model)
                    for d in deps
                    for e in brausers
                    if e.brause_id == d.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(d.model, ed.brause_id, ed.emp_id)
                          for d in deps
                          for ed in brausers_deps
                          if d.id == ed.brause_id]

    many_to_many = [(e.name, e.market_share, dep_name)
                    for dep_name, brause_id, emp_id in many_to_many_temp
                    for e in brausers if e.id == emp_id]

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

    print('\nЗадание A2')
    res_12_unsorted = []
    # Перебираем все компьютеры
    for i in deps:
        # Список браузеров, установленных компьютерами
        d_browsers = list(filter(lambda k: k[2] == i.model, one_to_many))
        # Если хотя бы один браузеров установлен
        if len(d_browsers) > 0:
            # Доли рынка каждого браузера
            braus_market_shares = [share for _, share, _ in d_browsers]
            # Общая доля рынка установленных браузеров
            braus_ms_sum = sum(braus_market_shares)
            res_12_unsorted.append((i.model, braus_ms_sum))

    # Сортировка по суммарной доле рынка

```

```

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

print('\nЗадание A3')
res_13 = {}
# Перебираем все компьютеры
for d in deps:
    if 'Mac' in d.model:
        # Список браузеров компьютеров
        d_browsers = list(filter(lambda i: i[2] == d.model, many_to_many))
        # Название браузеров
        d_browsers_names = [x for x, _, _ in d_browsers]
        # Добавляем результат в словарь
        # ключ - компьютер, значение - список названий браузеров
        res_13[d.model] = d_browsers_names

print(res_13)

if __name__ == '__main__':
    main()

```

Результат:

Задание A1

```
[('Google', 112.52, 'Asus Vivobook Pro'), ('Opera', 21.1, 'Asus Vivobook Pro'), ('Safari', 78.3, 'Honor 2020'), ('Yandex', 53.8, 'Honor 2020'), ('Firefox', 11.23, 'MacBook Pro')]
```

Задание A2

```
[('Asus Vivobook Pro', 133.62), ('Honor 2020', 132.1), ('MacBook Pro', 11.23)]
```

Задание A3

```
{'MacBook Air': ['Google', 'Opera', 'Safari'], 'MacBook Pro': ['Firefox']}
```

Process finished with exit code 0