**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Программа бакалавриата**

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

**ОТЧЁТ**

**По лабораторной работе №1**

**По дисциплине “Алгоритмы и структуры данных”**

**На тему “Генерация синтетических данных”**

**Вариант №1**

**Студент гр. 23Б16-пу**

**Пушкарев Н.П.**

**Преподаватель**

**Дик А.Г.**

**Санкт-Петербург**

**2024 г.**

**Оглавление**

[**Цель работы 3**](#_1fob9te)

[**Описание задачи 3**](#_3znysh7)

[**Теоретическая часть 4**](#_2et92p0)

[**Описание программы 4**](#_tyjcwt)

[**Основные шаги программы 5**](#_3dy6vkm)

[**Блок схема программы 6**](#_1t3h5sf)

[**Рекомендации пользователя 8**](#_lnxbz9)

[**Рекомендации программиста 8**](#_35nkun2)

[**Контрольный пример 8**](#_1ksv4uv)

[**Вывод 9**](#_2jxsxqh)

# 

# 

# Цель работы

Целью данной лабораторной работы является разработка программы для автоматизированной генерации датасета, содержащего информацию о покупках в магазинах на территории Санкт-Петербурга. Программа должна учитывать следующие параметры: название магазина, координаты, время покупки, категории товаров, бренды, количество товаров, стоимость и данные о платежных картах.

# Описание задачи

Задача заключается в автоматизированной генерации датасета на 50000 строк, который будет содержать информацию о покупках в магазинах на территории Санкт-Петербурга. Датасет должен включать следующие ключевые атрибуты и соответствовать определённым требованиям:

1. **Название магазина:** Генерировать не менее 30 уникальных названий магазинов из заранее заданного "словаря".
2. **Координаты и время:** Формировать два столбца: дата и время (в формате YYYY-MM-DDTHH:MM+ZZ:ZZ) и долгота с широтой (с округлением до 10^-8). Время покупок должно соответствовать рабочему времени магазинов (с 10:00 до 22:00).
3. **Категории:** Категории товаров должны соответствовать тематике магазинов и быть не менее 50 уникальными, формируемыми из другого "словаря".
4. **Бренды:** Генерировать не менее 500 уникальных брендов, относящихся к выбранным категориям.
5. **Номер карточки:** Генерировать номера карт в формате "1234 5678 1234 5678", с возможностью настройки вероятностей для различных банков (например, Сбербанк, Газпромбанк) и платежных систем (Visa, MasterCard).
6. **Количество товаров:** Генерировать случайное количество товаров в диапазоне от 5 до 10 штук в каждой покупке.
7. **Стоимость:** Генерировать стоимость товаров, основываясь на средней рыночной стоимости, исключая бесплатные или нулевые значения.

# Теоретическая часть

Для решения задачи используется язык программирования Python. В программе применяются следующие подходы:

- Чтение данных из файлов: используются заранее подготовленные файлы с данными о магазинах, категориях и брендах.

- Генерация случайных данных: выбор магазинов, категорий, брендов и количества товаров осуществляется случайным образом с использованием функции random.choice() и других методов генерации.

- Генерация номеров карт: функция create\_card() генерирует номера карт с учетом вероятностей выбора банка и платежной системы.

- Реалистичная оценка стоимости товаров: цены для различных категорий товаров рассчитываются с использованием функции realistic\_pricing(), которая моделирует распределение цен с учетом рыночных значений.

- Запись данных в файл: сгенерированные данные сохраняются в файл Excel с помощью библиотеки pandas.

- Перевод xlsx в xml: используется библиотеки xml.etree.ElementTree и ast, для корректного представления координат (долготы и широты).

1. main.py: Генерация датасета с магазинами и товарами с координатами различных магазинов, временем покупки, различными брендами и стоимостью. Создает файл output.xlsx.

2. get\_coordinates.py: Генерация координат магазинов с использованием запросов к сервису nominatim.

3. to\_xml.py: Перевод эксель-файла в xml .

Ограничения:

• Количество строк в датасете ограничивается вводом пользователя.

• У магазинов реальные координаты (округление до 8 знаков).

• Бренды и товары соответствуют друг другу.

• Стоимость товара — среднее значение.

• Веса банков и платежных систем определяются пользователем и должны в сумме давать число больше нуля.

• Логика выбора карт оплаты с ограничением на 5 повторов.

# Описание программы

файл main.pyб, функции:

Таблица 1. Описание функций

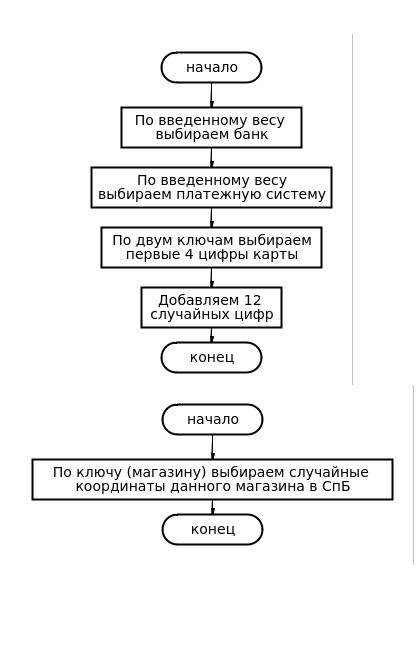
| Название функции | Описание |
| --- | --- |
| create\_card | генерация номера карты, с учетом процентного распределения банков и платёжных систем |
| generate\_random\_data | генерация случайных номеров строк в словарях магазинов, брендов и товаров с учетом их соответствия друг другу |
| get\_random\_coordinates | выбор одного случайного местоположения для магазина из словаря координат |
| pick\_random\_time | генерирует время в пределах 10-21 |
| random\_item\_count | генерация случайного количества товаров от 5 до 100 штук |
| realistic\_pricing | генерация цены товаров по принципу среднего между минимальной и максимальной для конкретной позиции |
| select\_bank\_and\_variant | функция для выбора карт и платежных систем, исходя из введенных пользователем весов |

# Основные шаги программы

1. Запуск программы (main.py):
2. Пользователь вводит веса банков и платежных систем.
3. Генерируются выбранное число записей
4. Данные собираются и записываются в файл output.xlsx.

# Блок схема программы







# Рекомендации пользователя

* Для запуска программы, откройте файл main.py в pycharm и нажмите Shift+F10
* Настройте весовое распределение банковских данных и платежных систем, убедившись что в сумме получается больше нуля
* Дождитесь пока в консоль не будет выведено время работы, т.е. окончание работы программы
* В итоге будет создан файл output.xlsx
* Запустите программу to\_xml.py
* В итоге будет создан data.xml

# Рекомендации программиста

* Поддерживайте актуальность используемых библиотек: pandas и xml.etree.ElementTree и версии Python, не менее Python 3.11 для сохранения актуальности и работоспособности кода на современных системах.
* Организуйте код в соответствии с best practices, следите за четким именованием переменных и функций.

# Контрольный пример

Запускаем файл main.py; вводим вероятностные веса банков (в сумме должно быть больше нуля); вводим вероятностные веса платёжных систем (в сумме должно быть больше нуля); ждём окончания работы программы; в итоге будет сгенерирован файл output.xlsx; запускаем to\_xml.py, в итоге будет сгенерирован файл data.xml, который можно открыть в любом браузере

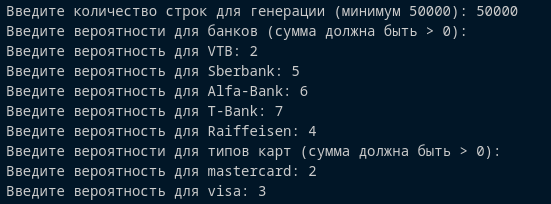


Рис 3. Пример ввода данных

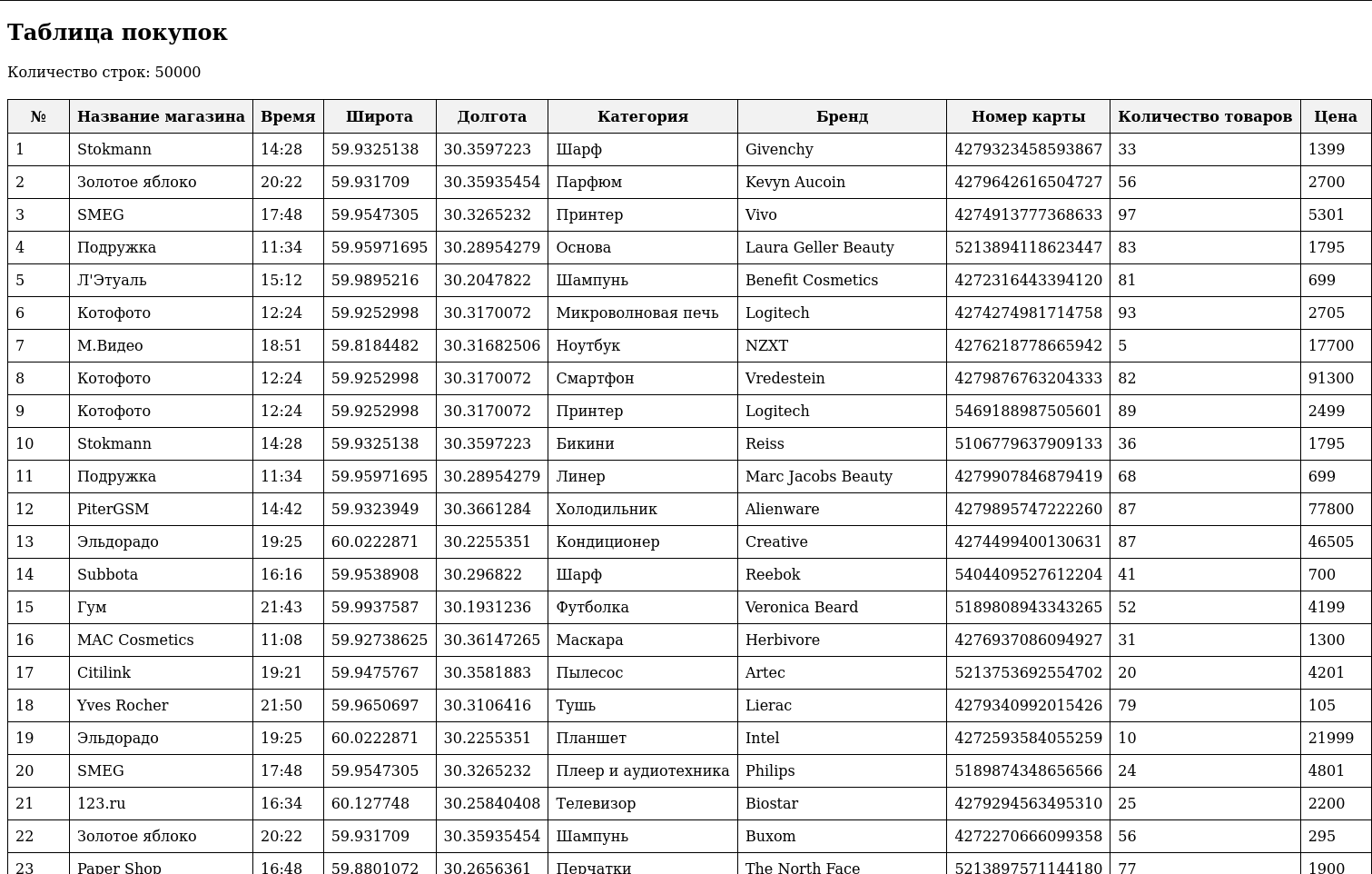


Рис 4. Пример сгенерированного датасета

# 

# Вывод

В ходе выполнения лабораторной работы была создана программа для автоматической генерации датасета о покупках. Программа учитывает различные параметры, включая магазины, координаты, бренды и данные о платежных картах. Итоговый датасет может быть использован для анализа покупательского поведения или других исследовательских задач. Программа выполняется за приемлемое время и генерирует данные с высокой степенью реалистичности.