**С****АНКТ-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**

**Черевко М.Е.**

**Преподаватель**

**Дик А.Г.**

**Санкт-Петербург**

**2024 г.**

Оглавление

1. [Цель работы 3](#_Toc178969255)
2. [Описание задачи (формализация задачи) 4](#_Toc178969256)
3. [Теоретическая часть 5](#_Toc178969257)
4. [Ход программы 6](#_Toc178969258)
5. [Блок схема программы 7](#_Toc178969259)
6. [Описание программы 10](#_Toc178969260)

[**Программа использует следующие данные** 11](#_Toc178969261)

[**Структура программы** 11](#_Toc178969262)

1. [Рекомендации пользователя 12](#_Toc178969263)
2. [Рекомендации программиста 13](#_Toc178969264)

[**Код программы** 13](#_Toc178969265)

1. [Контрольный пример 14](#_Toc178969266)
2. [Вывод 16](#_Toc178969267)
3. [Источники 17](#_Toc178969268)

# 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, его координаты, категории товаров этого магазина и соответствующие бренды, номер платёжной, карты через которую была совершена оплата, число покупок и общая стоимость. Можно задавать вероятность использования конкретных банков и платёжных систем.

# Описание задачи (формализация задачи)

Необходимо создать датасет, моделирующий покупки в магазинах Санкт-Петербурга, с учетом данных требований:

* **Название магазина**: используется список реальных магазинов,
* **Координаты магазина**: реальные координаты магазинов (широта и долгота).
* **Категория товара**: категории и товары подбираются в зависимости от профиля магазина (косметика, продукты и т.д.).
* **Бренд товара**: для каждой категории товаров сопоставлен список действующих брендов, один из которых выбирается случайным образом
* **Количество товаров**: на одну покупку приходится от 5 до 10 товаров.
* **Стоимость**: для каждой категории есть свой диапазон цен, в пределах которого генерируется стоимость товара
* **Номер карты**: Используются карты следующих банков: **Сбербанк**, **ВТБ**, **Тинькофф**, **Альфа-Банк**, ПСБ. Одну карту можно использовать не более 5 раз. Банковские карты генерируются на основе российских BIN-кодов с привязкой к платёжной системе (МИР, Visa, MasterCard).
* **Дата и время транзакции**: случайные значения в пределах рабочего времени магазина.
* **Банк: Сбербанк**, **ВТБ**, **Тинькофф**, **Альфа-Банк**, ПСБ
* **Платёжная система**: МИР, Visa, MasterCard.
* **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 xlsx или csv 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

**Ограничения**

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.

# Ход программы

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

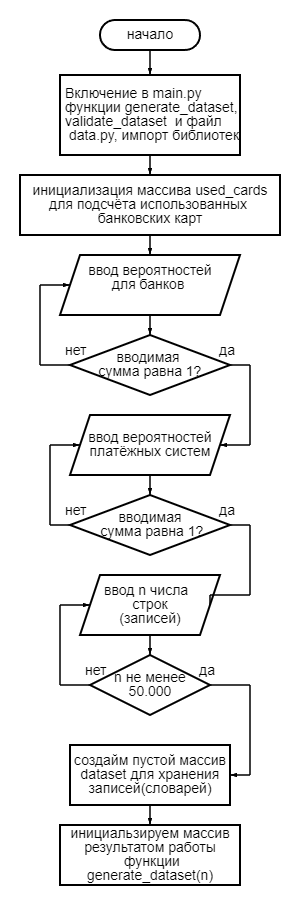
1. Инициализация программы и загрузка необходимых библиотек**, в**ключение в **main.py** функцию **generate\_datase**t из файла **data\_genegetion.py,** содержащего рабочий код и файл **data.py, хранящиц** данные для генерации
2. Ввод пользователем вероятностей для банков и платежных систем, ввод количества строк(записей) для генерации (не менее 50.000)
3. Инициализация массива **dataset**
4. Генерация функцией **generate\_purchase\_row()** словаря с данными и добавление его в конец массива **dataset**

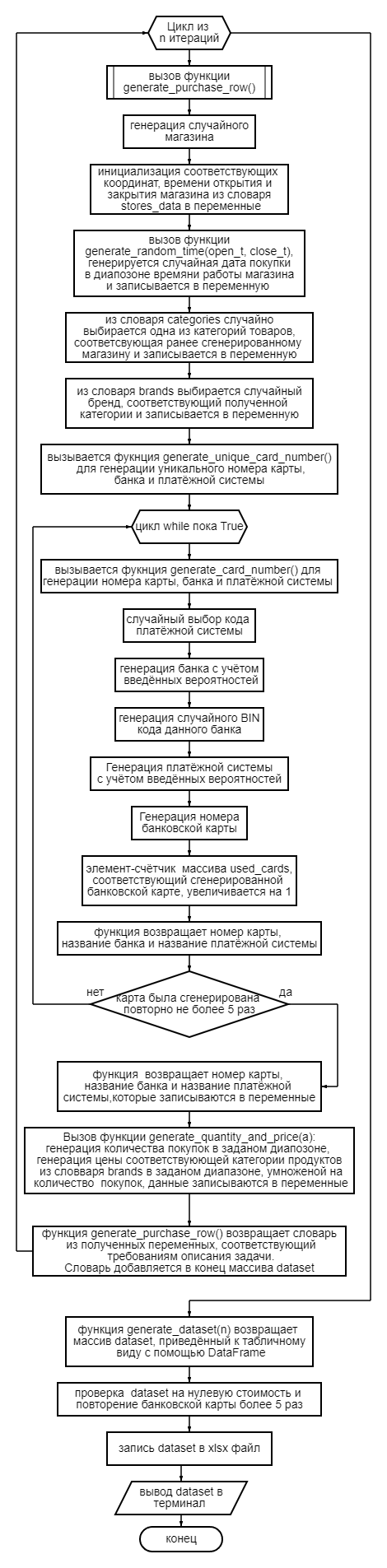
Цикл генерации включает:

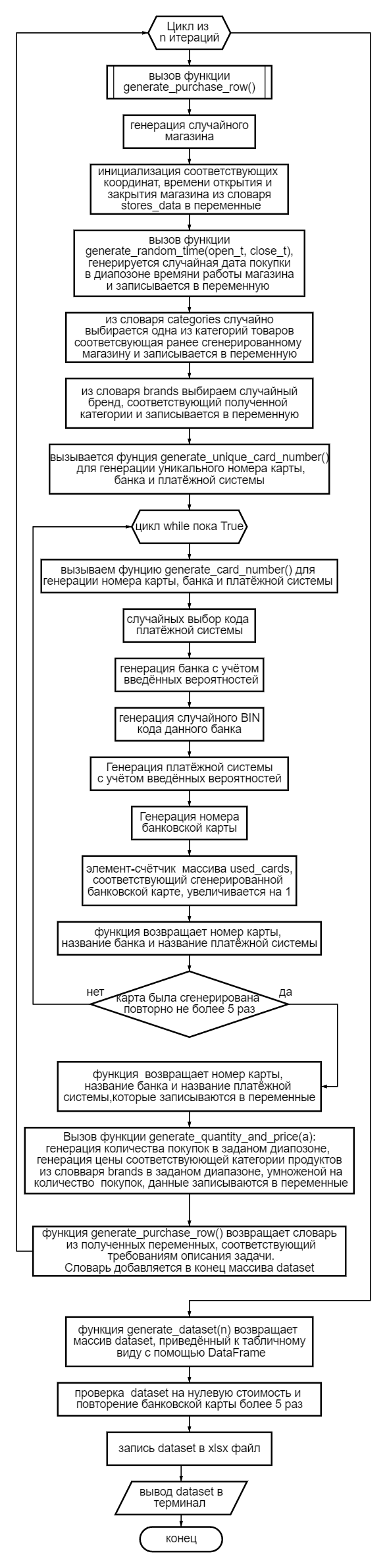
* + - Случайный выбор магазина.
    - Генерацию времени посещения магазина.
    - Случайный выбор товарной категории и бренда.
    - Генерацию номера платежной карты с учетом вероятностей для банков и платежных систем.
    - Расчет стоимости товаров на основе случайных цен и количества.
    - Создание словаря и добавление в конец массива **dataset**

1. **Завершение заполнения массива dataset. Проверка массива на правильность генерации(соответствие ограничениям) функцией validate\_dataset(dataset).**
2. **Уведомление об успешной генерации в терминале и запись данных в файл dataset.xlsx.**
3. Вывод массива **datase**t в терминал.
4. Завершение программы.

# Блок схема программы





**

# 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, openpyxl, pandas и collections для работы с данными и создания Excel-файлов.

Таблица dataset\_functions

|  |  |  |
| --- | --- | --- |
| Функция | Описание | Возвращаемое значение |
| generate\_random\_time | 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, str, str |
| generate\_unique\_card\_number | Проверка номера карты на повторную генерацию более 5 раз | str, str, str |
| generate\_quantity\_and\_price | Генерация случайного количества покупок и цены в заданном диапазоне | int, int |
| generate\_purchase\_row | Генерация итогового словаря(записи) соответствующий требованиям описания задаччи | list |
| generate\_dataset | Добовляет в конец массива полученные словари(записи) | DataFrame |
| validate\_card\_numbers | Проверка на повторение карты более 5 раз в массиве | ValueError |
| validate\_price | Проверка на нулевую стоимость | ValueError |
| validate\_dataset | Запуск validate\_price и validate\_card\_numbers |  |

### **Программа использует следующие данные:**

**Все данные загружаются из файла data.py.**

* **Магазины**: список магазинов (stores)
* **Банки**: список банков (banks)
* **Коды BIN**: Предоставлены для различных банков, чтобы использовать их при генерации номеров карт. (bin\_codes)
* **Платёжные системы**: Включает информацию о различных платёжных системах и их кодах. (payment\_systems), (payment\_systems\_codes)
* **Координаты и время работы магазинов** – словарь где каждому магазину сопоставлены координаты и время (stores\_data)
* **Категории**: словарь где каждому названию магазину сопоставлен список категорий товаров (categories)
* **Бренды:** словарь где каждой категории товаров сопоставлен список её брендов (brands)

### **Структура программы**

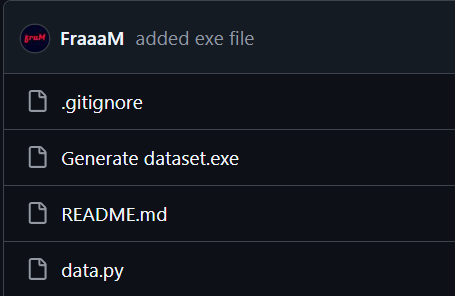
* **Модуль работы с магазинами**: Отвечает за загрузку и обработку данных о магазинах.
* **Модуль работы с брендами**: Отвечает за загрузку данных о категориях и брендах.
* **Модуль генерации данных**: Основной модуль, который отвечает за генерацию временных меток, номеров карт и данных о магазинах.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

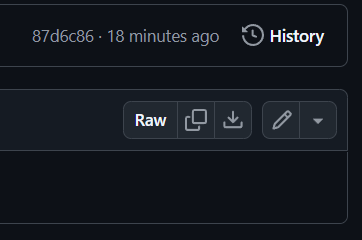
# Рекомендации пользователя

**Инструкция к запуску программы:**

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

**1. Перейти по ссылке:** [**https://github.com/FraaaM/Data\_set\_generation**](https://github.com/FraaaM/Data_set_generation)

**2. Нажать на Generate data.exe, перейдя на другую страницу**

**3. В правом верхнем углу нажать на окно установки файла**

**4. После установки запустить приложение и придерживаться последующим условиям генерации (пример взаимодействия с работающей программой представлен на страницах 14-16).**

**5. Вы сгенерировали таблицу dataset.xlsx с заданными данными.**

# Рекомендации программиста

* **Актуальность библиотек и языка программирования:**

Регулярно обновляйте используемые библиотеки и версии Python, чтобы обеспечить совместимость программы с современными операционными системами и избежать проблем с работоспособностью.

* **Организация кода:**

Структурируйте код, разделяя его на логически обоснованные модули и функции. Это упростит дальнейшую разработку и поддержку программы.

* **Документирование:**

Создавайте и поддерживайте документацию по проекту, чтобы другие разработчики могли быстро понять его структуру и функциональность. Это облегчает совместную работу и сопровождение кода.

### **Код программы**

**https://github.com/FraaaM/Data\_set\_generation\**

# Контрольный пример

* Запуск программы

Для запуска программы используйте exe файл Generate dataset.exe или через Visual Studio Code откройте папку с программой и запустите main.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 банков (Рис. 4) и различных платежных систем (Рис. 5). Эти веса определяют вероятность выбора конкретной платежной системы или банка при генерации данных о покупках.

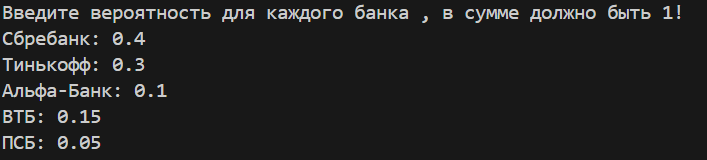


Рис. 4. пример ввода весов банков

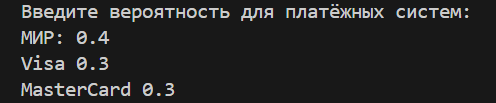


Рис. 5. пример ввода весов платежных систем

* Ввод количества записей

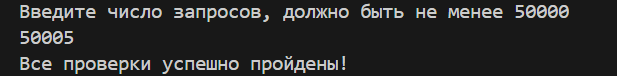
****После успешного ввода весов, программа предложит пользователю указать количество записей для генерации(Рис 6). Минимальное количество записей, которое можно сгенерировать, составляет 50,000. Если число записей меньше 50.000 то необходимо ввести заново (Рис 7).

Рис. 6 ввод числа записей для генерации

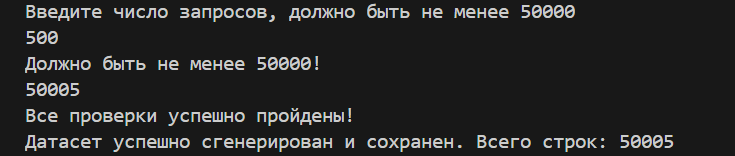


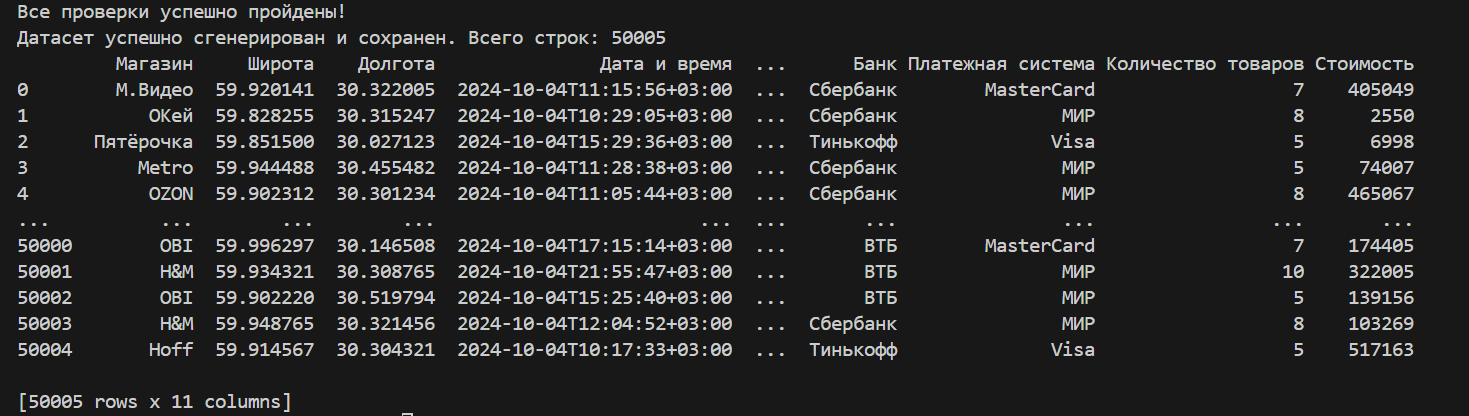
Рис 7. пример ввода количества записей меньше 50000

* 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 dataset.xlsx.

* Завершение работы

По завершении генерации пользователь увидит сообщение о том, что данные успешно сохранены и выведет массив в терминал (Рис. 8).

Рис 8. Завершение программы и вывод массива

# 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, дату и время, категории товаров, бренды, номера карт, наименование банка, платёжной системы, количество товаров и их стоимость.
* **Частичная настраиваемость**: Пользователи имеют возможность настраивать вероятность использования различных банков и платежных систем, задавать число генерируемых строк (записей).
* **Корректное генерирование данных**: Реализованные алгоритмы обеспечивают корректное формирование всех атрибутов записей. Например, стоимость товаров варьируется в соответствующем диапазоне, а номера карт формируются на основе актуальных российских BIN-кодов.
* **Удобный экспорт данных**: Сгенерированные данные сохраняются в формате Excel,
* **Добавлен консольный интерфейс для настройки вероятностей**.
* **Добавлен exe файл для быстрого и простого запуска программы**

# Источники

iBankie*.* Банковские карты и их классификация.[*https://ibankie.com/ru/*](https://ibankie.com/ru/)

pandas – обработка и анализ данных*. https://pandas.pydata.org/*

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

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

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

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

pyinstaller – программа для создания exe файла. *https://pyinstaller.org/en/stable/*

Collections – контейнеры***.*** *https://docs.python.org/3/library/collections.html*