

Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

Лабораторная работа №2

Вариант №1

Выполнил: Миляев Д.Д.

Шишминцев Д.В.

Проверил Мусаев А.А.

Санкт-Петербург,

2022

ВВЕДЕНИЕ

Целью данной лабораторной работы является реализация алгоритма по определению задуманного пользователем студента группы K3121. Инициализация человека должна происходить по средствам получения ответов формата да/нет на выводимые пользователю вопросы.

РЕШЕНИЕ

Для реализации алгоритма необходимо прежде всего разработать список вопросов, которые будут предоставлены пользователю. Кроме того, потребуется таблица корректных для всех студентов ответов на каждый из пунктов этого списка. Для ускорения процесса в чате группы был создан опрос, предлагающий каждому человеку от своего лица дать ответ на все вопросы. Далее полученная информация была записана в виде таблицы Excel и отправлена в беседу. Таким образом, мы унифицировали список вопросов и затратили на сбор ответов минимальное количество времени и усилий.

Следующим шагом был выбор структуры данных, в которой будем хранить информацию о студентах. Наилучшим вариантом является словарь, обращение к которому не занимает времени. Однако мы остановились на варианте использования базы данных (БД). Такой подход:

- показался наиболее интересным в плане реализации
- позволяет изучить взаимодействие БД с Python
- реализует удобное представление информации
- не захламляет код программы

БД была реализована в SQLite и выглядит следующим образом:

students			CREATE TABLE students (id INTEGER PRIMARY KEY, name TEXT, girl TEXT, is18 TEXT, glasses TEXT, brownEyes TEXT, c
id	INTEGER	"id" INTEGER	
name	TEXT	"name" TEXT	
girl	TEXT	"girl" TEXT	
is18	TEXT	"is18" TEXT	
glasses	TEXT	"glasses" TEXT	
brownEyes	TEXT	"brownEyes" TEXT	
darkHair	TEXT	"darkHair" TEXT	
siblings	TEXT	"siblings" TEXT	
pet	TEXT	"pet" TEXT	
fromSPb	TEXT	"fromSPb" TEXT	
iphone	TEXT	"iphone" TEXT	
tushavin	TEXT	"tushavin" TEXT	
coffe	TEXT	"coffe" TEXT	
homefood	TEXT	"homefood" TEXT	
watch	TEXT	"watch" TEXT	
nameVowel	TEXT	"nameVowel" TEXT	
pizza	TEXT	"pizza" TEXT	
smoke	TEXT	"smoke" TEXT	
heightOver175	TEXT	"heightOver175" TEXT	
partner	TEXT	"partner" TEXT	
winterOrSpring	TEXT	"winterOrSpring" TEXT	
gaming	TEXT	"gaming" TEXT	
campus	TEXT	"campus" TEXT	
movies	TEXT	"movies" TEXT	
anime	TEXT	"anime" TEXT	

Рисунок 1 – Структура БД

Процесс определения студента при фиксированном порядке вопросов можно представить в виде следующего графа.



Рисунок 2 – Граф отбора

Однако для большей интерактивности и разнообразия было принято решение выводить вопросы в случайном порядке.

При написании программы было задействовано две библиотеки: `sqlite3`, необходимая для отправки запросов в БД и `random` для генерации порядка вопросов. Код соответствует следующей логике:

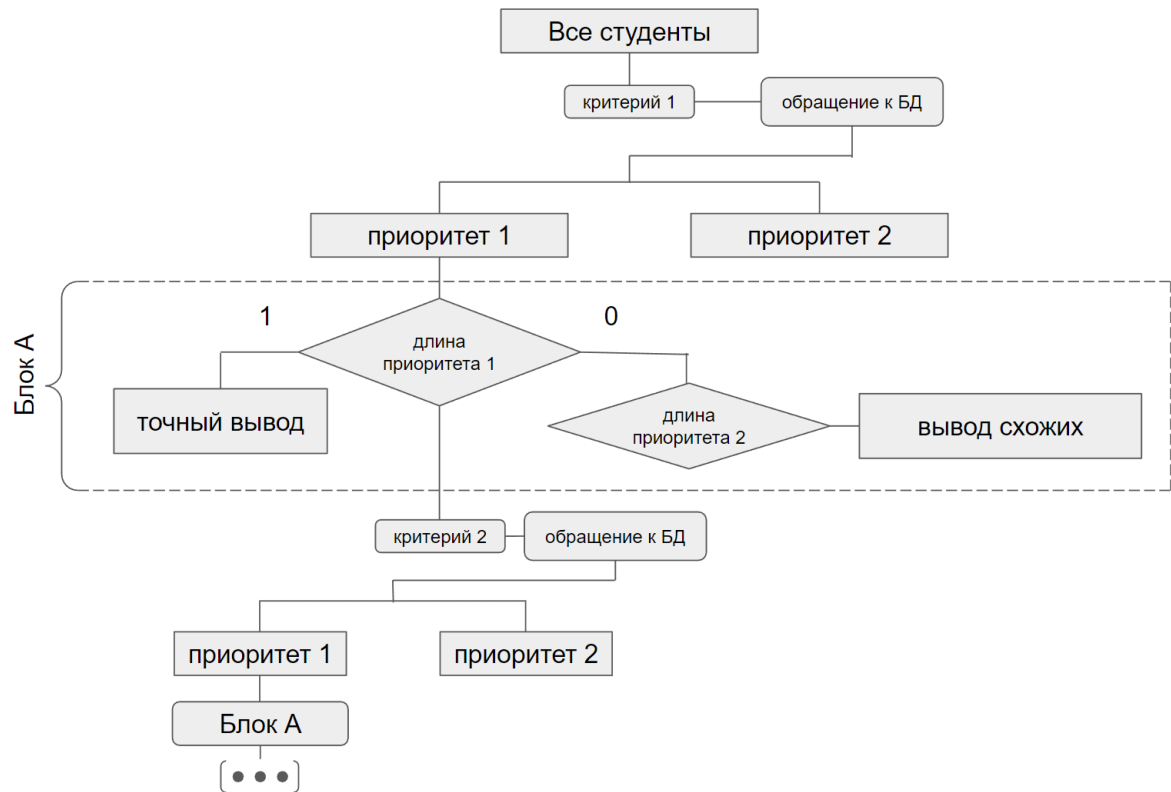


Рисунок 3 – Логика

Под приоритетом 1 понимается набор студентов, прошедших отбор по последнему критерию (ответу на вопрос), под приоритетом 2 – оставшиеся студенты. Представленная логика позволяет рассмотреть случай, когда пользователь допустил ошибку при вводе ответов (ввёл набор критериев, которому не соответствует ни один человек). Вопросы хранятся в словаре, откуда удаляются после использования, а потому не повторяются.

Код программы представлен в приложении.

ВЫВОД

Выполнение лабораторной работы позволило изучить принципы работы БД, взаимодействие между Python и SQLite, вспомнить и проанализировать структуры данных Python и реализовать интересную задачу.

СПИСОК ЛИТЕРАТУРЫ

1. Руководство по SQLite в Python: официальный сайт. – URL: <https://pythonru.com/osnovy/sqlite-v-python> (Дата обращения 18.10.2022).
2. Руководство по SQLite: официальный сайт. – URL: <https://metanit.com/sql/sqlite/> (Дата обращения 18.10.2022).

ПРИЛОЖЕНИЕ

Ссылка на гит с кодом и базой данных:

https://github.com/Evisom/itmo/tree/master/algorithms_data_structures/lab2