



**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 предмету

«Базовые компоненты интернет-технологий»

Выполнил:

студент группы № ИУ5-33Б

Пермяков Дмитрий

Проверил:

Преподаватель кафедры ИУ-5

Гапанюк Юрий

2022 г.

Код программы.

Также код программы можно найти на [GitHub](#).

Классы.

Путь: `libs\DevelopEnvironment`.

```
class DevelopmentEnvironment:
    def __init__(self, ID_environment, name_environment, count_users, *args):
        """
        :param ID_environment: id языка программирования. Int.
        :param name_environment: Название языка программирования. Str.
        :param args: IDE для этих языков программирования. Tuple.
        """
        self.__ID = ID_environment
        self.__name = name_environment
        self.__languageID = args
        self.__count_users = count_users

    @property
    def ID(self) -> int:
        return self.__ID

    @ID.setter
    def ID(self, ID_environment):
        self.__ID = ID_environment

    @property
    def name(self) -> str:
        return self.__name

    @name.setter
    def name(self, name_env):
        self.__name = name_env

    @property
    def languageID(self):
        return self.__languageID

    @languageID.setter
    def languageID(self, id):
        self.__languageID = id

    @property
    def count_users(self) -> int:
        return self.__count_users

    @count_users.setter
    def count_users(self, count):
        if type(count) != int:
            raise TypeError('Format of count_users must be Int!')
        self.__count_users = count
```

Путь: `libs\ProgrammingLanguage`.

```
class ProgrammingLanguage:
    def __init__(self, ID_language, name_language):
        self.__ID_language = ID_language
        self.__name = name_language

    @property
```

```

def ID(self) -> int:
    return self.__ID_language

@ID.setter
def ID(self, ID_language):
    self.__ID_language = ID_language

@property
def name(self) -> str:
    return self.__name

@name.setter
def name(self, name_language):
    self.__name = name_language

```

Путь: `libs\IdeLanguage`.

```

class IdeLanguage:
    def __init__(self, languageID, ideID):
        self.language_ID = languageID
        self.IDE_ID = ideID

```

Tasks.py

```

# Copyright © 2022 mightyKingRichard <dimapermyakov55@gmail.com>

from operator import itemgetter

def task_01(one_to_many) -> list:
    """ ЗАДАНИЕ №1.
        Вывести имя IDE, которое начинается с буквы 'V' и количество
        пользователей и отсортированный список языков,
        которые поддерживается данным IDE.
        :param one_to_many: Список один ко многим.
        :return: Отфильтрованный список по первой букве, с отсортированным
        внутренним списком.
    """
    # Пояснение. Сначала отфильтровал список one_to_many по принципу: первая
    # буква равна 'v', потом
    # создал новый список res, сортируя список класса. Новая переменная, так
    # как tuple нельзя изменять.
    return [(name, count, sorted(lst)) for name, count, lst in
            list(filter(lambda el: el[0][0].lower() == 'v', one_to_many))]

def task_02(one_to_many) -> list:
    """ Задание 2.
        Вывести список IDE, которое поддерживает меньше всего языков. Вывод
        совершить в порядке убывания.
        :param one_to_many:
        :return: Список кортежей, состоящий из имени IDE и кол-ва языков, которые
        поддерживает.
    """
    # Пояснение. Сначала создаю список, состоящий из кортежей. Первый элемент
    # - имя IDE, второй - кол-во языков,
    # которое он поддерживает.
    # Потом сортирую список по второму элементу.
    return sorted([(el[0], len(el[2])) for el in one_to_many],
                  key=itemgetter(1),
                  reverse=True)

```

```
def task_03(many_to_many) -> list:
    """
        Выведите список всех связанных IDE и языков, отсортированный по IDE,
        сортировка по языкам произвольная.
        :param many_to_many: список созданный связями многие ко многим.
        :return: Список IDE и языков, отсортированные по IDE.
    """
    # Пояснение. Изначально сортирую many_to_many по IDE.name, после чего
    # возвращаю список кортежей:
    # имя и список языков.
    return [(name, languages) for name, count, languages in
sorted(many_to_many, key=itemgetter(0))]
```

Main.py

```
# Copyright © 2022 mightyKingRichard <dimapermyakov55@gmail.com>

from libs.DevelopmentEnvironment import DevelopmentEnvironment
from libs.IdeLanguage import IdeLanguage
from libs.ProgrammingLanguage import ProgrammingLanguage
from tasks import task_01
from tasks import task_02
from tasks import task_03

LANGUAGES = [
    ProgrammingLanguage(0, 'C++'),
    ProgrammingLanguage(1, 'C#'),
    ProgrammingLanguage(2, 'Python'),
    ProgrammingLanguage(3, 'Swift'),
    ProgrammingLanguage(4, 'JavaScript'),
    ProgrammingLanguage(5, 'CSS'),
    ProgrammingLanguage(6, 'HTML'),
    ProgrammingLanguage(7, 'Rust'),
    ProgrammingLanguage(8, 'Go')
]
IDE = [
    DevelopmentEnvironment(0, 'PyCharm', 267209, 2, 6, 5),
    DevelopmentEnvironment(1, 'Atom', 171106, 5),
    DevelopmentEnvironment(2, 'LITE', 305374, 8),
    DevelopmentEnvironment(3, 'XCode', 253462, 3, 6, 5),
    DevelopmentEnvironment(4, 'Visual Studio', 343304, 1, 0, 2, 6, 7),
    DevelopmentEnvironment(5, 'IntelliJ IDEA', 426855, 4),
    DevelopmentEnvironment(6, 'Rider', 308911, 1),
    DevelopmentEnvironment(7, 'VScode', 908311, 0, 1, 2, 4, 5, 6, 7)
]
IDE_LANGUAGES = [
    IdeLanguage(0, 4),
    IdeLanguage(1, 6),
    IdeLanguage(2, 0),
    IdeLanguage(3, 3),
    IdeLanguage(4, 5),
    IdeLanguage(5, 1)
]

def main_code():
    # Пояснение: Так как у в одной среде разработки можно использовать
    # несколько языков,
    # я создаю список в списке. Т.е список языков для определённого IDE.
    one_to_many = [
        (ide.name, ide.count_users,
         [language.name for ell in ide.languageID for language in LANGUAGES
         if language.ID == ell])
    ]
```

```

        for ide in IDE
    ]
    many_to_many_temp = [
        (language.name, language.ID, ide_language.IDE_ID)
        for language in LANGUAGES
        for ide_language in IDE_LANGUAGES
        if language.ID == ide_language.language_ID
    ]
    many_to_many = [
        (ide_temp.name, ide_temp.count_users,
         [lang.name for el in ide_temp.languageID for lang in LANGUAGES if
          lang.ID == el])
        for name language, languageID, ideID in many_to_many_temp
        for ide_temp in IDE
        if ide_temp.ID == ideID
    ]

    # Можно посмотреть содержимое списков.
    # print(*one_to_many, sep='\n', end=f'\n {"=" * 30}\n')
    # print(*many_to_many_temp, sep='\n', end=f'\n {"=" * 30}\n')
    # print(*many_to_many, sep='\n', end=f'\n {"=" * 30}\n')

    print(f'{"-" * 10} Задание №1. {"-" * 10}')
    print(*task_01(one_to_many), sep='\n', end='\n\n')
    print(f'{"-" * 10} Задание №2. {"-" * 10}')
    print(*task_02(one_to_many), sep='\n', end='\n\n')
    print(f'{"-" * 10} Задание №3. {"-" * 10}')
    print(*task_03(many_to_many), sep='\n', end='\n\n')

if __name__ == '__main__':
    main_code()

```

Результат программы.

```
src — helper_update — -zsh › sleep — 83x25
(base) dmitriy@MacBook-Air-Dmitrij src % python3 main.py
----- Задание №1. -----
('Visual Studio', 343304, ['C#', 'C++', 'HTML', 'Python', 'Rust'])
('VScode', 908311, ['C#', 'C++', 'CSS', 'HTML', 'JavaScript', 'Python', 'Rust'])

----- Задание №2. -----
('VScode', 7)
('Visual Studio', 5)
('PyCharm', 3)
('XCode', 3)
('Atom', 1)
('LITE', 1)
('IntelliJ IDEA', 1)
('Rider', 1)

----- Задание №3. -----
('Atom', ['CSS'])
('IntelliJ IDEA', ['JavaScript'])
('PyCharm', ['Python', 'HTML', 'CSS'])
('Rider', ['C#'])
('Visual Studio', ['C#', 'C++', 'Python', 'HTML', 'Rust'])
('XCode', ['Swift', 'HTML', 'CSS'])

(base) dmitriy@MacBook-Air-Dmitrij src %
```