

# ЗАДАНИЕ

## Технические требования к заданию:

### Задача 1:

Мы работаем над системой управления участниками конкурсов. Наша задача написать программу, которая анализирует три множества участников разных конкурсов и выполняет следующие операции:

1. Определяет, кто участвовал во всех трёх конкурсах.
2. Определяет, кто участвовал хотя бы в одном конкурсе.
3. Находит участников, которые участвовали только в одном конкурсе.
4. Находит участников, которые участвовали ровно в двух конкурсах.

Указания:

1. Программа должна принимать три множества участников от пользователя.
2. Использовать операции с множествами для выполнения задач.
3. Результаты должны быть представлены в удобной форме, например, через вывод множеств.

Ожидаемый результат:

1. Участники всех трёх конкурсов:

Программа выводит множество участников, которые присутствуют во всех трёх конкурсах одновременно. Если таких участников нет, результат будет пустым множеством.

2. Участники хотя бы одного конкурса:

Программа выводит множество всех уникальных участников, которые участвовали хотя бы в одном из трёх конкурсов.

3. Участники только одного конкурса:

Программа выводит множество участников, которые участвовали исключительно в одном из конкурсов, но не участвовали в двух других.

4. Участники ровно двух конкурсов:

Программа выводит множество участников, которые участвовали ровно в двух из трёх конкурсов. Участники, участвовавшие во всех трёх или только в одном конкурсе, сюда не включаются.

5. Формат вывода:

Каждый результат представлен в виде множества (фигурных скобок) с именами участников. Если результат пуст, выводится пустое множество: `set()`

### Задача 2:

Создайте программу для управления реестром участников мероприятия, используя кортежи, множества и словари. Пользователь должен иметь возможность регистрировать новых участников, удалять их из реестра и просматривать полный список с деталями.

Указания:

1. Программа должна начинаться с пустого словаря участников.
2. Ключом в словаре должен быть кортеж, состоящий из имени и фамилии участника, а значением — множество с их интересами.
3. Пользователь может добавить нового участника в реестр, используя команду `add`.
4. Пользователь может удалить участника из реестра, используя команду `remove`.
5. Пользователь может просмотреть всех участников с их интересами, используя команду `list`.
6. Пользователь может выйти из программы, используя команду `exit`.

7. Программа должна обеспечивать обработку ошибок ввода, таких как запрос удаления несуществующего участника.

Ожидаемый результат:

Программа циклически запрашивает у пользователя команду (`add`, `remove`, `list` или `exit` для выхода), выполняет ее и затем снова запрашивает команду. Программа должна корректно обрабатывать каждую команду и выдавать соответствующие сообщения о статусе операции.

**ПО:** PyCharm (по желанию другое)

**Формат подачи файлов:** архив с файлами с расширением `.py` /файл с расширением `.py`/ссылка на гит репозиторий