**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 – 2**

**Студент гр. 23Б15-пу**

**Сериков К.Г.**

**Преподаватель**

**Дик А.Г.**

**Санкт-Петербург**

**2024 г.**

Оглавление

1. [Цель работы 3](#__RefHeading___1)
2. [Описание задачи (формализация задачи) 3](#__RefHeading___2)
3. [Теоретическая часть 4](#__RefHeading___3)
4. [Основные шаги программы 5](#__RefHeading___4)
5. [Блок схема программы 6](#__RefHeading___5)
6. [Описание программы 7](#__RefHeading___6)
7. [Рекомендации пользователя 8](#__RefHeading___7)
8. [Рекомендации программиста 8](#__RefHeading___8)
9. [Исходный код программы 8](#__RefHeading___9)
10. [Контрольный пример 9](#__RefHeading___10)
11. [Вывод 11](#__RefHeading___11)
12. [Источники 11](#__RefHeading___12)

# Цель работы

Целью лабораторной работы является разработка системы генерации датасета для списка визитов к врачу с учетом определенных требований и условий. Датасет должен включать личные данные пациентов, информацию о симптомах, анализах и специалистах, а также данные о банковских картах.

# Описание задачи (формализация задачи)

Задача состоит в создании датасета для списка визитов к врачу со следующими требованиями:

1. **ФИО**: славянские имена и фамилии.
2. **Количество строк датасета**: не меньше 50.000.
3. **Паспортные данные**: русские, белорусские и казахские паспортные данные с уникальными значениями.
4. **СНИЛС:** уникальное значение, привязан к клиенту (ФИО и паспортным данным).
5. **Симптомы**: ”словарь” минимум из 5000 симптомов. Могут быть комбинации не более чем из 10 симптомов.
6. **Выбор врача:** ”словарь” должен состоять минимум из 50 врачей.
7. **Дата посещения врача:** повторное посещение врача возможно только спустя 24 часа после выдачи анализов.
8. **Анализы:** ”словарь” должен состоять минимум из 250 анализов. Могут быть комбинации не более чем из 5 анализов.
9. **Дата получения анализов**: через 24-72 часа после посещения.
10. **Стоимость**: свободный вариант генерации данных в рублях.
11. **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. main.py: Главный файл, отвечает за ввод данных, вызывает функции генерации из файла functions.py, создает файл dataset.csv и выводит туда данные.
2. functions.py: Содержит все функции для генерации данных (generate\_name, generate\_passport, generate\_snils, generate\_card, generate\_unique\_card, generate\_doctors\_data, generate\_analyses\_date, generate\_price, generate\_visit\_date).
3. names\_data.py: Содержит данные для генерации ФИО (мужские и женские имена, фамилии и отчества).
4. doctors\_data.py: Содержит данные для генерации врачей, симптомов и анализов. У каждого врача есть свои подходящие симптомы и анализы.
5. cards\_data.py: Содержит данные для генерации карт (список банков, BIN-кодов и платежных систем).

Ограничения:

* Количество строк в датасете ограничивается вводом пользователя, но минимальное количество сгенерированных строк будет 50000.
* ФИО пациентов только славянские.
* Паспортные данные уникальные и могут быть только российские, белорусские и казахские.
* Уникальность СНИЛС.
* Веса банков и платежных систем определяются пользователем.
* Одной и той же банковской картой можно платить не более пяти раз.

# Основные шаги программы

1. Запуск программы (main.py):
2. Пользователь вводит количество людей, веса банков и платежных систем.
3. Запускается генерация словаря уникальных пациентов.
   1. Генерация ФИО.
   2. Генерация паспортных данных.
   3. Генерация СНИЛС.
   4. Генерация карт с заданным распределением.
4. Сбор всех данных
   1. Случайный выбор уникального клиента.
   2. Генерация врача.
      1. Генерация симптомов для соответствующего врача.
      2. Генерация анализов для соответствующего врача.
   3. Генерация даты визита и даты получения анализов.
   4. Генерация стоимости.
5. Запись данных в ячейки.
6. Запись данных в файл dataset.csv.

# Блок схема программы

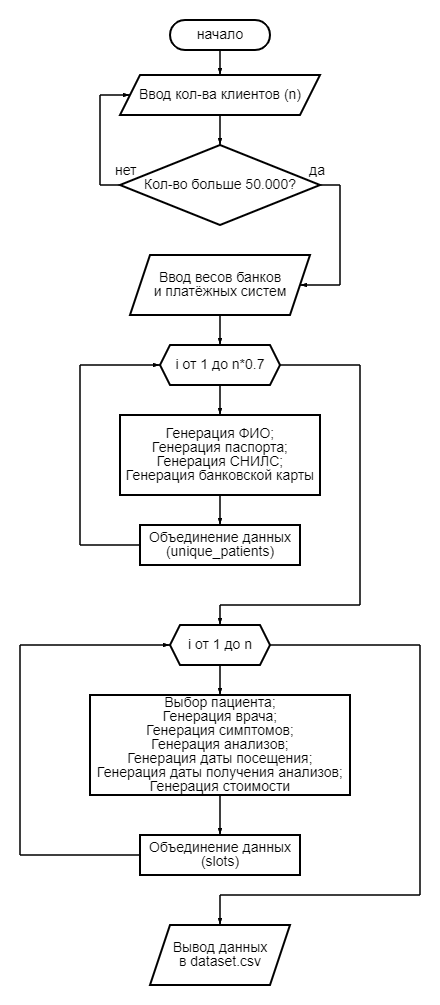


Рис 1. Блок-схема основной программы

# Описание программы

Программная реализация написана на языке Python 3.11.9 с использованием следующих библиотек: datetime [[1]](Datetime — The datetime module supplies classes for manipulating dates and times. // Python URL: datetime — Basic date and time types — Python 3.12.6 documentation (дата обращения: 03.10.2024).), random [[2]](#_Источники) и csv [[3]](#_Источники). Программа нацелена на генерацию данных для списка визитов к врачу. В процессе разработки программы использовались 5 файлов и 9 функций, каждая из которых имеет чётко определённое назначение:

Таблица 1. functions.py

|  |  |  |
| --- | --- | --- |
| Функция | Описание | Возвращаемое значение |
| generate\_name | Генерация ФИО. | str |
| generate\_passport | Генерация паспортных данных. | str |
| generate\_snils | Генерация СНИЛС. | str |
| generate\_doctors\_data | Генерация врача, симптомов и анализов. | list |
| generate\_visit\_date | Генерация даты и времени визита врача. | datetime |
| generate\_analyses\_date | Генерация даты получения анализов. | datetime |
| generate\_cards | Генерация банка, системы платежей и номера карты. | list |
| generate\_unique\_card | Проверка ограничение на использование карты не более 5 раз. | list |
| generate\_price | Генерация стоимости. | str |

# Рекомендации пользователя

Для запуска программы убедитесь, что у вас установлен Python. Код можно запустить в среде разработки или через командную строку, используя консоль для настройки параметров и генерации данных. Также убедитесь, что все файлы программы находятся в одной директории для корректного выполнения. Запуск программы производится через файл main.py, который автоматически генерирует список пациентов поликлиники в файл dataset.csv. Важно периодически проверять корректность данных перед генерацией походов. Если вы хотите использовать собственные файлы с данными, убедитесь в корректности структуры и заголовков: names\_data, doctors\_data, cards\_data. Также настройте веса для банков и платежных систем согласно вашим требованиям, убедившись, что веса в сумме больше нуля.

# Рекомендации программиста

Поддерживайте актуальную версию Python для обеспечения работоспособности программы на современных системах. Уделяйте внимание четкому именованию переменных и функций. Регулярно проводите тестирование программы на различных входных данных, чтобы убедиться в её надежности и корректности.

# Исходный код программы

https://github.com/romplle/spbu-algorithms-and-data-structures

# Контрольный пример

1. Запуск программы

Для запуска программы используйте файл **main.py.** Программа будет отвечать за генерацию визитов к врачам на основе заданных данных о банках и платежных системах.

2. Ввод количества людей и весов платежных систем и банков

После запуска программы пользователю предложено ввести количество клиентов и **веса для банков** (Рис. 2) **и платежных систем** (Рис. 3). Веса определяют вероятность выбора того или иного банка или платежной системы.

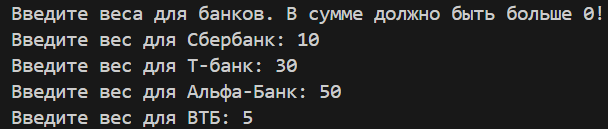


Рис 2. пример ввода весов платежных систем

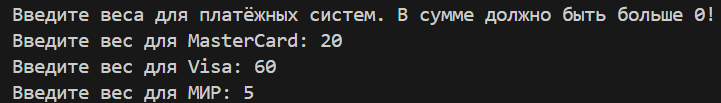


Рис 3. пример ввода весов банков

3. Ввод количества пациентов

После успешной генерации пациентов пользователю предлагается ввести количество строк в датасете (Рис. 4). Минимальное количество, которое можно сгенерировать, составляет 50.000(Рис. 5).



Рис 4. пример ввода количества билетов

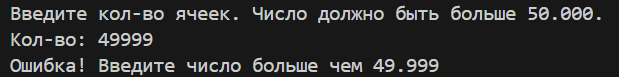


Рис 5. пример ввода количества билетов меньшего 50000

4. Генерация пациентов

После ввода количеств весов банков и платёжных систем программа приступает к генерации пациентов и их сохранении в dataset.csv.

# Вывод

В рамках данной работы были исследованы принципы генерации синтетических данных, применительно к моделированию посещений людей к врачу. Разработан алгоритм, который учитывает особенности врачей, симптомов и анализов. Было реализовано программное обеспечение для автоматической генерации датасета, включающего такие данные, как личные данные пассажиров, информация о симптомах, анализах и платежных системах. Программа позволяет настраивать параметры генерации банковских карт оплаты, обеспечивая соответствие заданным требованиям и реалистичность получаемого датасета.

# Источники

1. Datetime — The datetime module supplies classes for manipulating dates and times. // Python URL: [datetime — Basic date and time types — Python 3.12.6 documentation](https://docs.python.org/3/library/datetime.html) (дата обращения: 25.09.2024).
2. random — Generate pseudo-random numbers // Python URL: <https://docs.python.org/3/library/random.html> (дата обращения: 25.09.2024).
3. csv — CSV File Reading and Writing // Python URL: <https://docs.python.org/3/library/csv.html> (дата обращения: 25.09.2024).