**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 – 1**

|  |  |  |
| --- | --- | --- |
| Студент гр. 23Б16-пу |  | Колосков В.Ю. |
| Преподаватель |  | Дик А.Г. |

**Санкт-Петербург**

**2024 г**

**Оглавление**

[Цель работы 3](#_Toc178355394)

[Задача 3](#_Toc178355395)

[Теоретическая часть 3](#_Toc178355396)

[Описание алгоритма 4](#_Toc178355397)

[Блок-схема программы 5](#_Toc178355398)

[Описание программы 11](#_Toc178355399)

[Рекомендации пользователя 14](#_Toc178355400)

[Рекомендации для программиста 15](#_Toc178355401)

[Контрольный пример 15](#_Toc178355402)

[Вывод 16](#_Toc178355403)

[Список использованной литературы 16](#_Toc178355404)

# **Цель работы**

Создать генератор синтетических данных о покупке товаров в магазинах Санкт-Петербурга.

# **Формализация задачи**

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

1. **Дата и время**: совпадают с расписание работы магазинов.
2. **Название магазина**: названия магазинов, существующих на территории СПб.
3. **Координаты**: магазина, с точностью до 6 знаков после запятой.
4. **Товар**: товар, который соответствует выбранному магазину.
5. **Бренд**: бренд, соответствующий выбранному товару.
6. **Выбор вагона и места**: Учет классов вагонов и типов мест.
7. **Количество**: от 1 до 5 штук.
8. **Стоимость**: В зависимости от количества и выбранного товара.
9. **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.
10. **Количество строк датасета**: Не меньше 50000.

# **Теоретическая часть**

Для создания датасета использован программный модуль и набор csv таблиц:

1. main.py: Генерация всех данных, необходимых для решения задачи. Берет нужные данные из файлов general\_table.csv, shop\_locations.csv. После завершения программы создает файл output.csv
2. general\_table.csv: таблица, которая содержит глобальную категорию товаров (clothes, electronics, food), бренд, продающий данные товары, сами товары, среднюю стоимость каждого товара в долларах США.
3. shop\_locations.csv: таблица, которая содержит глобальную категорию товаров (clothes, electronics, food), название магазина, его координаты.

* Количество строк в датасете ограничивается вводом пользователя, но минимальное количество сгенерированных строк будет 1000.
* Веса банков и платежных систем определяются пользователем и должны в сумме давать число больше нуля.
* Логика выбора карт оплаты с ограничением на 5 повторов.
* Соответствие товара, его цены, бренда и магазина, где он был куплен
* Локации магазинов точно совпадают с реальными ( использовался API Яндекс карт)

# **Основные шаги программы**

1. Запуск программы (main.py):
2. Пользователь вводит процентные соотношения банков и платежных систем, а так же выбирает категорию товаров, количество строчек в файле output.csv.
3. Загружаются данные из файлов general\_table.csv, shop\_locations.csv
4. Выполняется файл main.py.
   1. Покупки генерируются с использованием данных о расположении магазинов, персональных данных и цене товаров, а также информации о платежных картах.
      1. Генерация случайного товара с учетом типа магазина и его стоимости.
      2. Генерация локации магазина и времени его работы.
      3. Генерация платежной карты с использованием весов для выбора платежной системы и банка.
5. Данные собираются и записываются в файл output.csv.

# **Блок-схема программы**

# **Описание программы**

Программа реализована на языке python 3.11.4, с использованием библиотек: os, random, geopy, datetime, pandas, tkinter. В программе использовались 10 функций и 20 структур данных. В таблице 1 представлено описание функций

Таблица 1. Описание функций

|  |  |  |
| --- | --- | --- |
| Имя функции | Тип возвращаемого значения | Описание функции |
| Creating\_generation | None | Проверяет правильность введенных данных, запускает функции генерации данных и добавления в датасет, сохраняет датасет в файле с расширением xlsx |
| Password\_generation | list | Генерирует серии и номеров паспортов людей |
| Card\_generation | list | Генерирует номера банковских карт, с которых человек может произвести платеж |
| People\_generation | list | Генерирует имена, фамилии и отчества людей, которые могут купить билеты |
| Train\_generation | list | Генерирует расписание поездов, а также билеты на эти поезда |
| Open\_file | list | Получает необходимые для генерации данные из файла и записывает их в список |
| Random\_datetime | datetime | Генерирует дату и время отправления поезда |
| Route\_generation | tuple | Генерирует пункт отправки и пункт назначения, рассчитывает расстояние между городами |
| Completion\_vagons | tuple | Сопоставляет место с его ценой, возвращает количество место, которым сопоставлена цена во всех поездах и количество вагонов, используемых в данном поезде |
| Add\_in\_dataset | dict | Добавляет строку в датасет, возвращает словарь строк датасета |

В таблице 2 представлено описание структур данных, используемых в программе.

Таблица 2. Описание структур

|  |  |  |  |
| --- | --- | --- | --- |
| Имя структуры | Тип структуры | Тип хранимых данных | Описание |
| Pay\_name | list | string | Хранит имена возможных платежных систем |
| Bank\_name | list | string | Хранит имена возможных банков |
| Pay\_insert | list | string | Хранит поля, в которые вводятся вероятности выпадения определенных платежных систем |
| Bank\_insert | list | string | Хранит поля, в которые вводятся вероятности выпадения определенных банков |
| Used\_series | set | string | Хранит использованные серии паспортов |
| Used\_password | set | string | Хранит использованные номера паспортов |
| Not\_code\_base | set | int | Хранит несуществующие номера регионов |
| Used\_card | set | string | Хранит использованные номера банковских карт |
| Pay\_variant | array | int | Хранит коды возможных платежных систем |
| List\_file | array | string | Хранит данные, полученные из файла |
| Name\_file | array | string | Хранит имена файлов, в которых находятся необходимые данные |
| Used\_people | array | string | Хранит полные имена людей |
| Gender\_people | array | bool | Хранит пол людей. True – мужчина. False - женщина |
| Branded\_trains | dict | array | Ключ – особые поезда, значение – массив из названий городов, в которые они могут ехать |
| Type\_vagon | dict | dict | Ключ – класс вагона. – значение словарь его подклассов, значение второго словаря – коэффициенты, на которые умножается стоимость билета в зависимости от вагона |
| scredule | array | tuple | Хранит информацию о поезде |
| My\_used\_train | array | tuple | Хранит информацию о заполненном поезде и билеты на места в его вагонах |
| dataset | dict | array | Хранит строки датасета, которые были сгенерированы |
| tickets | array | tuple | Хранит билеты на данный поезд |
| Using\_train | array | tuple | Список поездов, которые можно использовать |

# **Рекомендации пользователя**

Для запуска программы необходимо наличие устройства с установленной операционной системой Linux, macOS или Windows, а также среды разработки, поддерживающей запуск python версии 3.11.4

Необходимо установить python библиотеки: geopy, pandas, tkinter. Для этого необходимо открыть терминал и прописать следующие команды: pip install geopy, pip install pandas, pip install tkinker

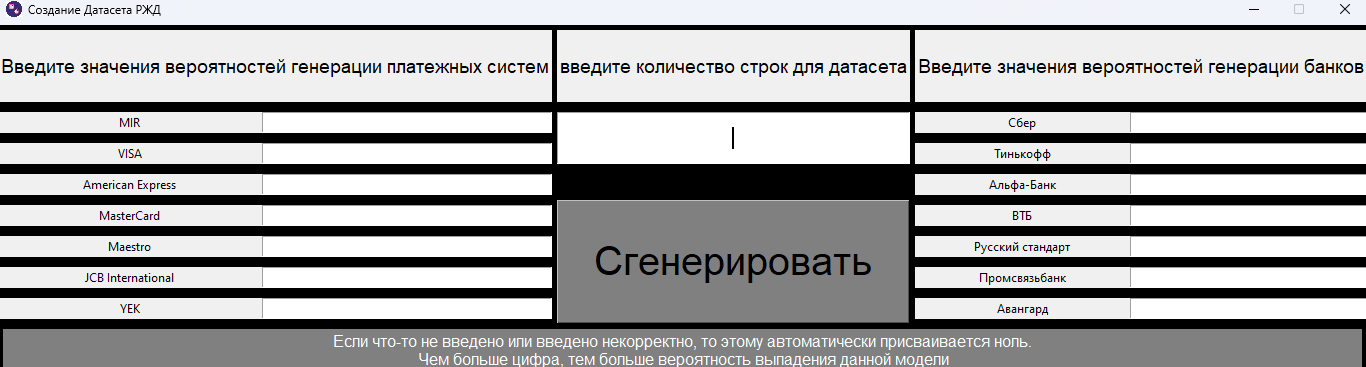
Под строкой «Введите значения вероятности генерации платежных систем» находятся названия платежных систем, а справа от них, соответствующая каждой системе, строка ввода. В эти строки необходимо ввести значения вероятности генерации для каждой платежной системы (рисунок. 7). Если оставить поле пустым или данные, введенные в него, будут некорректными, то вероятность автоматически заменится на 0.

Рисунок 7

Под строкой “Введите значения вероятностей генерации банков” находятся названия банков, а справа от них, соответствующая каждой системе, строка ввода. В эти строки необходимо ввести значения вероятности генерации для каждой платежной системы. (рисунок. 8) Если оставить поле пустым или данные, введенные в него, будут некорректными, то вероятность автоматически заменится на 0.

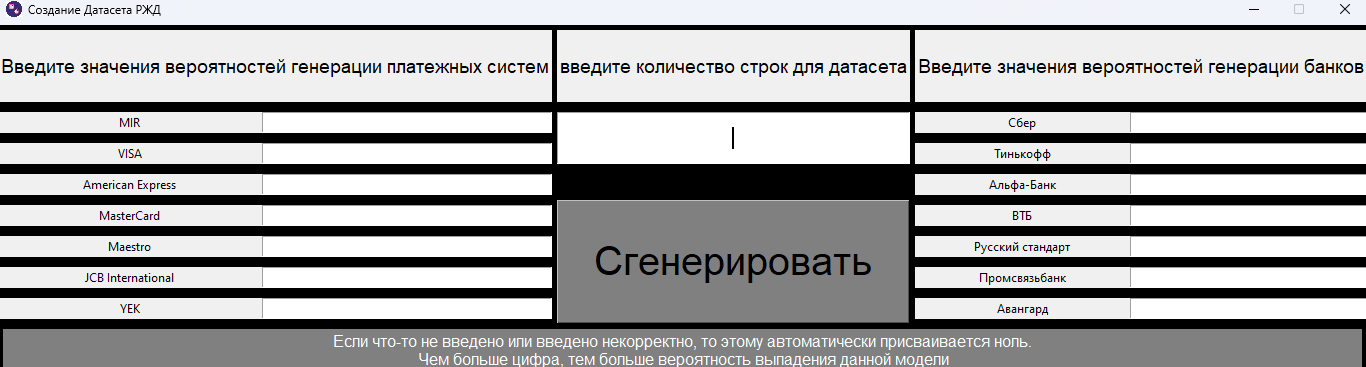


Рисунок 8

В поле расположенное под строкой “Количество строк в датасете” необходимо ввести количество строк, которые нужно сгенерировать для датасета (рисунок. 9). Если оставить поле пустым или данные, введенные в него, будут некорректными, то программа выдаст ошибку.

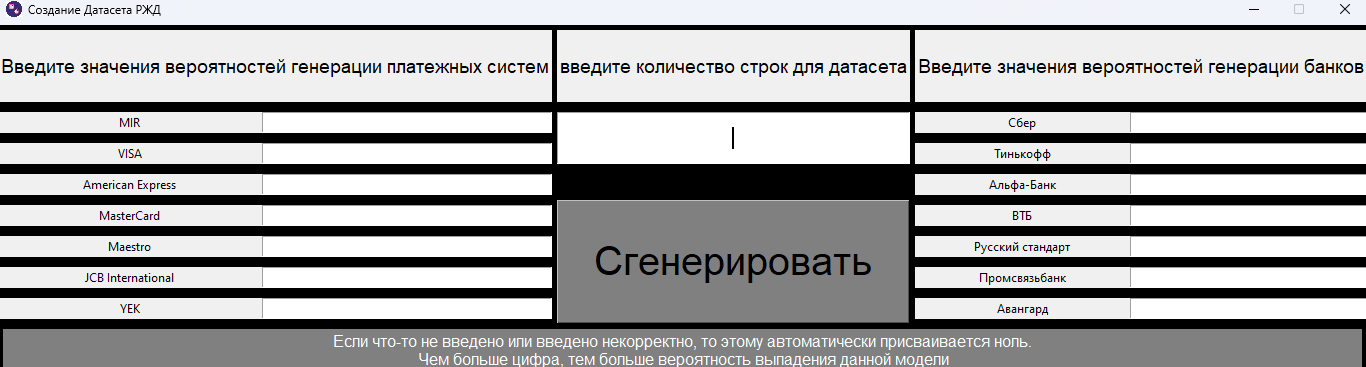
После того, как вы ввели данные, для генерации датасета необходимо нажать на кнопку “Сгенерировать” и дождаться уведомления об успехе. Excel файл будет сохранен в ту директорию, в которой находится программа.

Рисунок 9

# **Рекомендации для программиста**

Для внесения изменений в исходный код программы необходимо наличие устройство с установленное операционной системой Linux, macOS или Windows, а также среды разработки, поддерживающей запуск python версии 3.11.4

Исходный код программы доступен по ссылке ниже:

[study/dataset/main.py at main · ARnoD0rian/study (github.com)](https://github.com/ARnoD0rian/study/blob/main/dataset/main.py)

# **Контрольный пример**

В этом разделе представлен пример работы программы.

На рисунке 10 представлены параметры, установленные при генерации.

Рисунок 10. Меню программы

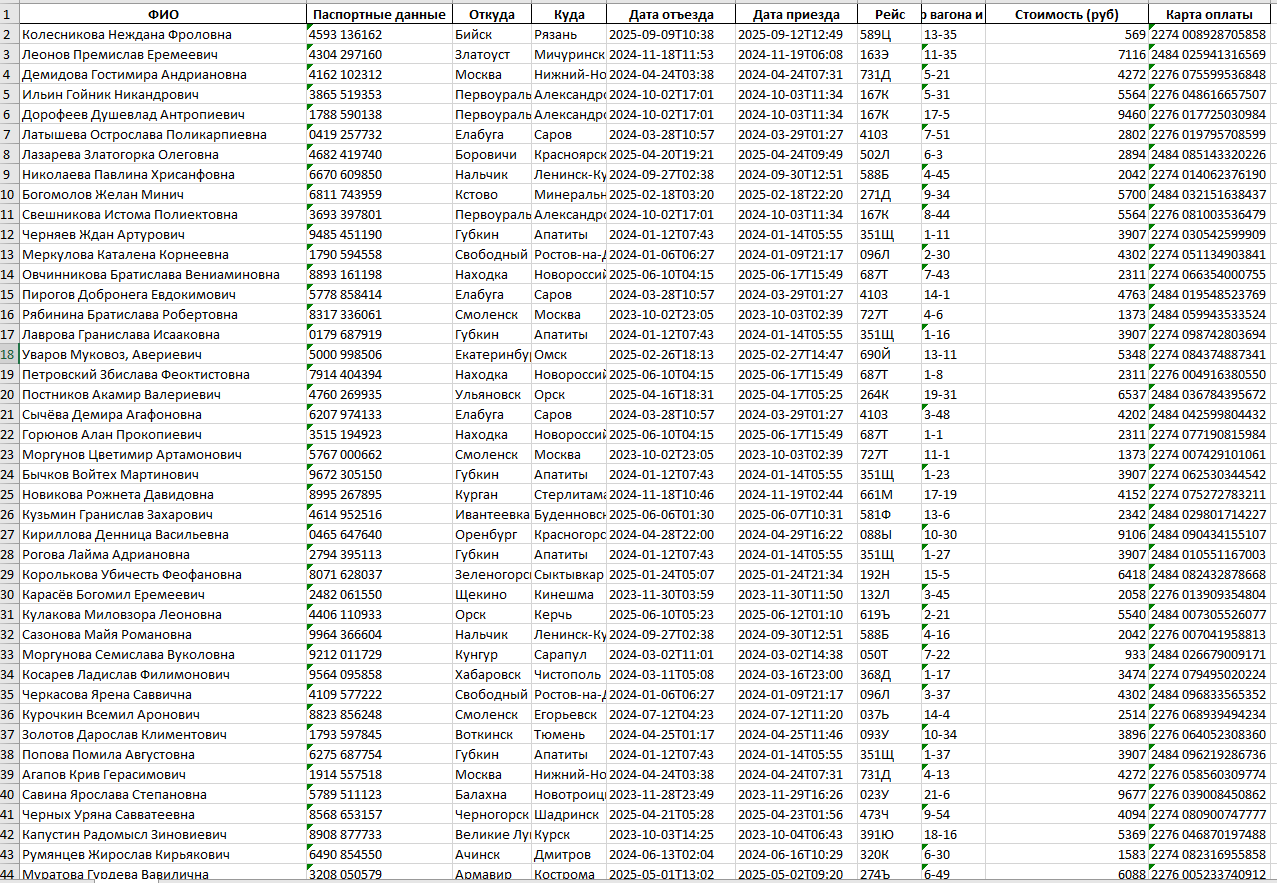
На рисунке 11 представлена часть данных, которые программа сгенерировала.

Рисунок 11. Результат работы программы

# **Вывод**

В ходе выполнения данной работы были изучены принципы генерации синтетических данных о покупке ЖД билетов, разработан алгоритм генерации синтетических данных данного типа и написана программа, которая при вводе определенных параметров генерирует датасет покупки ЖД билетов.

# Список использованной литературы

[1] Определение номера поезда: [(poezd.ru)](https://poezd.ru/text/poleznaya_informaciya/nomer_poezda.php)

[2] Определение расстояние на поверхности земли: [(kobzarev.com)](https://www.kobzarev.com/programming/calculation-of-distances-between-cities-on-their-coordinates/)