**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 – 1**

**Студент гр. 23Б15-пу**

**Бек В.А.**

**Преподаватель**

**Дик А.Г.**

**Санкт-Петербург**

**2024 г.**

Оглавление

1. [Цель работы 3](#_Toc178936363)
2. [Описание задачи (формализация задачи) 4](#_Toc178936364)
3. [Теоретическая часть 5](#_Toc178936365)

[**Краткое описание создания датасета** 5](#_Toc178936366)

[**Ограничения датасета** 7](#_Toc178936367)

1. [Блок схема программы 8](#_Toc178936368)
2. [Описание программы 11](#_Toc178936369)

[**Программа использует следующие данные** 11](#_Toc178936370)

[**Структура программы** 12](#_Toc178936371)

1. [Рекомендации пользователя 13](#_Toc178936372)
2. [Рекомендации программиста 14](#_Toc178936373)

[**Код программы** 15](#_Toc178936374)

1. [Контрольный пример 16](#_Toc178936375)
2. [Вывод 18](#_Toc178936376)
3. [Источники 19](#_Toc178936377)

# Цель работы

Цель данной работы заключается в разработке программы для генерации большого объёма данных (датасета), имитирующего сценарий покупки в магазине. Датасет должен включать различные атрибуты, такие как название магазина, координаты, категории товаров, бренды, номер платёжной карты, количество покупок и их стоимость. Программа должна быть гибкой и настраиваемой, предоставляя возможность изменять параметры, такие как список магазинов, категории товаров, бренды, а также вероятность использования конкретных банков и платёжных систем.

# Описание задачи (формализация задачи)

Задача заключается в создании датасета, моделирующего покупки в магазинах Санкт-Петербурга, с учетом следующих требований:

1. **Название магазина**: используется список реальных магазинов, сгруппированных по категориям товаров.
2. **Координаты магазина**: реальные координаты магазинов с округлением до 8 знаков после запятой (широта и долгота).
3. **Категория товара**: товары выбираются в зависимости от профиля магазина (электроника, продукты и т.д.).
4. **Бренд товара**: для каждой категории предусмотрен список брендов, из которых выбирается случайным образом.
5. **Количество товаров**: на одну покупку должно приходиться от 5 до 10 товаров.
6. **Стоимость**: цена товара зависит от категории и может колебаться на ±10% от среднего значения по этой категории.
7. **Номер карты**: генерируются банковские карты на основе российских BIN-кодов с привязкой к платёжной системе (МИР, Visa, MasterCard). Используются карты следующих банков: **Сбербанк**, **ВТБ**, **Тинькофф**, **Газпромбанк**, **Альфа-Банк**. Одну карту можно использовать не более 5 раз.
8. **Дата и время транзакции**: случайные значения в пределах рабочего времени магазина.
9. **Количество строк**: не менее 50 000 записей.

Датасет должен обеспечивать гибкость настройки, включая возможность регулировки вероятностей использования различных банков и платёжных систем, а также соответствовать всем заданным ограничениям. Результат сохраняется в формате для Excel с возможностью последующей обработки данных.

# Теоретическая часть

Для создания датасета используются различные методы и библиотеки, обеспечивающие гибкость и эффективность процесса генерации данных.

Основные подходы включают:

* **Случайная выборка**: Применяется для выбора значений атрибутов, таких как названия магазинов, категории товаров и бренды. Используется библиотека **random** для генерации случайных чисел и выборок, что позволяет избежать однообразия в датасете.
* **Вероятностные распределения**: Для настройки вероятности выбора банков и платежных систем используются вероятностные модели. Библиотека **random** позволяет задать вес для каждой опции, обеспечивая возможность регулирования вероятности использования различных банков.
* **Генерация временных данных**: Для генерации временных данных в пределах рабочего времени магазинов используется библиотека **datetime**. Это позволяет создавать случайные временные метки, соответствующие рабочим часам магазинов.

### **Краткое описание создания датасета**

Данный датасет предназначен для анализа покупок в различных магазинах. Он включает в себя следующие ключевые шаги процесса генерации данных:

1. **Запуск программы (shopping\_dataset\_creator.py):**
   * Инициализация программы и загрузка необходимых библиотек.
2. **Загрузка данных:**
   * Импорт координат магазинов и времени их работы из файла store\_coordinates.txt, содержащего информацию о магазинах, их местоположении и времени работы.
   * Загрузка категорий и брендов из файла brands.txt, содержащего данные о товарных категориях и их средних ценах.
3. **Создание словарей:**
   * Создание словарей:
     + В котором указаны доступные товарные категории для каждого магазина.
     + В котором указаны BIN коды разных банков
     + В котором указаны платёжные системы.
4. **Загрузка и установка вероятностей:**
   * Ввод пользователем вероятностей для банков и платежных систем с проверкой корректности сумм, чтобы общая вероятность составляла 100.
5. **Ввод количества строк для генерации:**
   * Пользователь вводит желаемое количество строк для генерации, с минимальным значением 50 000.
6. **Генерация данных о покупках:**
   * Процесс включает:
     + Случайный выбор магазина.
     + Генерацию времени посещения магазина.
     + Случайный выбор товарной категории и бренда.
     + Генерацию номера платежной карты с учетом вероятностей для банков и платежных систем.
     + Расчет стоимости товаров на основе случайных цен и количества.
7. **Запись данных в файл:**
   * Сохранение сгенерированных данных в файл shopping\_dataset.xlsx с установленными значениями ширины колонок для удобного просмотра.
8. **Завершение работы программы:**
   * Завершение работы программы и уведомление пользователя о успешной генерации датасета.

### **Ограничения датасета**

При создании датасета для имитации покупок в магазинах необходимо учитывать ряд ограничений, чтобы гарантировать реалистичность и качество сгенерированных данных. Эти ограничения включают:

* **Общее количество строк**: Минимальное количество записей в датасете должно составлять не менее 50 000.
* **Словарь магазинов**: Название магазинов должно формироваться на основе "словаря", который должен включать как минимум 30 различных магазинов. Это гарантирует разнообразие в представленных данных и реалистичность сценариев покупок.
* **Координаты**: Координаты (широта и долгота) должны округляться до 8 знаков после запятой. Также необходимо учитывать, что даты и время транзакций должны соответствовать реальному времени посещения магазина.
* **Категории товаров**: "Словарь" категорий товаров должен включать как минимум 50 категорий. При этом категории, товары и их бренды должны соответствовать магазину, в котором они продаются.
* **Бренды**: "Словарь" брендов должен содержать не менее 500 различных брендов.
* **Номера карточек**: Генерируемые номера карточек должны ограничиваться максимальным количеством повторов, равным 5.
* **Количество товаров в транзакции**: Минимальное количество товаров, входящих в одну покупку, должно составлять не менее 5 штук.
* **Стоимость товаров**: Данные о стоимости не могут содержать бесплатных или пустых значений. Если в процессе генерации возникают такие случаи, необходимо выводить сообщение об ошибке и корректировать данные.

# Блок схема программы

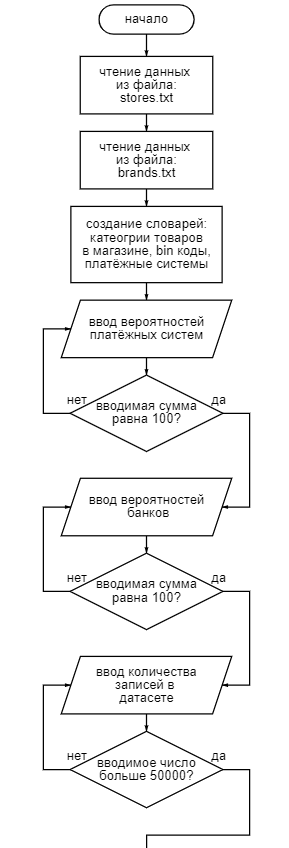


Рис 1. Блок-схема программы (1)

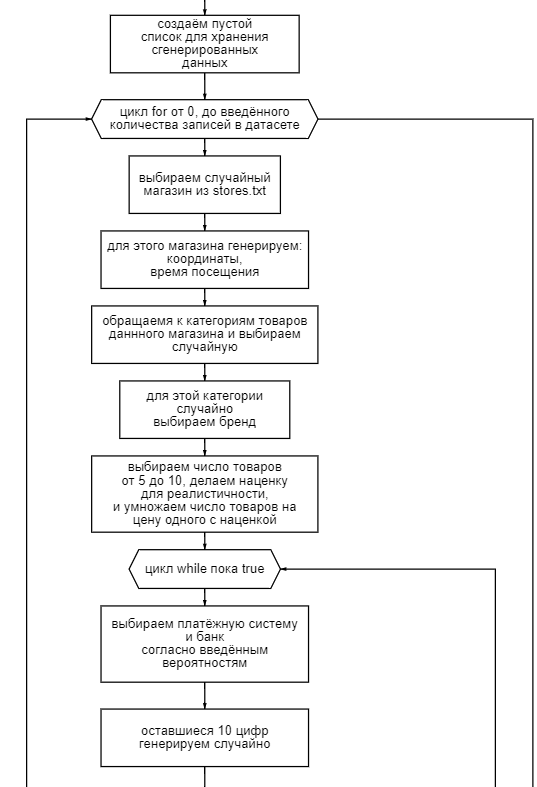


Рис 2. Блок-схема программы (2)

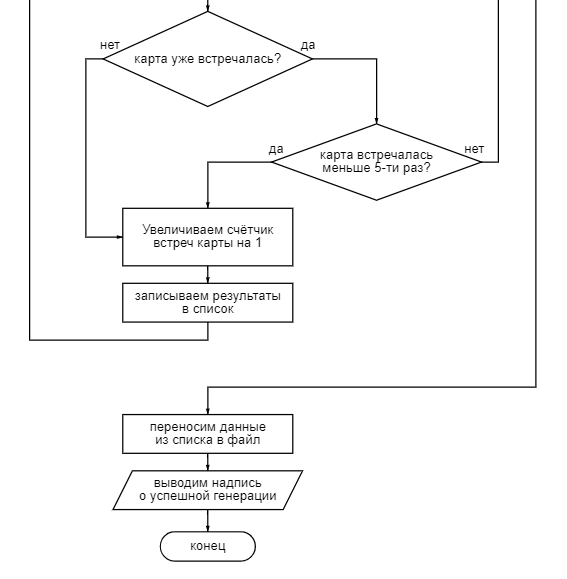
****

Рис 3. Блок-схема программы (3)

# Описание программы

В данном проекте реализована программа на языке Python (версия 3.11.9), которая генерирует набор данных для симуляции покупок в различных магазинах. Программа использует библиотеки random, datetime и openpyxl для работы с данными и создания Excel-файлов.

Таблица 1. shopping\_dataset\_creator.py

|  |  |  |
| --- | --- | --- |
| Функция | Описание | Возвращаемое значение |
| generate\_datetime | Генерация случайного времени посещения магазина между открытием и закрытием. | str |
| generate\_card\_number | Генерация номера карты на основе кода BIN и кода платёжной системы. | str |
| generate\_store\_data | Генерация данных о магазине, включая категорию, бренд и информацию о платёжной карте. | list |
| get\_probabilities | Запрос вероятностей для платёжных систем и банков от пользователя. | list |

### **Программа использует следующие данные**

* **Магазины**: Информация о магазинах загружается из файла stores.txt и включает название магазина, координаты, время открытия и закрытия.
* **Категории и бренды**: Загрузка данных из файла brands.txt, который содержит категории товаров, соответствующие бренды и среднюю стоимость.
* **Коды BIN**: Предоставлены для различных банков, чтобы использовать их при генерации номеров карт.
* **Платёжные системы**: Включает информацию о различных платёжных системах и их кодах.

### **Структура программы**

* **Модуль работы с магазинами**: Отвечает за загрузку и обработку данных о магазинах.
* **Модуль работы с брендами**: Отвечает за загрузку данных о категориях и брендах.
* **Модуль генерации данных**: Основной модуль, который отвечает за генерацию временных меток, номеров карт и данных о магазинах.
* **Модуль пользовательского ввода**: Отвечает за запрос вероятностей от пользователя и количество строк для генерации данных.

# Рекомендации пользователя

Для успешного использования данной программы по генерации данных о покупках, пользователю рекомендуется следовать следующим инструкциям:

1. **Настройка вероятностей:**

* При запуске программы вам будет предложено ввести вероятности для каждой платёжной системы и банка. Убедитесь, что сумма всех вероятностей равна 100 и все значения неотрицательные.

1. **Выбор количества строк:**

* При вводе количества строк для генерации, обратите внимание на минимальное значение, равное 50,000. Программа не будет работать, если введено меньшее количество.

1. **Открытие и использование файла результата:**

* После завершения генерации, результат будет сохранён в файле shopping\_dataset.xlsx. Для удобного просмотра откройте его в программе Microsoft Excel или другой совместимой программе для работы с электронными таблицами.

Следуя этим рекомендациям, вы сможете максимально эффективно использовать программу для генерации данных о покупках.

# Рекомендации программиста

**Установка необходимых библиотек:**

* Перед запуском программы убедитесь, что все необходимые библиотеки установлены. В данном случае вам потребуется библиотека openpyxl для работы с Excel файлами. Установите её с помощью команды:



**Подготовка файлов данных:**

* Убедитесь, что файлы stores.txt и brands.txt содержат корректные данные в ожидаемом формате:
* stores.txt должен содержать информацию о магазинах, включая название, координаты (широта и долгота), время открытия и закрытия. Каждая запись должна быть в отдельной строке и разделена запятыми.
* brands.txt должен включать категории товаров и соответствующие бренды с указанием средней цены. Каждая категория должна быть отделена двоеточием, а бренды перечислены через точку с запятой.

**Код программы:**

**<https://github.com/Kliooo/Algorithms-and-data-structures>**

# Контрольный пример

* Запуск программы

Для запуска программы используйте файл shopping\_dataset\_generation.py. Эта программа предназначена для генерации набора данных о покупках на основе заданных параметров, таких как магазины, категории товаров, бренды и платежные системы.

* Ввод весов платежных систем и банков

После запуска программы пользователю будет предложено ввести вероятности для различных платежных систем (Рис. 4) и банков (Рис. 5). Эти веса определяют вероятность выбора конкретной платежной системы или банка при генерации данных о покупках.

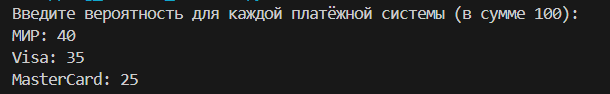


Рис 4. пример ввода весов платежных систем

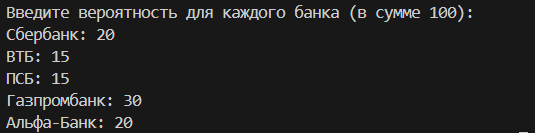


Рис 5. пример ввода весов банков

* Ввод количества записей

После успешного ввода весов, программа предложит пользователю указать количество записей для генерации (Рис. 6). Минимальное количество записей, которое можно сгенерировать, составляет 50,000 (Рис. 7).



Рис 6. пример ввода количества записей

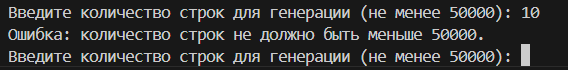


Рис 7. пример ввода количества записей меньше 50000

* Генерация данных

После ввода количества записей программа начинает процесс генерации данных о покупках, используя информацию о магазинах, категориях и брендах, а также введенные вероятности. Сгенерированные данные сохраняются в файл shopping\_dataset.xlsx.

* Завершение работы

По завершении генерации пользователь увидит сообщение о том, что данные успешно сохранены (Рис. 8).



Рис 8. Завершение программы

# Вывод

В ходе выполнения данной работы была разработана программа, предназначенная для генерации большого объема данных, моделирующего сценарии покупок в магазинах. Программа успешно реализует основные требования, включая:

* **Создание реалистичного датасета**: С использованием списков реальных магазинов и их категорий, программа генерирует записи, которые включают название магазина, координаты, категории товаров, бренды, номера карт, количество товаров и их стоимость.
* **Корректное генерирование данных**: Реализованные алгоритмы обеспечивают корректное формирование всех атрибутов записей. Например, стоимость товаров варьируется в пределах ±10% от средней цены, а номера карт формируются на основе актуальных российских BIN-кодов.
* **Удобный экспорт данных**: Сгенерированные данные сохраняются в формате Excel, что обеспечивает удобный доступ и возможность последующей обработки в различных программах.
* **Пользовательский интерфейс для настройки вероятностей**: Внедрение функций, позволяющих пользователю вводить вероятности для банков и платежных систем, добавляет интерактивности и позволяет гибко управлять процессом генерации.

# Источники

* iBankie — Банковские карты и их классификация*.*

[*https://ibankie.com/ru/*](https://ibankie.com/ru/)

*дата обращения: (25.09.2024)*

* random — Генерация псевдослучайных чисел*.* [*https://docs.python.org/3/library/random.html*](https://docs.python.org/3/library/random.html)

*дата обращения: (25.09.2024)*

* datetime — Работа с датами и временем.[*https://docs.python.org/3/library/datetime.html*](https://docs.python.org/3/library/datetime.html)

*дата обращения: (25.09.2024)*

* openpyxl — Чтение/запись файлов Excel с использованием Python*.* [*https://openpyxl.readthedocs.io/en/stable/*](https://openpyxl.readthedocs.io/en/stable/)

*дата обращения: (25.09.2024)*

* Редактор блок-схем.

[*https://programforyou.ru/block-diagram-redactor*](https://programforyou.ru/block-diagram-redactor)

*дата обращения: (25.09.2024)*