**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования »

Рубежный контроль №1

Вариант Е10

|  |  |
| --- | --- |
| Выполнил: |  |
| студент группы  РТ5-31Б |  |
| Ермаков И.А. |  |
|  |  |

Москва, 2024 г

**Текст программы**

# используется для сортировки

from operator import itemgetter

class Computer:

def \_\_init\_\_(self, computer\_id, owner\_name, processing\_power, browser\_id):

self.computer\_id = computer\_id

self.owner\_name = owner\_name

self.processing\_power = processing\_power

self.browser\_id = browser\_id

class Browser:

def \_\_init\_\_(self, browser\_id, name):

self.browser\_id = browser\_id

self.name = name

class ComputerBrowser:

def \_\_init\_\_(self, computer\_id, browser\_id):

self.computer\_id = computer\_id

self.browser\_id = browser\_id

# Список браузеров

browsers = [

Browser(1, 'Интернет-браузер Chrome'),

Browser(2, 'Браузер безопасности Firefox'),

Browser(3, 'Мобильный браузер Safari'),

]

# Список компьютеров

computers = [

Computer(1, 'Андреев', 3.2, 1),

Computer(2, 'Петров', 4.1, 2),

Computer(3, 'Антонов', 2.9, 1),

Computer(4, 'Иванов', 4.0, 3),

Computer(5, 'Смирнов', 3.0, 1),

]

# Связи компьютеров и браузеров для связи многие-ко-многим

computer\_browsers = [

ComputerBrowser(1, 1),

ComputerBrowser(2, 2),

ComputerBrowser(3, 1),

ComputerBrowser(4, 3),

ComputerBrowser(5, 1),

]

def main():

# Соединение данных один-ко-многим: браузер и компьютеры, использующие его

one\_to\_many = [

(c.owner\_name, c.processing\_power, b.name)

for b in browsers

for c in computers

if c.browser\_id == b.browser\_id

]

# Соединение данных многие-ко-многим: компьютер и браузеры, используемые компьютером

many\_to\_many\_temp = [

(b.name, cb.browser\_id, cb.computer\_id)

for b in browsers

for cb in computer\_browsers

if b.browser\_id == cb.browser\_id

]

# Завершаем связь многие-ко-многим

many\_to\_many = [

(c.owner\_name, c.processing\_power, browser\_name)

for browser\_name, browser\_id, computer\_id in many\_to\_many\_temp

for c in computers if c.computer\_id == computer\_id

]

# Задание 1: Список браузеров с "браузер" в названии и компьютеры, использующие их

print('Задание E1: Список браузеров с "браузер" в названии и компьютеры')

for browser in browsers:

if "браузер" in browser.name.lower():

print(f"Браузер: {browser.name}")

associated\_computers = [c.owner\_name for c in computers if c.browser\_id == browser.browser\_id]

print("Компьютеры:")

for computer in associated\_computers:

print(f" - {computer}")

print() # пустая строка для разделения

# Задание 2: Список браузеров со средней мощностью компьютеров, отсортированный по средней мощности

print('Задание E2: Браузеры со средней мощностью компьютеров')

average\_processing\_by\_browser = [

(browser.name, round(

sum(c.processing\_power for c in computers if c.browser\_id == browser.browser\_id) /

len([c for c in computers if c.browser\_id == browser.browser\_id]), 2

))

for browser in browsers if any(c.browser\_id == browser.browser\_id for c in computers)

]

average\_processing\_by\_browser.sort(key=lambda x: x[1])

for browser, avg\_power in average\_processing\_by\_browser:

print(f"Браузер: {browser}, Средняя мощность: {avg\_power}")

print()

# Задание 3: Список компьютеров с владельцами на "А" и названия их браузеров

print('Задание E3: Компьютеры с владельцами на "А" и их браузеры')

for c in computers:

if c.owner\_name.startswith("А"):

used\_browsers = [b.name for cb in computer\_browsers for b in browsers if cb.computer\_id == c.computer\_id and cb.browser\_id == b.browser\_id]

print(f"Владелец: {c.owner\_name}")

print("Браузеры:")

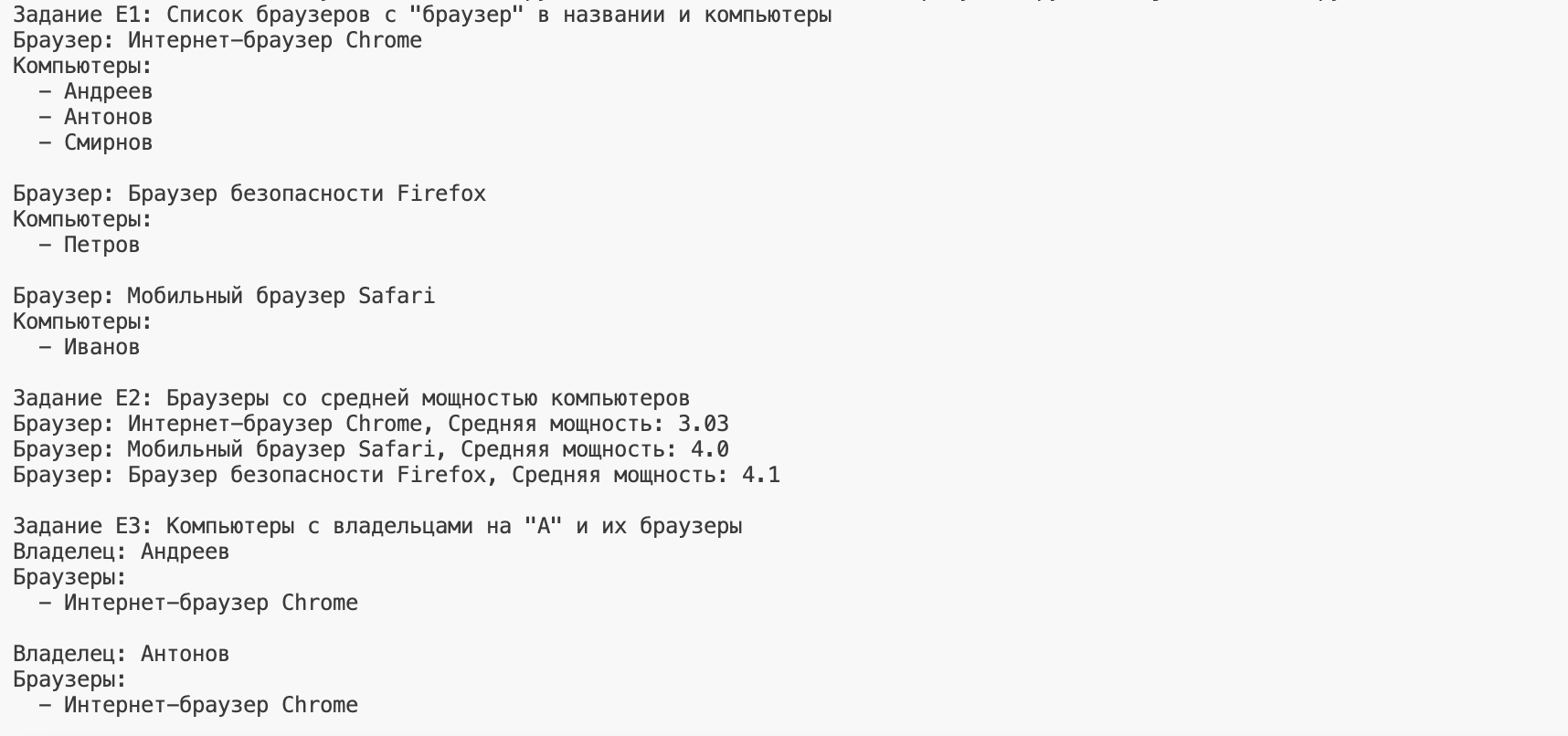
for browser in used\_browsers:

print(f" - {browser}")

print() # пустая строка для разделения

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

main()

**Результат программы:**