*Федеральное государственное бюджетное образовательное учреждение* *высшего профессионального образования*

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления (ИУ5)

**Отчет**

**по рубежному контролю №1**

**Дисциплина: Разработка Интернет-Приложений**

Студент гр. ИУ5-53Б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Якубов А. Р.

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Гапанюк Ю.Е.

(Подпись, дата) (И.О. Фамилия)

Москва, 2020

1. **Задание**

**Вариант В.**

1. «Раздел» и «Документ» связаны соотношением один-ко-многим. Выведите список всех разделов, у которых название начинается с буквы «В» и названия их документов.
2. «Раздел» и «Документ» связаны соотношением один-ко-многим. Выведите список документов с минимальным количеством страниц разделе, отсортированный по минимальному количеству.
3. «Раздел» и «Документ» связаны соотношением многие-ко-многим. Выведите список всех связанных разделов и документов, отсортированных по разделам, сортировка по документам произвольная.

**2. Текст программы**

*# используется для сортировки*from operator import itemgetter  
  
class sec:  
 *"""Section"""* def \_\_init\_\_(self, id, title, str\_count, doc\_id):  
 self.id = id  
 self.title = title  
 self.sec\_id = doc\_id  
 self.str\_count = str\_count  
  
class doc:  
 *"""Document"""* def \_\_init\_\_(self, id, doc\_title):  
 self.id = id  
 self.doc\_title = doc\_title  
  
  
class docsec:  
 *"""  
 'Document section' для реализации  
 связи многие-ко-многим  
 """* def \_\_init\_\_(self, sec\_id, doc\_id):  
 self.sec\_id = sec\_id  
 self.doc\_id = doc\_id  
  
  
*# Documents*docs = [  
 doc(1, **'ВСАСОИУ\_лаб1\_метода'**),  
 doc(2, **'РиП\_Лаб1\_отчет'**),  
 doc(3, **'СиТ\_методичка\_по\_первой\_лабе'**),  
]  
  
*# Sections*sect = [  
 sec(1, **'Теоретическая часть'**, 50, 1),  
 sec(2, **'Задание'**, 2, 1),  
 sec(3, **'Задание'**, 2, 2),  
 sec(4, **'Код программы'**, 4, 2),  
 sec(5, **'Вывод'**, 1, 2),  
 sec(6, **'Теоретическая часть'**, 30, 3),  
 sec(7, **'Задание'**, 5, 3),  
  
  
]  
  
Document\_Section = [  
 docsec(1, 1),  
 docsec(2, 1),  
 docsec(3, 1),  
 docsec(4, 2),  
 docsec(5, 2),  
 docsec(6, 3),  
 docsec(7, 3),  
]  
  
  
def main():  
 *"""Основная функция"""* one\_to\_many = [(s.title, s.str\_count, d.doc\_title)  
 for d in docs  
 for s in sect  
 if s.sec\_id == d.id]  
  
many\_to\_many\_temp = [(d.doc\_title, ds.sec\_id, ds.doc\_id)  
 for d in docs  
 for ds in Document\_Section  
 if d.id == ds.sec\_id]  
  
 many\_to\_many = [(s.title, doc\_name)  
 for doc\_name, sec\_id, doc\_id in many\_to\_many\_temp  
 for s in sect if s.id == doc\_id]  
  
 print(**'Задание B1'**)  
 res\_11 = list(filter(lambda x: x[0].startswith(**'В'**), one\_to\_many))  
 print(res\_11)  
  
 print(**'**\n**Задание B2'**)  
 res\_12\_unsorted = []  
 for d in docs:  
 d\_sect = list(filter(lambda i: i[2] == d.doc\_title, one\_to\_many))  
 if len(d\_sect) > 0:  
 count = [str\_count for \_, str\_count, \_ in d\_sect]  
 count\_min = min(count)  
 res\_12\_unsorted.append((d.doc\_title, count\_min))

res\_12 = sorted(res\_12\_unsorted, key=itemgetter(1), reverse=False)  
 print(res\_12)  
  
 print(**'**\n**Задание B3'**)  
 res\_13 = sorted(many\_to\_many, key=itemgetter(0))  
 print(res\_13)  
  
  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

1. **Вывод**

