

Выполнил: Шепелев Артем

Описание сценария генерации результатов опроса:

Был использован сценарий случайной генерации результатов опроса. Для этого были использованы текстовые файлы с мужскими (143 штуки) и женскими (136 штук) именами, список песен (500 штук), а также список фамилий (250 штук). Перед генерацией опроса эти данные считывались в списки. При генерации одного результата опроса, с помощью модуля random генерировались псевдослучайные числа, с помощью которых случайным образом выбирались: пол, возраст (от 6 до 60 лет), имя, фамилия, а также 5 случайных песен, за которые были отданы голоса. Далее, эти данные построчно, разделяясь символами двоеточия вводились в файл (всего 50000 записей).

Описание сценария обработки результатов опроса:

Для обработки результатов опроса с целью поиска списка песен в порядке возрастания рейтинга, был использован ассоциативный массив. В качестве ключа в нем выступало название песни, а в качестве значения – кортеж из двух элементов: рейтинг песни и число появлений. Рейтинг песни считался следующим образом: если песне отдавали 5 место, то её общему рейтингу добавлялся один балл, таким образом, если песне отдавали 1 место, то её общему рейтингу добавлялось 5 баллов. После формирования данного словаря, проводилась сортировка по рейтингу, по убыванию значения и дальнейший вывод.

Для обработки результатов опроса с целью поиска предпочтений каждой из категорий был также использован ассоциативный массив. В качестве ключа выступала песня, а в качестве значения список из 4 элементов: число появлений данной песни для каждой из категорий. После формирования ассоциативного массива, данный словарь последовательно сортировался по каждому из элементов списка, который выступал в роли значений ключей

ассоциативного массива. Помимо этого, был заведен счетчик, который останавливал вывод содержимого отсортированных списков после достижения значения 10, т.е. было выведено 10 элементов в каждой категории.

Для обработки результатов опроса с целью поиска людей, которые проголосовали за наиболее популярные песни из их категории был использован ассоциативный массив. В качестве ключа использовалась песня, а в качестве значения – список из 5 элементов: 4 элемента с числом появлений данной песни в каждой из категории, а также список из людей, проголосовавших за данную песню. После составления словаря, была проведена последовательная сортировка по числу появлений песен в каждой из категории, и затем, для заданных n значений из начала списка упорядоченных элементов, был выбран список людей, проголосовавших за данную песню.

Данные по производительности обоих сценариев:

Объем исходных данных (файл с результатами опроса) ~ 9.5 Мб. Объем результатов работы первого алгоритма: ~ 32 Кб. Объем результатов работы второго алгоритма: ~ 3 Кб. Объем результатов работы третьего алгоритма: ~1727 Кб.

На генерацию результатов опроса ушло 2.12 секунды. На работу первого алгоритма ушло 0.6414 секунды. На работу второго алгоритма ушло 0.907 секунды. На работу третьего алгоритма ушло 1.393 секунды. В целом, на работу всех 4 алгоритмов уходит ~ 5.12 секунд.

Параметры компьютера, на котором проводились испытания: Процессор: AMD Phenom II X4 925 2.80 ГГц, тактовая частота увеличена до 3.07 ГГц. Оперативная память: 4 Гб DDR3. Жесткий диск: Seagate SATA2 7200 об. в. мин.

В силу сложившегося ограничения на загружаемый файл, из отчёта были удалены: файл с результатами опроса, html-отчеты. Однако, они легко генерируются из основной программы. Для этого запустите программу `program.py` и в меню программы, выберите пункт 0 и нажмите Enter.