Московский государственный технический университет имени Н.Э.Баумана

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Факультет РТ Радиотехнический

Кафедра ИУ5 Системы обработки информации и управления

**Отчет по рубежному контролю № 1 по курсу**

**Базовые компоненты интернет-технологий**

**Вариант 23**

|  |  |  |
| --- | --- | --- |
| Исполнитель |  |  |
| студент группы РТ5-31б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Яковенко Ю.С. |
|  |  | “\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2021 г. |
|  |  |  |
| Проверил |  |  |
| Доцент кафедры ИУ5 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Гапанюк Ю.Е. |
|  |  | “\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2021 г. |

Москва -  2021

**Описание задания**

1. Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.
2. Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.
3. **Вариант Е.**
   1. «Язык программирования» и «Синтаксическая конструкция» связаны соотношением один-ко-многим. Выведите список всех конструкций, у которых в названии присутствует слово «while», и список языков программирования, где есть данная конструкция.
   2. «Язык программирования» и «Синтаксическая конструкция» связаны соотношением один-ко-многим. Выведите список языков программирования со средней частотой встречаемости данных конструкций, отсортированный по средней частоте встречаемости.
   3. «Язык программирования» и «Синтаксическая конструкция» связаны соотношением многие-ко-многим. Выведите список всех конструкций, у которых частота встречаемости больше 25, и названия языков программирование, в которых они встречаются.

**Текст программы**

class SyncCons:

"""Синтаксическая конструкция"""

def \_\_init\_\_(self, id, name, freq, pl\_id):

"""

Args:

id (int): id синтаксической конструкции

name (str): название конструкции

freq (int): средняя частота использование в программе

pl\_id (int): id языка программирования (programming languange)

"""

self.id = id

self.name = name

self.freq = freq

self.pl\_id = pl\_id

class PL:

"""ЯП"""

def \_\_init\_\_(self, id, name):

"""

Args:

id (int): id ЯП

name (str): название ЯП

"""

self.id = id

self.name = name

class ConsLp:

"""

Конструкции языка (для реализации связи многие-ко-многим)

"""

def \_\_init\_\_(self, cons\_id, pl\_id):

self.cons\_id = cons\_id

self.pl\_id = pl\_id

# Языки программирования

pls = [

PL(1, 'Java'),

PL(2, 'JS'),

PL(3, 'Python'),

PL(4, 'C++')

]

# Синтаксические конструкции

# Сложно выделить конструкцию, которая есть только в одном языке

cons = [

SyncCons(1, 'for', 10, 1),

SyncCons(2, 'while', 5, 3),

SyncCons(3, 'do while', 3, 4),

SyncCons(4, 'switch', 7, 4),

SyncCons(5, 'class', 15, 4),

SyncCons(6, 'function', 20, 2)

]

cons\_lps = [

ConsLp(1,1),

ConsLp(2,1), # конструкции в языке Java

ConsLp(1,2),

ConsLp(5,2), # конструкции в языке JS

ConsLp(6,2),

ConsLp(1,3),

ConsLp(6,3), # конструкции в языке Python

ConsLp(4,4),

ConsLp(3,4), # конструкции в языке C++

]

def main():

# Соединение данных один-ко-многим

one\_to\_many = [(c.name, c.freq, p.name)

for c in cons

for p in pls

if c.pl\_id == p.id]

# Соединение данных многие-ко-многим

many\_to\_many\_temp = [(pl.name, cl.pl\_id, cl.cons\_id)

for pl in pls

for cl in cons\_lps

if pl.id == cl.pl\_id]

many\_to\_many = [(c.name, c.freq, pl\_name)

for pl\_name, pl\_id, cons\_id in many\_to\_many\_temp

for c in cons if c.id == cons\_id]

# Задание Е1

"""

«ЯП» и «Синтаксическая конструкция» связаны соотношением один-ко-многим.

Выведите список всех конструкций, у которых в названии присутствует слово while, и список языков программирования, где есть

данная конструкция.

"""

print("Задание Е1")

res1 = list(filter(lambda i: "while" in i[0], one\_to\_many))

print(res1)

# Задание Е2

"""

«ЯП» и «Синтаксическая конструкция» связаны соотношением один-ко-многим.

Выведите список языков программирования со средней частотой встречаемости данных конструкций, отсортированный по средней частоте встречаемости.

"""

print("Задание Е2")

dictOfCons = {}

for item in one\_to\_many: # создание словаря (язык программирования : список с частотой встречаемости всех конструкций в этом языке)

if (item[2] in dictOfCons):

dictOfCons[item[2]].append(item[1])

else:

dictOfCons[item[2]] = [item[1]]

res2 = sorted([(key, round(sum(dictOfCons[key]) / len(dictOfCons[key]), 2))

for key in dictOfCons.keys()], key = lambda x: x[1])

print(res2)

"""

«ЯП» и «Синтаксическая конструкция» связаны соотношением многие-ко-многим.

Выведите список всех конструкций, у которых частота встречаемости больше 25, и названия языков программирование, в которых они встречаются.

"""

print("Задание Е3")

d = {}

"""

создание словаря

конструкция языка: {

общая частота встречаемости во всех языках программирования : (int),

список с языками программирования, где конструкция встречается : ([str])

}

"""

for item in many\_to\_many:

if (item[0] in d):

d[item[0]]["number"] += item[1]

d[item[0]]["PLs"].append(item[2])

else:

d[item[0]] = {

"number": item[1],

"PLs": [item[2]]

}

res3 = [(key, d[key]["PLs"])

for key in d.keys()

if d[key]["number"] > 25

]

print(res3)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Примеры выполнения программы**

