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.**

**“Изучение объектно-ориентированных возможностей языка Python.”**

6

(количество листов)

Вариант № 1

| Исполнитель |  |  |
| --- | --- | --- |
| студент группы РТ5-21 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Аюшиев Т.О. |
|  |  | “27” октября 2022 г. |
|  |  |  |
| Проверил |  |  |
| Доцент кафедры ИУ5 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Гапанюк Ю.Е. |
|  |  | “27” октября 2022 г. |

# **Вариант Е. Предметная область 1.**

1. “Студент” и “Группа” связаны отношением один-ко-многим. Выведите список всех групп, у которых в названии присутствует слово “Group”, и список студентов в них.
2. “Студент” и “Группа” связаны отношением один-ко-многим. Выведите список всех групп со средней зарплатой студентов в каждой группе, отсортированный по среднему рейтингу. Средний рейтинг должен быть округлен до 2 знаков после запятой.
3. “Студент” и “Группа” связаны отношением многие-ко-многим. Выведите список всех студентов, у которых фамилия начинается с буквы “A”, и список группы, в которой они учатся.

# **Листинг программы.**

from operator import itemgetter

class Student:

def \_\_init\_\_(self, id, name, group\_id, rating): # Constructor

self.id = id

self.name = name

self.group\_id = group\_id

self.rating = rating # 0 - 100

class Group:

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

self.name = name

self.id = id

# Многие-ко-многим

class StudentGroup:

def \_\_init\_\_(self, group\_id, student\_id): # Constructor

self.group\_id = group\_id

self.student\_id = student\_id

# Students

student = [

Student(1, "Almaev", 1, 54.96),

Student(2, "Petrov", 2, 73.34),

Student(3, "Ayushiev", 3, 99.12),

Student(4, "Vasilev", 4, 12.34),

Student(5, "Akhmetzyanov", 5, 3.14),

]

#Groups

group = [

Group("Group 1", 1),

Group("RT-31B", 2),

Group("Group 3", 3),

]

# Students and Groups

student\_group = [

StudentGroup(1, 1),

StudentGroup(1, 2),

StudentGroup(2, 3),

StudentGroup(2, 4),

StudentGroup(3, 5),

]

# Student и Group свзяаны соотношением один-ко-многим.

# Выведите список всех групп, у которых в названии присутствует

# слово "Group", и список студентов в них.

def task1():

result = []

for i in group:

mid\_result = []

if "Group" in i.name:

mid\_result.append(i.name)

for j in student\_group:

if j.group\_id == i.id:

for k in student:

if k.id == j.student\_id:

mid\_result.append(k.name)

result.append(mid\_result)

return result

# Student и Group связаны соотношением один-ко-многим.

# Выведите список всех групп со средней зарплатой студентов в каждой группе,

# отстортированный по среднему рейтингу. Средний рейтинг должен быть округлен до 2 знаков после запятой

def task2():

result = []

for i in group:

sum = 0

count = 0

for j in student\_group:

if j.group\_id == i.id:

for k in student:

if k.id == j.student\_id:

sum += k.rating

count += 1

result.append((i.name, round(sum / count, 2)))

return sorted(result, key=itemgetter(1), reverse=True)

# Студент и группа связаны отношением многие-ко-многим.

# Выведите список всех студентов, у которых фамилия начинается с буквы "A",

# и список группы, в которых они учатся.

def task3():

result = []

for i in student:

if i.name[0] == "A":

for j in student\_group:

if i.id == j.student\_id:

for k in group:

if j.group\_id == k.id:

result.append((i.name, k.name))

return result

def main():

print("\nTask 1:")

print(task1())

print("\nTask 2:")

print(task2())

print("\nTask 3:")

print(task3())

if \_\_name\_\_ == '\_\_main\_\_':

main()

# **Результаты работы программы.**

