**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Кафедра “фундаментальная информатика и информационные технологии”**

**ОТЧЕТ**

**по лабораторной работе №1**

**по дисциплине «Алгоритмы и структуры данных»**

**на тему «генерация синтетических данных»**

**Вариант – 5**

**Студент гр. 22Б16-пу \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Олизько С.С.**

**Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дик А.Г.**

**Санкт-Петербург**

**2023 г**

Цель работы...................................................................................................................3

Задача ........................................................................................................................... 3

Теоретическая часть .....................................................................................................3

Алгоритм метода ...........................................................................................................4

Блок-схема программы .................................................................................................5

Описание программы ...................................................................................................7

Рекомендации пользователя ......................................................................................11

Рекомендации для программиста ..............................................................................11

Контрольный пример ...................................................................................................12

Вывод ............................................................................................................................13

Источники......................................................................................................................13

**Цель работы:**

Целью проекта является освоить и понять методы и принципы создания синтетических данных.

**Задача:**

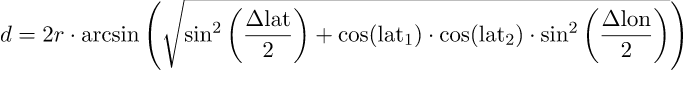
Создать алгоритм и разработать программу, которые позволят генерировать синтетические данные покупок железнодорожных билетов. Программа должна быть способна принимать заданные параметры и генерировать датасет, отражающий покупку билетов на железнодорожные поездки.  
 **Теоритическая часть:**

Для создания датасета удобно разделить генерацию данных на две основные части: создание случайного расписания поездов и создание билетов для отдельных пассажиров.

При создании расписания поездов учитывается принадлежность поездов определенного номера к определенному типу, а так же то, что направление движения поезда связано с четностью его номера.   
Расстояние между пунктами отправления и назначения рассчитывается по формуле гаверсинусов, что позволяет быстро сгенерировать его и получить идентичный результат при повторной генерации.   
  
Формула гаверсинусов, также известная как формула синусов или теорема синусов, применяется для нахождения длин сторон и углов треугольника. Эта формула включает в себя отношения длин сторон треугольника к синусам соответствующих углов. Формула синусов может быть полезна для решения различных геометрических задач.

Однако формула гаверсинусов для точек антиподов применяется в геодезии и связана с вычислением расстояний на поверхности Земли между двумя точками, которые находятся на противоположных сторонах относительно центра Земли. Такие точки называются антиподами.

Формула гаверсинусов для точек антиподов позволяет вычислить расстояние между этими точками на поверхности Земли с использованием широт и долгот этих точек. Эта формула связана с законами сферической тригонометрии и имеет следующий вид:



*Рис 1: формула гаверсинусов*

Где:

* d - расстояние между двумя точками на сфере (обычно в километрах или милях).
* r - радиус сферы (например, средний радиус Земли, который примерно равен 6,371 километру).
* lat1 и lat2 - широты первой и второй точек в радианах.
* Δlat - разница между широтами двух точек в радианах.
* Δlon - разница между долготами двух точек в радианах.

Эта формула позволяет рассчитать кратчайшее расстояние между двумя точками на сфере, и она часто используется в геодезии и географических приложениях, таких как определение расстояний между географическими координатами, например, для GPS-навигации и картографии.

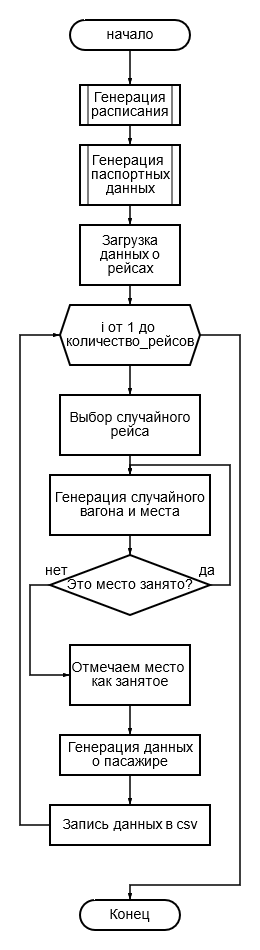
Для генерации даты используются библиотеки datetime и random. При генерации учитывается наличие поездов разового и сезонного следования.

Для генерации данных о конкретном купленном билете создаются данные пассажира с помощью библиотеки Faker и генерируется билет на один из полученных ранее рейсов.

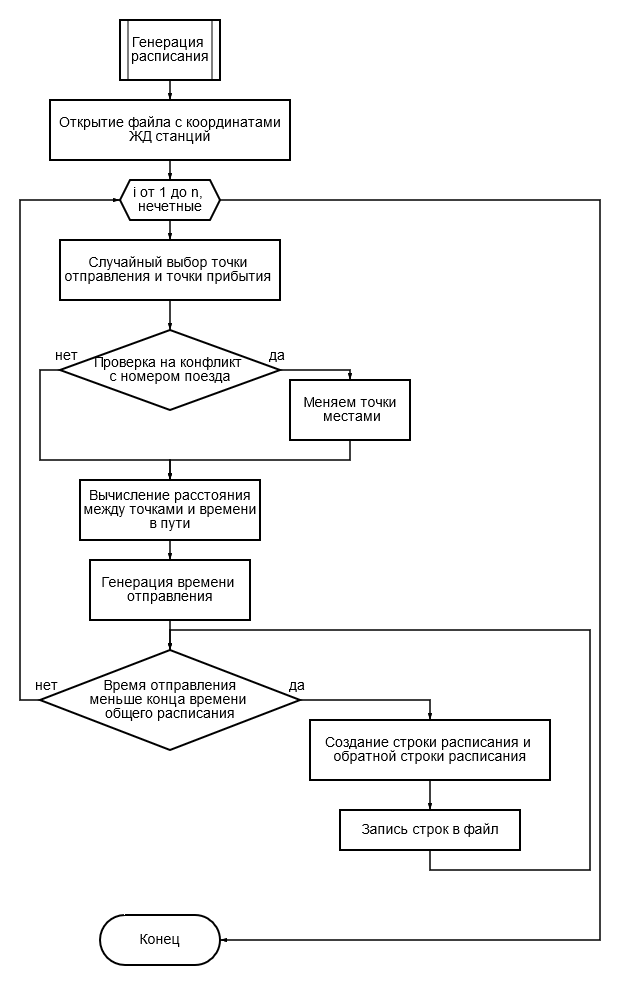
**Алгоритм метода:**

1. Открытие файла с координатами ЖД станций и загрузка данных в массив
2. Генерация нечетных номеров поездов
3. Генерация пункта отправления, назначения, определение их очередности и расстояния между ними
4. Генерация времени отправления и времени прибытия в зависимости от расстояния и типа поезда
5. Создание строк расписания маршрута с интервалом между рейсами
6. Запись строк в файл с расписанием
7. Запись четных строк с обратными рейсами
8. Создание окна с визуальным интерфейсом
9. Ввод вероятностей для банковских карт
10. Загрузка данных из файла с расписанием
11. Генерация полей в заданном количестве
    1. Выбор случайного рейса
    2. Генерация случайного вагона и места
    3. Генерация данных о пассажире
12. Запись данных в csv файл

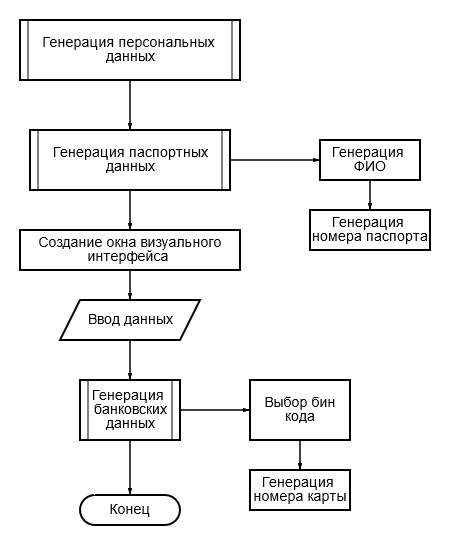
**Блок схема программы:**

****

*Рис 2: блок-схема*

****

*Рис 3: блок-схема*

****

*Рис 4: блок-схема*

**Описание программы:**

Программная реализация написана на языке python 3.10.4, с использованием библиотек: faker, random, tkinter, math, datetime, csv. В программе использовались 11 функций и 29 структур данных.

В таблице ниже представлено описание функций:

*Таблица 1: функции*

|  |  |  |
| --- | --- | --- |
| Имя функции | Тип возвращаемого значения | Описание функции |
| generate\_personal\_data | Tuple of strings | Генерирует случайные персональные данные (ФИО и паспортные данные) для пользователя, включая имя, отчество, фамилию, серию и номер паспорта. |
| start\_program | None | Запускает программу, проверяя настройки, введенные пользователем, и закрывает главное окно интерфейса. |
| update\_interface | None | Обновляет интерфейс на основе выбранных настроек пользователя, скрывая или показывая ползунки настройки вагонов и банков. |
| generate\_person | List of strings | Генерирует случайные персональные данные для пользователя, включая имя, отчество, фамилию, серию и номер паспорта, а также номер карты оплаты. |
| generate\_random\_seat | Tuple of string and float | Генерирует случайный выбор вагона и места для поезда, включая тип вагона и его номер, а также цену места. |
| calculate\_the\_distance | float | Рассчитывает расстояние между двумя географическими точками на основе их координат (широты и долготы). |
| determine\_direction | string | Определяет направление (северо-запад или юго-восток) между двумя географическими точками на основе их координат. |
| determine\_train\_speed | int | Определяет скорость поезда (км/ч) на основе номера поезда. |
| generate\_odd\_numbers | List of int | Генерирует список нечетных чисел в заданном диапазоне, учитывая ограничение на количество. |
| load\_timetable | List of tuples | Загружает данные о рейсах из файла "timetable.txt" и возвращает их в виде списка кортежей. |
| generate\_tickets\_csv | None | Генерирует базу данных билетов на основе данных о рейсах, сохраняя их в файле CSV. |

Описание структур данных:

*Таблица 2: структуры данных*

|  |  |  |  |
| --- | --- | --- | --- |
| Имя структуры | Тип структуры | Тип хранимых данных | Описание |
| probab\_flag | tkinter IntVar | Integer | Хранит значение, которое указывает на выбор пользователя относительно вероятности настроек. |
| num\_tickets | tkinter IntVar | Integer | Хранит количество билетов, выбранное пользователем. |
| bin\_codes\_oper\_w | List of tkinter DoubleVar | Floating-Point Numbers | Список переменных, каждая из которых хранит веса для выбора типов банковских операций. |
| bin\_codes\_bank\_w | List of tkinter DoubleVar | Floating-Point Numbers | Список переменных, каждая из которых хранит веса для выбора банков. |
| bin\_codes\_oper\_1 | List of strings | String | Список строк, представляющих типы банковских операций. |
| bin\_codes\_bank\_1 | List of strings | String | Список строк, представляющих названия банков. |
| bin\_codes\_oper | List of lists | List of strings | Список списков, где каждый внутренний список содержит бин-коды для соответствующего типа банковских операций. |
| bin\_codes\_bank | List of lists | List of strings | Список списков, где каждый внутренний список содержит бин-коды для соответствующего банка. |
| seat\_prices | Nested dictionary | Floating-Point Numbers | Словарь, который содержит цены на билеты для различных типов поездов и вагонов. |
| station\_data | List of tuples | String, Float | Список кортежей, каждый из которых содержит информацию о станции, включая название станции и её координаты (широту и долготу). |
| russian\_alphabet | String | Char | Строка, которая содержит русские буквы алфавита. |
| num\_trains | Integer | Integer | Хранит количество поездов. |
| start\_date | datetime object | Date and Time | Дата и время начала создания расписания поездов. |
| end\_date | datetime object | Date and Time | Дата и время окончания создания расписания поездов. |
| timetable | List | Strings, Numbers | Список строк, представляющих данные о расписании поездов. |
| train\_speed\_mps | Integer | Integer | Скорость поезда в метрах в секунду. |
| timetable\_data | List of tuples | Strings, Date and Time, Floating-Point Numbers | Список кортежей, представляющих данные о рейсах из файла "timetable.txt". |
| output\_filename | String | String | Имя файла, в который будут сохранены сгенерированные билеты. |

**Рекомендации пользователю:**

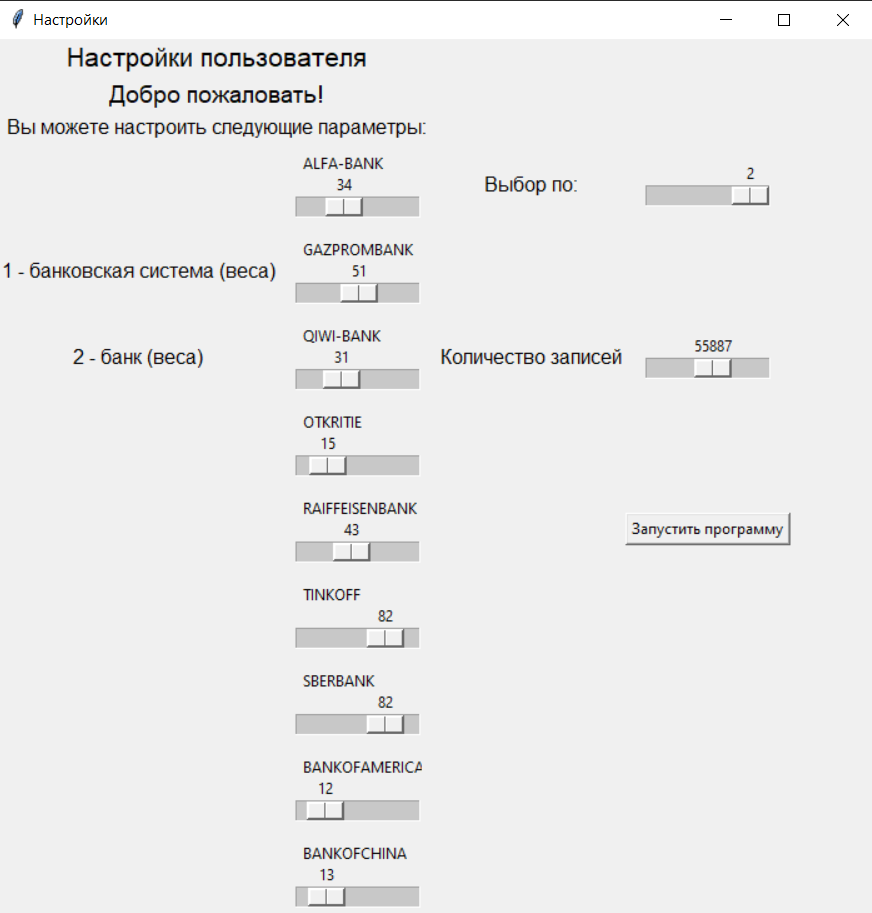
1. *Запуск программы*: Для запуска программы, убедитесь, что у вас установлен Python и библиотеки faker, tkinter, csv. Запустите код и используйте графический интерфейс для настройки параметров и генерации данных.
2. *Настройка параметров*: Вы можете выбрать тип данных, которые хотите сгенерировать, включая информацию о пользователях, билетах и другие параметры. Обязательно укажите количество записей (билетов), которое вы хотите сгенерировать.
3. *Предосторожность*: Обратите внимание на веса (параметры вероятности) для выбора типов банковских операций и банков. Убедитесь, что они не равны нулю, иначе программа выдаст ошибку.
4. *Ввод и вывод данных:* Программа сохранит сгенерированные данные в файл tickets.csv. Убедитесь, что вы указали корректное количество записей и ожидайте окончания процесса генерации данных.

**Рекомендации для программиста:**

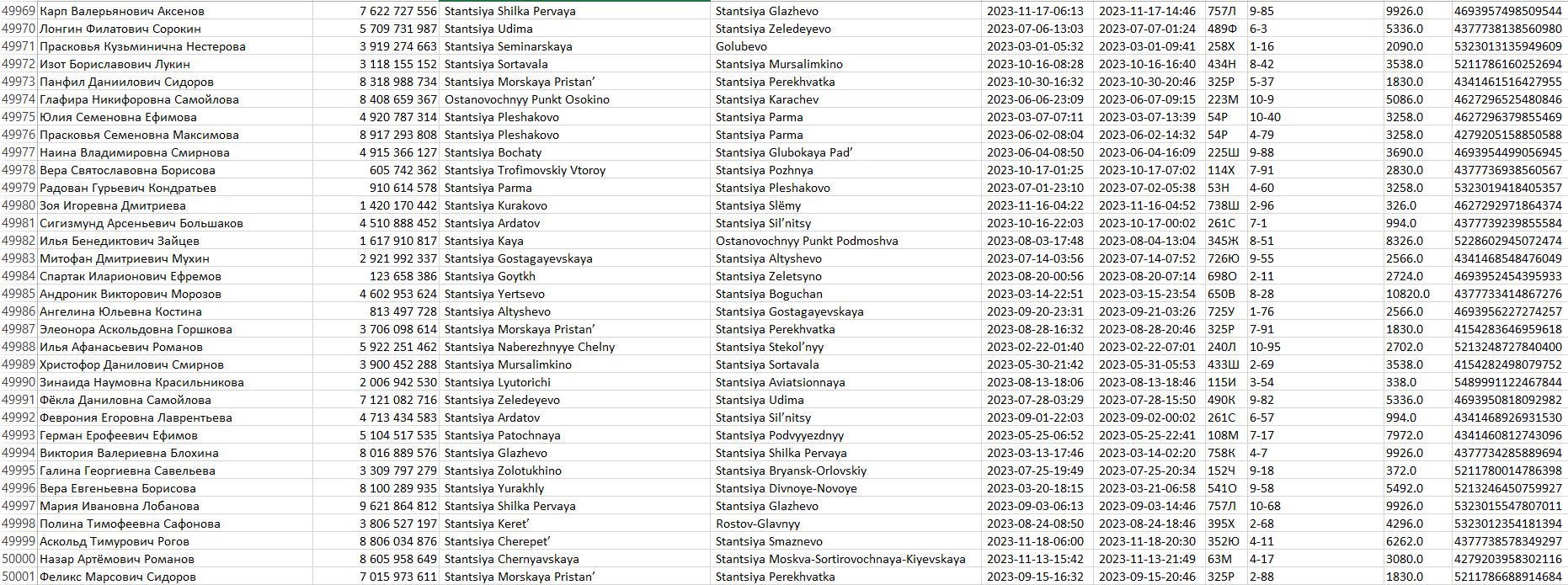
Для внесения изменений в код программы, вам потребуется устройство с установленным Python.

**Исходный код программы доступен по ссылке ниже:**

https://github.com/StephanOlizko?tab=repositories

**Контрольный пример:**  
  


*Рис 5: пример окна программы*



*Рис 6: пример вывода программы*

**Вывод:**

В рамках данной работы были исследованы основы генерации синтетических данных. Был разработан алгоритм создания датасета, моделирующего покупку железнодорожных билетов, а также написана соответствующая программа для генерации этого датасета.

**Источники:**

1. <https://faker.readthedocs.io/en/master/>
2. <https://docs.python.org/3/library/random.html>
3. <https://docs.python.org/3/library/tkinter.html>
4. <https://docs.python.org/3/library/csv.html>
5. <https://www.kobzarev.com/programming/calculation-of-distances-between-cities-on-their-coordinates/>