|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**  **«Московский государственный технический университет имени Н.Э. Баумана**  **(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ                              Информатика и системы управления

КАФЕДРА                  Системы обработки информации и управления

**Рубежный контроль №1**

**По курсу «Разработка интернет-приложений»**

Подготовила:

Студентка группы ИУ5-55Б.

Очеретная С.В.

24.10.2021

Проверил:

Гапанюк Ю.Е.

Москва, 2021 г

# Вариант предметной области: 9

Класс 1: Операционная система

Класс 2: Компьютер

# Вариант запросов: Д

1. «Операционная система» и «Компьютер» связаны соотношением один-ко-многим.

Выведите список всех операционных систем, у которых название хранимой записи содержит "Windows", и названия компьютеров с этими ОС.

1. «Операционная система» и «Компьютер» связаны соотношением один-ко-многим.

Выведите список компьютеров со средней датой публикации операционных систем в каждом компьютере, отсортированный по средней дате публикации.

(отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений).

1. «Операционная система» и «Компьютер» связаны соотношением многие-ко-многим.

Выведите список всех компьютеров, у которых название начинается с буквы «А», и список их операционных систем.

# Модели:

**Операционная система:**

class OperatingSystem:

    """Операционная система"""

    def \_\_init\_\_(self, id, name, publication\_year, computer\_id):

        self.id = id

        self.name = name

        self.publication\_year = publication\_year

        self.computer\_id = computer\_id

**Компьютер**

class Computer:

    """Компьютер"""

    def \_\_init\_\_(self, id, name):

        self.id = id

        self.name = name

**Компьютеры с операционными системами:**

class OperatingSystemComputer:

    """

    Операционные системы компьютеров

    для реализации связи многие-ко-многим

    """

    def \_\_init\_\_(self, computer\_id, operation\_system\_id):

        self.operation\_system\_id = operation\_system\_id

        self.computer\_id = computer\_id

# База данных:

**Операционные системы:**

operating\_systems = [

    OperatingSystem(1, 'Linux', 1991, 7),

    OperatingSystem(2, 'Windows 7', 2009, 5),

    OperatingSystem(3, 'Windows 10', 2015, 1),

    OperatingSystem(11, 'MacOS', 1984, 10),

    OperatingSystem(22, 'FreeDOS', 1994, 12)

]

**Компьютеры:**

computers = [

    Computer(1, 'Lenovo IdeaCentre 3 07ADA05'),

    Computer(2, 'HyperPC Lumen 3070Ti i5'),

    Computer(3, 'iRU Home 315'),

    Computer(4, 'Intel NUC Frost Canyon'),

    Computer(5, 'iRU Home 310 298386'),

    Computer(6, 'Dell Vostro 3267-5090'),

    Computer(7, 'Dell Optiplex MT 3070-7681'),

    Computer(8, 'Acer Nitro N50-600'),

    Computer(9, 'Acer Predator PO3-600'),

    Computer(10, 'Apple Mac mini i7'),

    Computer(11, 'Apple Mac Pro Intel Xeon 8'),

    Computer(12, 'HP Pavilion Gaming TG01-2011ur')

]

**Компьютеры с операционными системами:**

computers\_with\_operation\_systems = [

    OperatingSystemComputer(1, 3),

    OperatingSystemComputer(1, 2),

    OperatingSystemComputer(1, 1),

    OperatingSystemComputer(2, 3),

    OperatingSystemComputer(3, 1),

    OperatingSystemComputer(3, 22),

    OperatingSystemComputer(4, 3),

    OperatingSystemComputer(5, 2),

    OperatingSystemComputer(6, 2),

    OperatingSystemComputer(7, 1),

    OperatingSystemComputer(8, 1),

    OperatingSystemComputer(9, 1),

    OperatingSystemComputer(9, 2),

    OperatingSystemComputer(9, 3),

    OperatingSystemComputer(9, 11),

    OperatingSystemComputer(10, 11),

    OperatingSystemComputer(11, 11),

    OperatingSystemComputer(11, 12),

    OperatingSystemComputer(12, 1),

    OperatingSystemComputer(12, 2),

    OperatingSystemComputer(12, 3),

    OperatingSystemComputer(12, 11),

    OperatingSystemComputer(12, 22),

]

**Выполнение задания:**

Для заданий Д-1, Д-2:

# Соединение данных один-ко-многим

  operating\_systems\_join\_computers = [{'operating\_systems': o, 'computers': c}

    for o in operating\_systems

    for c in computers

    if o.computer\_id == c.id

  ]

**Выполнение задания Д-1**

Выведем id, name, publication\_year таблицы "Операционная система"

При этом name != Windows

И выведем компьютеры этих ОС

  D1 = [(x['operating\_systems'].id, x['operating\_systems'].name, x['operating\_systems'].publication\_year, x['computers'].name)

    for x in operating\_systems\_join\_computers

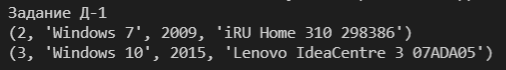
    if x['operating\_systems'].name.find('Windows') != - 1

  ]

  for x in D1:

    print(x)

**Принтскрин с результатом:**



**Выполнение задания Д-2**

Выведем имя компьютера, среднее по дате публикации ОС этого компьютера

Сортируя по этому среднему

  # Заведем таблицу с накапливаемой суммой дат и кол-вом ОС:

  computer\_sum\_count\_dict = {}

  for os\_computers\_row in operating\_systems\_join\_computers:

    computer\_name = os\_computers\_row['computers'].name

    publication\_year = os\_computers\_row['operating\_systems'].publication\_year

    if computer\_name in computer\_sum\_count\_dict:

      computer\_sum\_count\_dict[computer\_name]['sum'] = computer\_sum\_count\_dict[computer\_name]['sum'] + publication\_year

      computer\_sum\_count\_dict[computer\_name]['count'] = computer\_sum\_count\_dict[computer\_name]['count'] + 1

    else:

      computer\_sum\_count\_dict[computer\_name] = {'sum': publication\_year, 'count': 1}

  D2 = sorted(

    [(computer\_name, computer\_sum\_count\_dict[computer\_name]['sum'] / computer\_sum\_count\_dict[computer\_name]['count'])

      for computer\_name in computer\_sum\_count\_dict

      if computer\_sum\_count\_dict[computer\_name]['count'] != 0

    ],

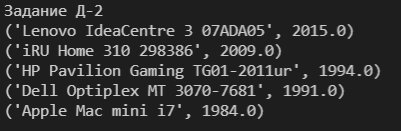
    key=itemgetter(1), reverse=True

  )

  for x in D2:

    print(x)

**Принтскрин с результатом:**



**Выполнение задания Д-3**

Соединение данных многие-ко-многим

  many\_to\_many = [(c.name, co.computer\_id, co.operation\_system\_id)

    for c in computers

        for co in computers\_with\_operation\_systems

            if c.id == co.computer\_id]

  computers\_with\_operation\_systems\_table = [(operating\_system.name, operating\_system.publication\_year, computer\_name)

    for computer\_name, computer\_id, operating\_system\_id in many\_to\_many

        for operating\_system in operating\_systems if operating\_system.id == operating\_system\_id]

  D3 = {}

  for computer in computers:

    if computer.name.startswith('A'):

        operating\_systems\_of\_computer = list(filter(lambda i: i[2] == computer.name, computers\_with\_operation\_systems\_table))

        operating\_systems\_names = [x for x, \_, \_ in operating\_systems\_of\_computer]

        D3[computer.name] = operating\_systems\_names

  print(D3)

**Принтскрин с результатом:**

