**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Программа бакалавриата**

**“Большие данные и распределенная цифровая платформа”**

**ОТЧЕТ**

**по лабораторной работе №1**

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

**на тему «Генерация** **датасета»**

**Вариант – 3**

**Студент гр. 23Б15-пу**

**Абрахин Е.Д.**

**Преподаватель**

**Дик А.Г.**

**Санкт-Петербург**

**2024 г.**

**Оглавление**

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

Описание задачи............................................................................................................3

Теоретическая часть......................................................................................................4

Основные шаги программы...........................................................................................5

Блок-схема программы..................................................................................................6

Описание программы.....................................................................................................8

Рекомендации пользователя........................................................................................9

Рекомендации для программиста................................................................................9

Исходный код программы..............................................................................................9

Контрольный пример...................................................................................................10

Вывод.............................................................................................................................1

Источники......................................................................................................................11

**Цель работы**

Целью лабораторной работы является разработка системы генерации датасета для покупки железнодорожных билетов с учетом определенных требований и условий. Датасет должен включать личные данные пассажиров, информацию о поездах, вагонах и местах, а также данные о платежных картах для оплаты билетов.

**Описание задачи (формализация задачи)**

Задача состоит в создании датасета для железнодорожных билетов со следующими требованиями:

1. **ФИО**: Славянские имена и фамилии.
2. **Паспортные данные**: Русские паспортные данные с уникальными значениями.
3. **Откуда и куда**: Города в РФ, с различием отправной и конечной точек.
4. **Дата отъезда и приезда**: Случайные даты в пределах года.
5. **Рейс**: Номера поездов в заданных диапазонах.
6. **Выбор вагона и места**: Учет классов вагонов и типов мест.
7. **Стоимость**: В зависимости от длины маршрута и типа вагона.
8. **Карта оплаты**: Генерация карт с возможностью многократного использования с повторением не больше пяти раз и возможностью настраивать вероятность к какому банку и платежной системе принадлежит карта.

**Теоретическая часть**

Для создания датасета использованы несколько программных модулей:

1. TimetableGeneration.py: Генерация расписания поездов с учетом различных типов поездов, временных интервалов и расстояний между городами. Учитывает специфику поездов (скоростные, высокоскоростные и т.д.) и создает файл timetable.csv.
2. PersonGeneration.py: Генерация личных данных пассажиров с использованием библиотеки Faker. Включает создание паспортных данных, ФИО и платежных карт с учетом введенных весов для выбора банка и платежной системы.
3. SeatGeneration.py: Генерация информации о вагонах и местах, включая расчет стоимости билета в зависимости от типа вагона.
4. RailwayTicketsGeneration.py: Основной скрипт для генерации билетов, который объединяет данные из вышеуказанных модулей и записывает их в CSV файл.

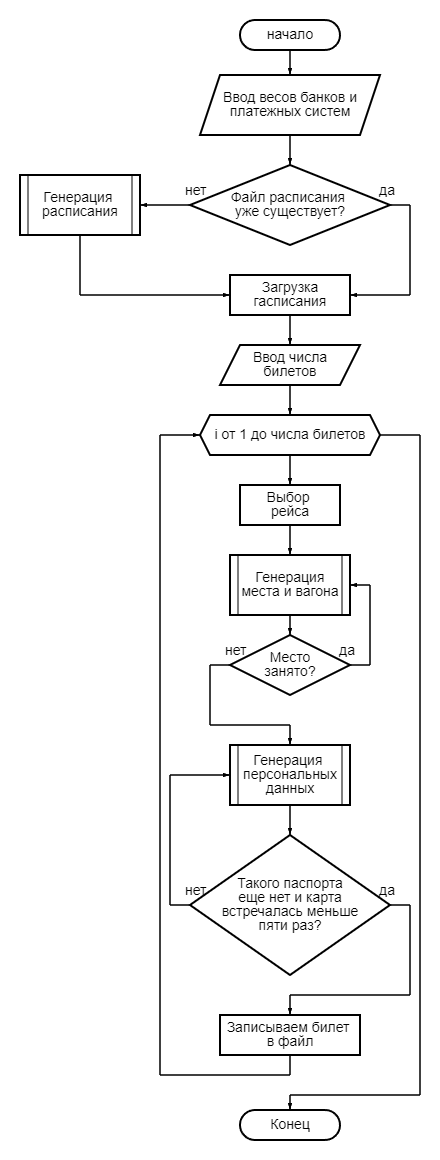
Ограничения:

* Количество строк в датасете ограничивается вводом пользователя.
* ФИО пассажиров только славянские.
* Уникальность паспортных данных и их принадлежность к русским.
* Разные точки отправки и прибытия только по территории РФ.
* Рейсы уникальные из определенных номеров, которые бывают до полного возвращения на вокзал.
* Вагоны и места генерируются по заданным типам.
* Веса банков и платежных систем определяются пользователем и должны в сумме давать число больше нуля.
* Логика выбора карт оплаты с ограничением на 5 повторов.

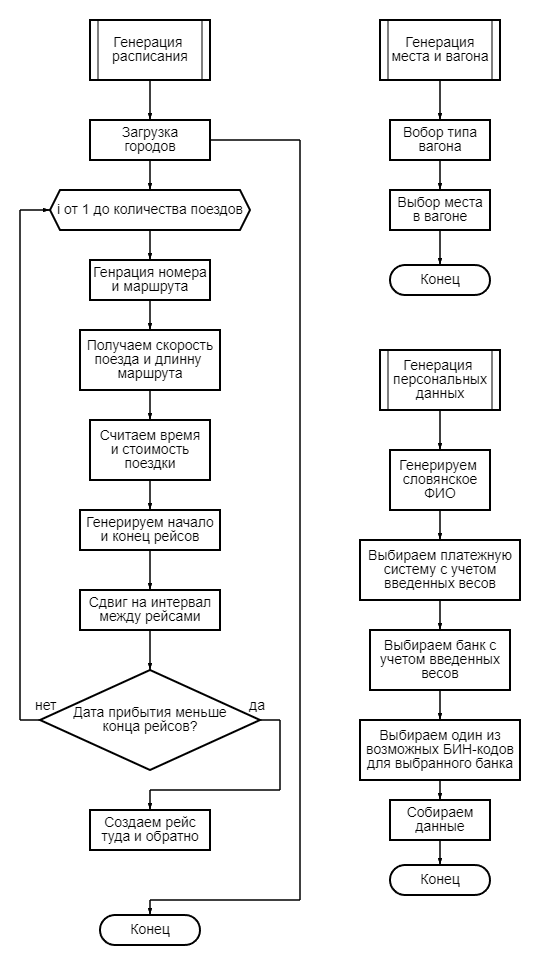
**Основные шаги программы**

1. Запуск программы (RailwayTicketsGeneration.py):
2. Пользователь вводит веса банков и платежных систем.
3. Загружается расписание поездов из файла timetable.csv через функцию loadTimetable. Если файл отсутствует, сначала генерируется расписание с помощью скрипта TimetableGeneration.py.
   1. Генерация расписания поездов, включая расчет расстояний, скорости, времени поездки и стоимости билетов на основе данных о городах и параметров поездов.
4. Пользователь вводит количество билетов для генерации.
   1. Билеты генерируются с использованием данных о расписании, персональных данных и месте в вагоне, а также информации о платежных картах.
      1. Генерация случайного места в поезде с учетом типа вагона и его стоимости.
      2. Генерация персональных данных: Имя, фамилия, отчество, серия и номер паспорта.
      3. Генерация платежной карты с использованием весов для выбора платежной системы и банка.
      4. Происходит расчет итоговой стоимости билета.
5. Данные собираются и записываются в файл tickets.csv.

**Блок схема программы**



*Рис 1: блок-схема основной программы*



*Рис 2: блок-схема подпрограмм*

**Описание программы**

Программная реализация написана на языке Python 3.11.9 с использованием следующих библиотек: faker, random, math, datetime, и csv. Программа организована через модульную структуру с акцентом на генерацию данных для железнодорожных билетов. В процессе разработки программы использовались 11 функций, каждая из которых имеет чётко определённое назначение и 4 основных модуля:

1. RailwayTicketsGeneration.py:

* Функции:
  + generateTimetable(): Запуск генерации расписания.
  + loadTimetable(filename): Загрузка расписания из файла или генерация нового.
  + generateTicketsCsv(timetableData, numTickets, outputFilename): Генерация билетов и запись в CSV файл.

2. PersonGeneration.py:

* Функции:
  + generatePersonalData(): Генерация личных данных.
  + generatePerson(): Генерация полного набора данных для пассажира, включая платежные карты.

3. SeatGeneration.py:

* Функции:
  + generateRandomSeat(): Генерация информации о вагоне и месте.

4. TimetableGeneration.py:

* Функции:
  + calculateDistance(): Расчет расстояния между городами с помощью формулы гаверсинусов.
  + determineDirection(): Определение направления движения поезда.
  + determineTrainSpeed(): Определение скорости поезда по его номеру.
  + generateOddNumbers(): Генерация нечетных чисел для номеров поездов.

**Рекомендации пользователя**

* Для запуска программы, убедитесь, что у вас установлен Python и библиотеки faker, csv. Запустите код в среде разработки или командной строке и используйте консоль для настройки параметров и генерации данных.
* Убедитесь, что все модули программы находятся в одной директории для корректного выполнения.
* Запуск производиться файлом RailwayTicketsGeneration.py, который автоматически генерирует расписание в файл timetable.csv если его нет.
* Периодически проверяйте корректность данных перед генерацией билетов.
* Отдельно сгенерировать timetable.csv можно запустив TimetableGeneration.py.
* При желании использовать свой файл расписания, убедитесь, что он называется timetable.csv и его структура его заголовков совпадает с используемой: TrainNumber, DepartureCity, DestinationCity, DepartureTime, ArrivalTime, Cost.
* Настройте веса для платежных систем и банков в соответствии с вашими требованиями, но убедитесь, что веса банков и платежных систем больше нуля.

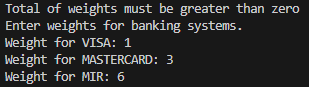
**Рекомендации программиста**

* Поддерживайте актуальность используемых библиотек и версии Python для сохранения актуальности и работоспособности кода на современных системах.
* Организуйте код в соответствии с best practices, следите за четким именованием переменных и функций.
* Периодически проводите тестирование на различных входных данных для обеспечения надежности и корректности программы.

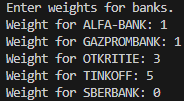
**Исходный код программы**

[**https://github.com/FasterXaos/Algorithms\_and\_Data\_Structures**](https://github.com/FasterXaos/Algorithms_and_Data_Structures)

**Контрольный пример**



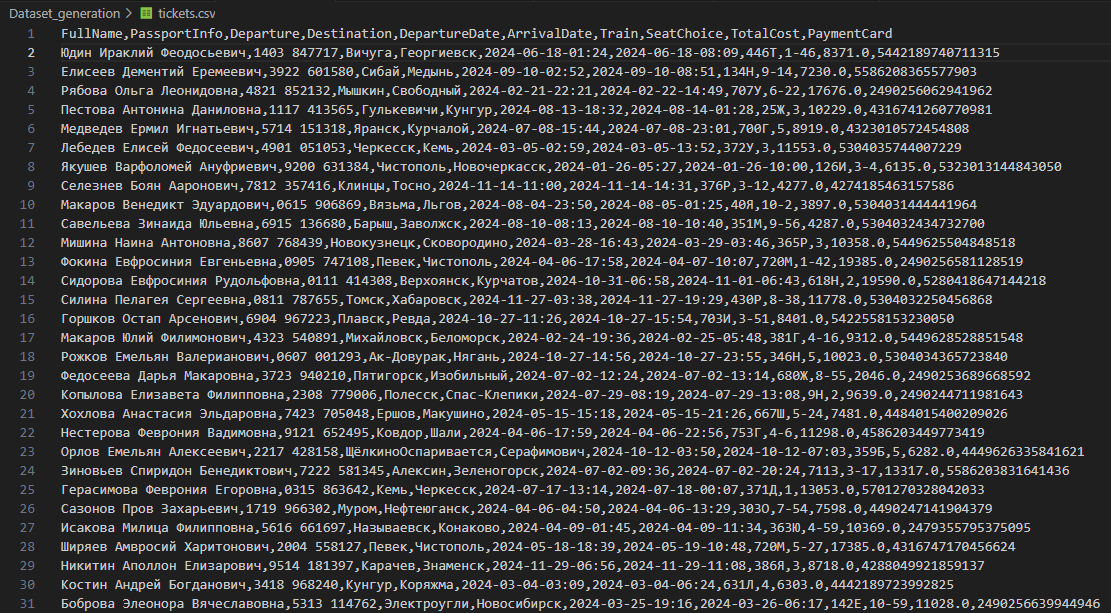
*Рис 3: пример ввода весов платежных систем*



*Рис 4: пример ввода весов банков*



*Рис 5: пример ввода количества билетов*



*Рис 6: пример датасета билетов*

**Вывод**

В рамках данной работы были исследованы принципы генерации синтетических данных, применительно к моделированию покупок железнодорожных билетов. Разработан алгоритм, который учитывает особенности пассажирских данных, маршрутов, типов поездов и вагонов, а также ценообразования билетов. Было реализовано программное обеспечение для автоматической генерации датасета, включающего такие данные, как личные данные пассажиров, информация о поездах, билетах и платежных системах. Программа позволяет настраивать параметры генерации банковских карт оплаты, обеспечивая соответствие заданным требованиям и реалистичность получаемого датасета.

**Источники**

<https://faker.readthedocs.io/en/master/>

<https://docs.python.org/3/library/random.html>

<https://docs.python.org/3/library/csv.html>

<https://on55.ru/statya/koordinaty-lat-i-lng-vsex-gorodov-rossii>

<https://www.freebinchecker.com/russian-federation_cjsc-alfa-bank-bin-list-bank?hl=ru>

<https://www.freebinchecker.com/russian-federation_gazprombank-ojsc-bin-list-bank?hl=ru>

<https://www.freebinchecker.com/otkritie-bank-jsc-issuer-bin-list?hl=ru>

<https://www.freebinchecker.com/tinkoff-credit-systems-bank-cjsc_russian-federation-issuer-bin-list-country?hl=ru>

<https://www.freebinchecker.com/russian-federation_sberbank-of-russia-bin-list-bank?hl=ru>