**Гаврилов Леонид ИУ5-53 Вариант В, 4.**

1. «Компьютер» и «Программа» связаны соотношением один-ко-многим. Выведите список всех отделов компьютеров, у которых название начинается с буквы «А», и названия их программ.
2. «Компьютер» и «Программа» связаны соотношением один-ко-многим. Выведите список компьютеров с минимальной ценой ПО в каждом компьютере, отсортированный по минимальной цене ПО.
3. «Компьютер» и «Программа» связаны соотношением многие-ко-многим. Выведите список всех связанных компьютеров и программ, отсортированный по компьютерам, сортировка по программам произвольная.

**#Код**

class Program:

  """Программа"""

  def \_\_init\_\_(self, id, name, license, price, computers):

    self.id = id

    self.name = name

    self.license = license

    self.price = price

    self.computers = computers

  def \_\_repr\_\_(self):

    return "Program {{ id: {}, name: \"{}\", license: \"{}\", price: {}, computers: {} }}".format(

      self.id, self.name,

      self.license, self.price,

      self.computers,

    )

class Computer:

  """Компьютер"""

  def \_\_init\_\_(self, id, ItName):

    self.id = id

    self.ItName = ItName

  def \_\_repr\_\_(self):

    return "Computer {{ id: {}, ItName: \"{}\" }}".format(self.id, self.ItName)

class ComputerPrograms:

  """'Программы компьютера' для реализации связи многие-ко-многим"""

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

    self.computer\_id = computer\_id

    self.program\_id = program\_id

  def \_\_repr\_\_(self):

    return "ComputerPrograms {{ computer\_id: {}, program\_id: {} }}".format(

      self.computer\_id, self.program\_id

    )

if \_\_name\_\_ == "\_\_main\_\_":

  # Компьютеры

  computers = [

    Computer(0, "Zotac"),

    Computer(1, "Asus"),

    Computer(2, "Gansor"),

    Computer(3, "Lenovo"),

    Computer(4, "Samsung"),

    Computer(5, "Aopen"),

  ]

  # Программы

  programs = [

    Program(0, "Microsoft Windows 10 Pro", "Проприетарная", 12\_000, [0, 1, 2]),

    Program(1, "Gentoo GNU/Linux", "Свободное программное обеспечение", 0, [3, 4, 5]),

    Program(2, "WinRAR", "Проприетарная", 1\_805.51, [0]),

    Program(3, "7zip", "Свободное программное обеспечение (LGPLv2.1+)", 0, [1, 2, 3, 4, 5]),

    Program(4, "Vim", "Свободное программное обеспечение (Vim license)", 0, [2, 3, 4]),

    Program(5, "Emacs", "Свободное программное обеспечение (GPLv3)", 0, [4]),

    Program(6, "Sublime Text 3", "Проприетарная", 6\_139.44, [0, 4]),

    Program(7, "VSCode", "MIT", 0, [1, 5]),

  ]

  computer\_programs = [

    # Zotac

    ComputerPrograms(0, 0), # Windows 10

    ComputerPrograms(0, 2), # WinRAR

    ComputerPrograms(0, 6), # Sublime

    # Asus

    ComputerPrograms(1, 0), # Windows 10

    ComputerPrograms(1, 3), # 7zip

    ComputerPrograms(1, 7), # VSCode

    # Gansor

    ComputerPrograms(2, 0), # Windows 10

    ComputerPrograms(2, 3), # 7zip

    ComputerPrograms(2, 4), # Vim

    # Lenovo

    ComputerPrograms(3, 1), # Gentoo

    ComputerPrograms(3, 3), # 7zip

    ComputerPrograms(3, 4), # Vim

    # Samsung

    ComputerPrograms(4, 1), # Gentoo

    ComputerPrograms(4, 3), # 7zip

    ComputerPrograms(4, 4), # Vim

    ComputerPrograms(4, 5), # Emacs

    ComputerPrograms(4, 6), # Sublime

    # Aopen

    ComputerPrograms(5, 1), # Gentoo

    ComputerPrograms(5, 3), # 7zip

    ComputerPrograms(5, 7), # VSCode

  ]

  # ================================

  print("Задание A1")

  print(

    [

      (

        c.ItName,

        [

          (p.name, p.license, p.price)

          for p in programs

          if c.id in p.computers

        ],

      )

      for c in computers

      if c.ItName[0].lower() == 'a'

    ]

  )

  print()

  print("Задание A2")

  print(

    sorted(

      [

        (

          c.ItName,

          sum(

            p.price

            for p in programs

            if c.id in p.computers

          ),

        )

        for c in computers

      ],

      key=lambda x: x[1],

      reverse=False,

    )

  )

  print()

  # Схема: [(Computer, [Program, ...]), ...]

  computers\_to\_many\_programs = [

    (

      c,

      [

        next(

          p

          for p in programs

          if p.id == cp.program\_id

        )

        for cp in computer\_programs

        if cp.computer\_id == c.id

      ]

    )

    for c in computers

  ]

  print("Задание А3")

  print(

    {

      x[0].ItName: [y.name for y in x[1]]

      for x in sorted(

        computers\_to\_many\_programs,

        key=lambda x: x[0].ItName

      )

    },

  )

Пример работы программы:

