

## **Практическое занятие № 10**

**Тема:** Составление программ с применением множеств в IDE PyCharm Community.

**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с применением множеств в IDE PyCharm Community.

### **Постановка задачи.**

Книжные магазины предлагают следующие коллекции книг.

Магистр – Лермонтов, Достоевский, Пушкин, Тютчев

ДомКниги – Толстой, Грибоедов, Чехов, Пушкин.

БукМаркет – Пушкин, Достоевский, Маяковский.

Галерея – Чехов, Тютчев, Пушкин. Определить:

1. Полный список всех книг магазинов.
2. Какие книги есть во всех магазинах.
3. Хотя бы одну книгу, которая есть не во всех магазинах.

**Тип алгоритма:** функциональный.

## Текст программы:

```
# -*- coding: utf8 -*-
# Создать словарь и определить:
# 1. Полный список всех книг магазинов.
# 2. Какие книги есть во всех магазинах.
# 3. Хотя бы одну книгу, которая есть не во всех магазинах.

books = {
    'Magister': {'Lermontov', 'Dostoevskiy', 'Pushkin', 'Tutchev'},
    'DomKnigi': {'Tolstoy', 'Griboedov', 'Chehov', 'Pushkin'},
    'BookMarket': {'Pushkin', 'Dostoevskiy', 'Mayakovskiy'},
    'Gallery': {'Chehov', 'Tutchev', 'Pushkin'}
}

def all_books(slovar_knig):
    # all_books_list = []
    # for i in slovar_knig.values():
    #     for j in i:
    #         if j not in all_books_list:
    #             all_books_list.append(j)
    # return all_books_list

    all_books = set().union(*slovar_knig.values())
    return all_books

def can_buy_everywhere(slovar_knig):
    common_books = set.intersection(*slovar_knig.values())
    return common_books

def unique_book(slovar_knig):
    all_books = set().union(*slovar_knig.values())
    common_books = set.intersection(*slovar_knig.values())
    unique_book = all_books - common_books
    return unique_book

print(f'Список всех книг во всех магазинах - ', all_books(books))
print(f'Книги которые можно купить во всех магазинах - ',
      *can_buy_everywhere(books))
print(f'Книги которые можно купить НЕ во всех магазинах(уникальные) - ',
      unique_book(books))
```

## Протокол работы программы:

Список всех книг во всех магазинах - {'Tolstoy', 'Mayakovskiy', 'Tutchev', 'Dostoevskiy', 'Chehov', 'Lermontov', 'Griboedov', 'Pushkin'}

Книги которые можно купить во всех магазинах - Pushkin

Книги которые можно купить НЕ во всех магазинах(уникальные) - {'Tolstoy', 'Mayakovskiy', 'Tutchev', 'Dostoevskiy', 'Chehov', 'Lermontov', 'Griboedov'}

Process finished with exit code 0

**Вывод:** В данной программе реализован следующий подход:

1. Используется структура данных словарь для хранения информации о книгах в разных магазинах.
2. Реализованы три функции:
  - `all_books(slovar_knig)`: формирует полный список всех книг, представленных в магазинах.
  - `can_buy_everywhere(slovar_knig)`: определяет книги, которые представлены во всех магазинах.
  - `unique_book(slovar_knig)`: находит уникальные книги, доступные не во всех магазинах.
3. Осуществляется вызов каждой функции с данными из словаря книг и вывод результатов на экран.

Этот алгоритм относится к обработке сравнения данных и решает задачу анализа информации о наличии определенных книг в разных магазинах. Результаты включают список всех книг, книги, доступные во всех магазинах, а также уникальную книгу, которую можно купить не в каждом магазине.

Готовые программные коды выложены на GitHub.