

**Московский государственный технический
университет им. Н.Э. Баумана**

**Факультет «Радиотехнический»
Кафедра «Системы обработки информации и управления»**

Курс «Парадигмы и конструкции языков программирования»

**Отчет по рубежному контролю №1
Вариант запросов Е
Вариант предметной области №12**

Выполнил:
студент группы РТ5-31Б:
Кузнецов С. А.

Подпись и дата:

Проверил:
преподаватель кафедры ИУ5
Гапанюк Ю. Е.

Подпись и дата:

Москва, 2024

Текст программы

Файл «main.py»:

```
# используется для сортировки
from operator import itemgetter

class Lang:
    """Язык программирования"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class Env:
    """Средство разработки"""
    def __init__(self, id, name, init_release_yr, lang_id):
        self.id = id
        self.name = name
        self.init_release_yr = init_release_yr
        self.lang_id = lang_id

class EnvLang:
    """
    'Языки в средах разработки' для реализации
    связи многие-ко-многим
    """
    def __init__(self, lang_id, env_id):
        self.lang_id = lang_id
        self.env_id = env_id

#Языки программирования
langs = [
    Lang(1, "C"),
    Lang(2, "Java"),
    Lang(3, "C++"),
    Lang(4, "C#")
]

#Средства разработки
envs = [
    Env(1, "Visual Studio", 1997, 4),
    Env(2, "Eclipse", 2001, 2),
    Env(3, "Code::Blocks", 2005, 3),
    Env(4, "Vim", 1991, 1),
    Env(5, "Xcode", 2003, 3),
]

langs_envs = [
```

```

    EnvLang(1, 1),
    EnvLang(3,1),
    EnvLang(4,1),

    EnvLang(2, 2),
    EnvLang(1, 2),

    EnvLang(1,3),
    EnvLang(3,3),

    EnvLang(1,4),
    EnvLang(3,4),

    EnvLang(1,5),
    EnvLang(3,5)
]

#Запросы
def query1(list_to_sort):
    """Вывести список всех ЯП, содержащих в названии заглавную 'С', и список ра-
ботающих с ними IDE"""

    res_11 = list(filter(lambda i: ("С" in i[0]), list_to_sort))

    return res_11

def query2(list_to_sort):
    """Вывести список ЯП, где для каждого языка указан год выхода самой ранней
IDE, поддерживающей его.
Список отсортирован по году выхода по возрастанию.
Язык не выводится, если для него не существует IDE."""

    res_12_unsorted = list()

    for l in langs:
        env_dates = [i[2] for i in list(filter(lambda i: i[0] == l.name,
list_to_sort))] # Список лет выхода каждой IDE для конкретного языка
        if len(env_dates) > 0: # Может так оказаться, что для языка нет IDE
            res_12_unsorted.append((l.name, min(env_dates)))

    res_12 = sorted(res_12_unsorted, key=itemgetter(1))

    return res_12

def query3(list_to_sort):
    """Вывести список сред разработки, название которой начинается с буквы 'V', а
также названия языков, поддерживаемых каждой,
при условии, что связь между 'ЯП' и 'IDE' - 'многие ко многим'."""

    res_13 = dict()

```

```

    for e in envs:
        if 'V' in e.name:
            # Список языков для данной среды
            e_langs = [i[2] for i in list(filter(lambda x: x[0] == e.name,
list_to_sort))]
            res_13[e.name] = e_langs

    return res_13

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(l.name, e.name, e.init_release_yr)
        for l in langs
        for e in envs
        if e.lang_id==l.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(l.name, le.lang_id, le.env_id)
        for l in langs
        for le in langs_envs
        if l.id==le.lang_id]

    many_to_many = [(e.name, e.init_release_yr, lang_name)
        for lang_name, lang_id, env_id in many_to_many_temp
        for e in envs if e.id==env_id]

    print('Задание E1')
    print(query1(one_to_many))

    print('\nЗадание E2')
    print(query2(one_to_many))

    print('\nЗадание E3')
    print(query3(many_to_many))

if __name__ == '__main__':
    main()

```

Результат выполнения программы

Задание E1

[('C', 'Vim', 1991), ('C++', 'Code::Blocks', 2005), ('C++', 'Xcode', 2003), ('C#', 'Visual Studio', 1997)]

Задание E2

[('C', 1991), ('C#', 1997), ('Java', 2001), ('C++', 2003)]

Задание E3

{'Visual Studio': ['C', 'C++', 'C#'], 'Vim': ['C', 'C++']}