

```
import random

class Entity:
    cnt = 1

    def __init__(self, title='Simple title', author='No author'):
        self.id = self.__class__.cnt
        self.title = title
        self.author = author
        self.__class__.cnt += 1

    def __del__(self):
        self.__class__.cnt -= 1

class Document(Entity):
    def __init__(self, section_id, title='Simple title', author='No author'):
        super().__init__(title, author)
        self.section_id = section_id

class Section(Entity):
    pass

sections = [
    Section('Desktop', 'Egor'),
    Section('Trash', 'Egor'),
    Section('Downloads', 'Egor'),
]

documents = [
    Document(random.choice(sections).id, 'PK-1', 'Egor'),
    Document(random.choice(sections).id, 'PK-2', 'Egor'),
    Document(random.choice(sections).id, 'Steam', 'Egor'),
]

result = {
    section.title: [doc.title for doc in documents if doc.section_id == section.id]
    for section in sections
}

print('Задание №1')
for section_title, docs in result.items():
```

```

print(f'Раздел: {section_title}')
for doc_title in docs:
    print(f' Документ: {doc_title}')

print("\nЗадание №2")
result = [(len(docs), section_title, docs) for section_title, docs in result.items()]
for data in sorted(result):
    count, section_title, docs = data
    print(f'Раздел: {section_title} (Размер раздела: {count})')
    for doc_title in docs:
        print(f' Документ: {doc_title}')

```

Вывод:

Задание №1

Раздел: Desktop

Раздел: Trash

Документ: PK-1

Документ: Steam

Раздел: Downloads

Документ: PK-2

Задание №2

Раздел: Desktop (Размер раздела: 0)

Раздел: Downloads (Размер раздела: 1)

Документ: PK-2

Раздел: Trash (Размер раздела: 2)

Документ: PK-1

Документ: Steam

```

import random

class Entity:
    cnt = 1

    def __init__(self, title='Simple title', author='No author'):
        self.id = self.cnt
        self.title = title
        self.author = author
        self.__class__.cnt += 1

    def __del__(self):
        self.__class__.cnt -= 1

class Document(Entity):
    pass

```

```

class Section(Entity):
    pass

class DocumentInSection:
    cnt = 1

    def __init__(self, section_id=None, document_id=None):
        self.id = self.cnt
        self.section_id = section_id
        self.document_id = document_id
        self.__class__.cnt += 1

    def __del__(self):
        self.__class__.cnt -= 1

sections = [
    Section('Desktop', 'Egor'),
    Section('Trash', 'Egor'),
    Section('Downloads', 'Egor'),
]

documents = [
    Document('PK-1', 'Egor'),
    Document('PK-2', 'Egor'),
    Document('Steam', 'Egor'),
]

# Да может появиться два одинаковых документа, но будем считать что это копии
documents_in_sections = [
    DocumentInSection(random.choice(sections).id, random.choice(documents).id),
    DocumentInSection(random.choice(sections).id, random.choice(documents).id),
    DocumentInSection(random.choice(sections).id, random.choice(documents).id),
]

key_word = 'D'
sec_dict = {
    sec.title: [d_in_s.document_id for d_in_s in documents_in_sections if d_in_s.section_id
== sec.id] for sec in sections if key_word in sec.title
}
result = {
    title: [doc.title for doc in documents if doc.id in data] for title, data in sec_dict.items()
}

```

```
print('Задание №3')
for section_title, docs in result.items():
    print(f'Раздел: {section_title}')
    for doc_title in docs:
        print(f'Документ: {doc_title}')
```

Вывод:

Задание №3

Раздел: Desktop

Документ: РК-1

Документ: РК-2

Раздел: Downloads

Документ: РК-2