**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**Факультет прикладной математики – процессов управления**

**отчет**

**по лабораторной работе**

**по дисциплине «Алгоритмы и структуры данных»**

**на тему «Генерация базы данных»**

**3 вариант**

|  |  |  |
| --- | --- | --- |
| Студент гр. 22.Б16 |  | Фатюшкин Д.В. |
| Преподаватель |  | Дик А.Г. |

**Санкт-Петербург**

**2023 г.**

**Оглавление**

[**1.** **Цель работы** 3](#_Toc152021108)

[**2.** **Задача** 3](#_Toc152021109)

[**3.** **Теоретическая часть** 3](#_Toc152021110)

[**4.** **Алгоритм метода** 4](#_Toc152021111)

[**5.** **Описание программы** 8](#_Toc152021112)

[**5.1.** **Описание функций** 8](#_Toc152021113)

[**6.** **Рекомендации пользователя** 10](#_Toc152021114)

[**7.** **Рекомендации программиста** 12](#_Toc152021115)

[**8.** **Контрольный пример** 13](#_Toc152021116)

[**9.** **Вывод** 14](#_Toc152021118)

# **Цель работы**

Приобретение и закрепление знаний в области генерации данных.

# **Задача**

Разработать алгоритм и программу, генерирующую набор данных, который обладает следующими наборами свойств:  
1. Пользователь - [example@mail.ru](mailto:example@mail.ru)  
2. IP адрес - 10.10.10.12  
3. Платформы - youtube.com  
4. Дата просмотра - 12.12.2022  
5. Кол-во рекламы - 2 раза  
6. Время просмотра рекламы - 1:03 минут  
7. Вид рекламы - кроссовки Nike

# **Теоретическая часть**

Генерация синтетических данных нужна для использования в самых разных областях. Несколько примеров приведено ниже:

1) Медицинская диагностика: Синтетические данные могут быть использованы для обучения моделей машинного обучения в задачах диагностики различных заболеваний. Например, модель может быть обучена на основе синтетически сгенерированных данных для распознавания определенных паттернов и симптомов заболеваний.

2) Автомобильная промышленность: Синтетические данные могут использоваться для обучения и тестирования моделей автопилотов в симуляторах. Это позволяет собирать большой объем данных и улучшать модели без необходимости физического наличия реальных автомобилей.

3) Финансовый анализ: Для обучения моделей, предсказывающих цены акций или валют, можно использовать синтетические данные, адекватно моделирующие рыночные тенденции и параметры.

4) Кибербезопасность: Синтетические данные могут быть созданы для генерации большого количества вирусных или вредоносных атак, которые могут быть использованы для обучения систем обнаружения и предотвращения таких угроз.Генерация синтетических данных является мощным инструментом, который может помочь в решении разнообразных задач в различных областях, устраняя ограничения, связанные с доступностью и конфиденциальностью реальных данных.

# **Алгоритм метода**

1. Считываются данные из полей ввода.
2. Данные проверяются на корректность. В случае ошибки интерфейс консольного типа попросит ввести данные заново, указав на некорректность.
3. Считываются данные из файлов «IPmask.txt», «mails.txt», «namesForMail.txt», «objadv.txt» и «platforms.txt» из папки «option\_files» в директории исполняемого файла «index.js».
4. Генерируется множество email-ов и ip-адресов, сохраняются в отдельные файлы «ip.txt» и «mails.txt» в папку «tmp\_files» в директории исполняемого файла «index.js»
5. Внутри файла «ip.txt», сгенерированного ранее, происходит перемешивание ip-адресов, так как они расположены в порядке возрастания.
6. Генерируется для каждого пользователя платформа, количество рекламы, дата и время просмотра рекламы, вид рекламы.
7. Все это записывается в файл с указанным пользователем именем в папку «tmp\_files»

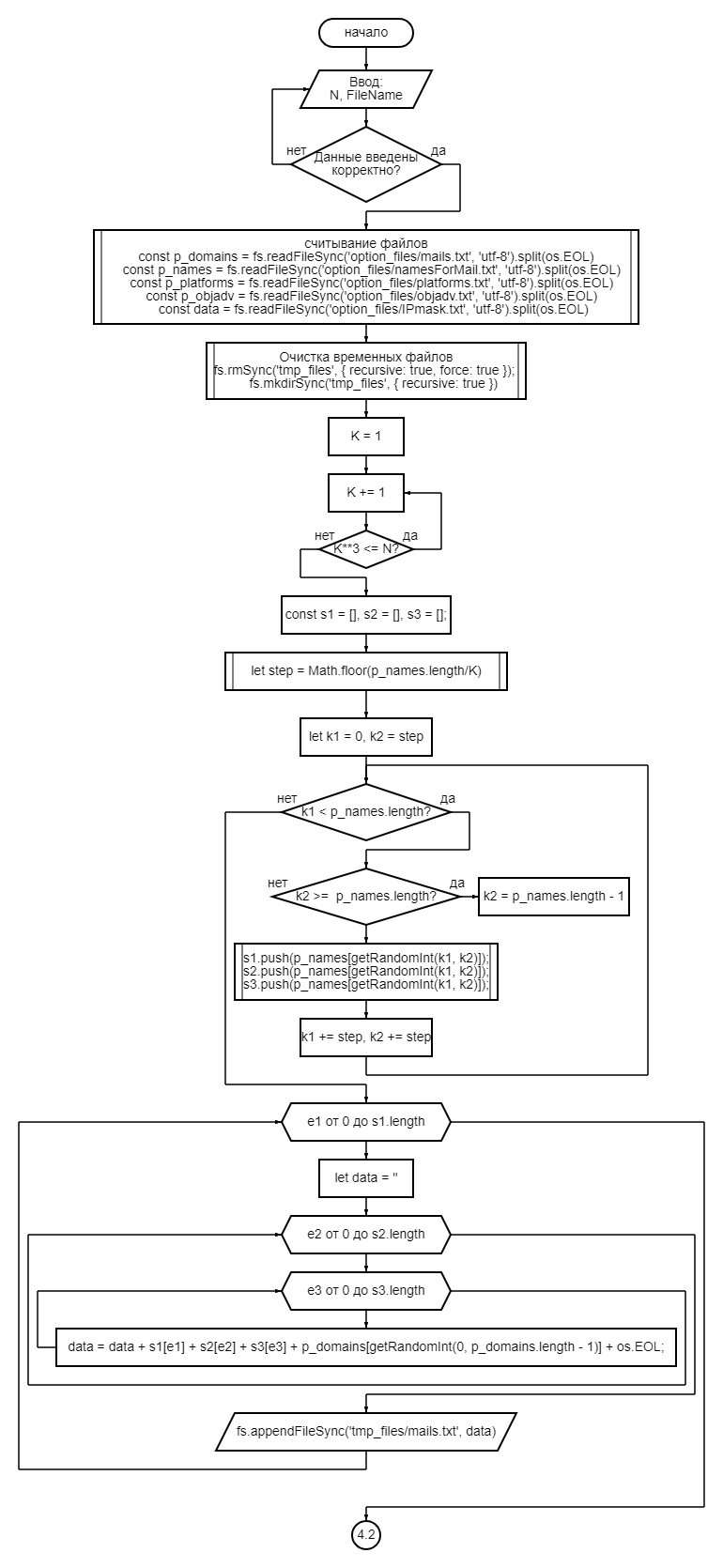


Рисунок 4.1. Блок-схема алгоритма генерации датасета.

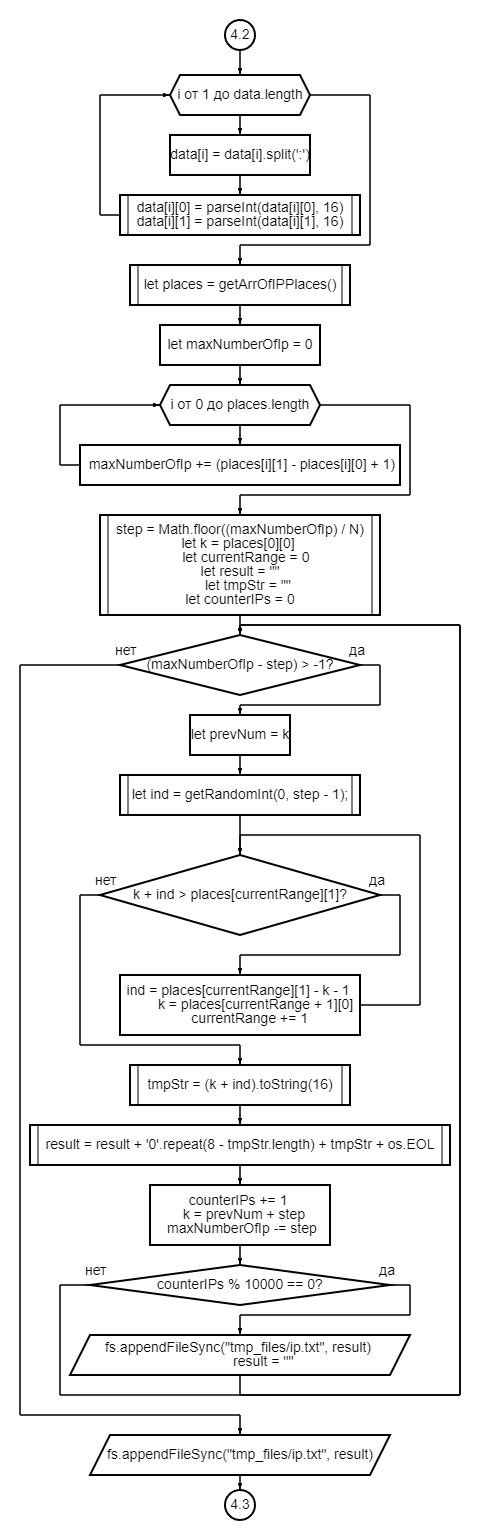


Рисунок 4.2. Блок-схема алгоритма генерации датасета.

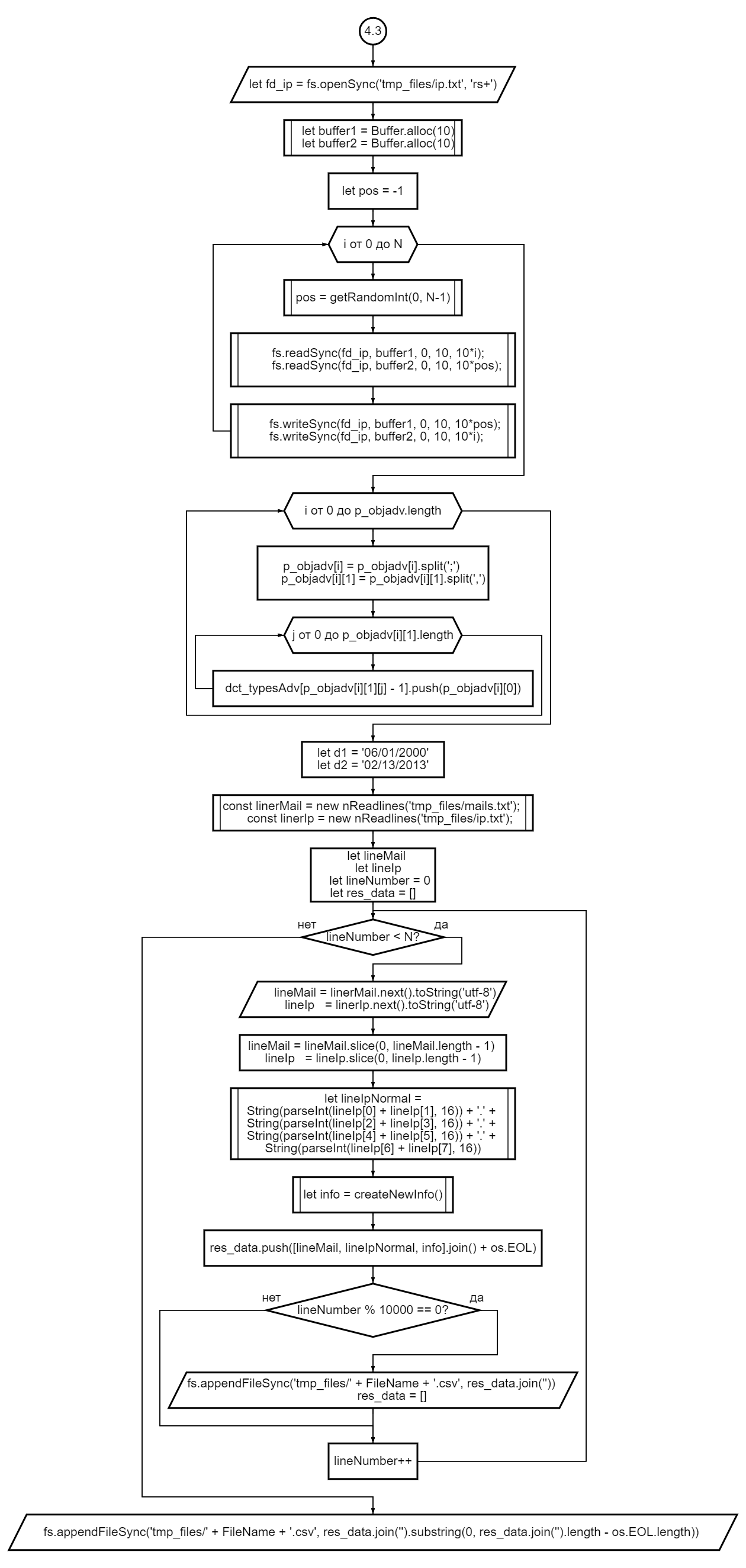


Рисунок 4.*3*. Блок-схема алгоритма генерации датасета.

# **Описание программы**

# **Описание функций**

В программе используются 10 функций: 4 связанных с интерфейсом и 6 – с генерацией данных. В таблице 5.1 представ­лено описание функций.

Таблица 5.1. Описание функций

|  |  |  |
| --- | --- | --- |
| Имя функции | Входные данные | Описание функции |
| makeData | Натуральное число и строка формата FileName | Генерация датасета с использованием введенных пользователем данных и данных из файлов папки «option\_files» |
| getRandomInt | 2 натуральных числа | Получает на вход 2 числа a и b, и возвращает рандомное целое число из диапазона [a;b] |
| clearLastLine | - | Очищает последнюю строку из потока вывода в консоли и возвращает курсор на очищенную позицию |
| getArrOfIPPlaces | - | Возвращает в специальном виде массив с промежутками ip-адресов, доступных для использования в генерации |
| randomDate | 2 значения строкового типа в формате “date” | Вохвращает рандомную дату между 2 указанными включительно |
| getRandomMonth | 2 значения строкового типа в формате “date” | Возвращает рандомный по счету месяц из указанного промежутка |
| createNewInfo | - | Возвращает 1 сгенерированную строку для датасета |
| UI | - | Вызывает графический интерфейс |
| isNumeric | Строка | Проверяет строку на соответствие числовому типу |
| ask | - | Запрашивает у пользователя ввод данных |

Описание основных переменных функции makeData представлено в таблице 5.2.

Таблица 5.2. Переменных функции generate\_data

|  |  |  |
| --- | --- | --- |
| Тип | Имя | Описание |
| Int | N | Количество строк в датасете |
| String | FileName | Файл, в который будет записываться датасет |
| String[] | p\_domains | Массив имен для доменов почт |
| String[] | p\_names | Массив для генерации названий почт (без учета домена) |
| String[] | p\_platforms | Массив платформ для генерации датасета |
| String[] | p\_objadv | Массив типов рекламы и ее возможной даты для генерации |
| String[  String[]  ] | data | Массив недоступных для генерации промежутков IP-адресов в формате масок |
| obj[obj[]] | dct\_typesAdv | Специальный массив объектов-массивов, содержащий в себе данные из «p\_objadv» в переработанном формате |
| int | maxNumberOfIp | Максимальное возможное число ip-адресов, которое можно использовать с учетом исключающих масок |
| date | d1 | Начало временного промежутка для генерации |
| date | d2 | Конец временного промежутка для генерации |
| String[] | res\_data | Чанк сгенерированных данных для записи, дописывающийся в файл и обнуляющийся после достижения конкретного размера |
| int | lineNumber | Счетчик строк, от которого зависит, когда обнулится чанк res\_data и информация из него дозапишется в датасет, после чего чанк снова будет заполняться. |

# 

# **Рекомендации пользователя**

Перед началом работы убедитесь, что в папке «option\_files» в каталоге с программой имеются файлы со списком платформ, доменов почтовых сервисов, масками адресов, типами рекламы и именами для генерации почт. По умолчанию – «platforms.txt», «mails.txt», «IPmask.txt», «objadv.txt» и «namesForMail.txt» соответственно. Если таковые отсутствуют, то создайте их самостоятельно. Примеры заполнения прилагаются:

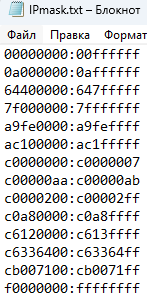


Рисунок 6.1 Содержимое файла «IPmask.txt»

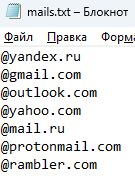


Рисунок 6.2 Содержимое файла «mails.txt»

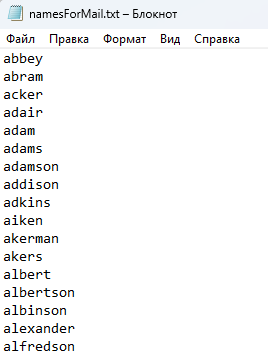


Рисунок 6.3 Содержимое файла «namesForMail.txt»

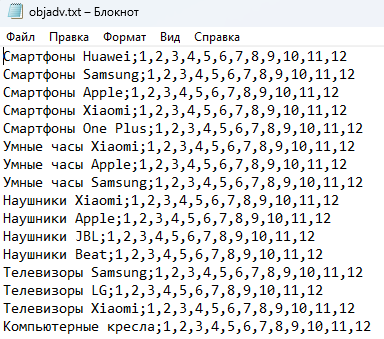


Рисунок 6.4 Содержимое файла «objadv.txt». Цифры через двуеточие и запятые – месяцы, подходящие для рекламы данного типа

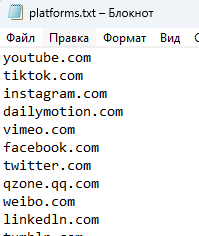


Рисунок 6.5 Содержимое файла «objadv.txt»

Запуск программы осуществляется с помощью вызова команды «node index» (если файл с исходным кодом называется «index.js») с помощью консоли или другого средства, например, ide, в каталоге с исполняемым файлом.

После запуска программы необходимо ввести количество строк для генерации и название файла для вывода датасета, нажимая «Enter» для подтверждения ввода. После этого датасет будет сгенерирован и сохранен в папке «tmp\_files» директории исполняемого файла с указанным именем. Программа завершит работу самостоятельно после окончания генерации.

Для быстрой работы программы необходимо иметь высококачественный и производительный твердотельный накопитель (SSD).

# **Рекомендации программиста**

Для запуска программы необходима 64-битная операционная система Windows и Node.js версии 18.\* и NPM (на других версиях так же сохраняется работоспособность, но возможны ошибки, тестирование других версий не производилось). Возможны ошибки связанные с библиотеками NPM. Можно попробовать удалить полностью папку node\_modules и в командной строке (внутри директории исходного файла index.js, например) прописать:

npm init –y

npm i

Это позволит npm скачать чистые пакеты из репозиториев.

Необходимое количество оперативной памяти: не более 100 мб.

# **Контрольный пример**

В данном разделе представлен контрольный пример, демонстрирующий способность программы генерировать данные, основываясь на ранее заданных параметрах.



Рисунок 8.1. Окно программы после запуска. Ожидание ввода. Необходимо ввести число (количество строк в датасете) и нажать «Enter». Если ничего не ввести, то будет выбрано дефолтное количество строк = 100000



Рисунок 8.2. Ожидание ввода. Необходимо название файла для вывода датасета и нажать «Enter». Если ничего не ввести, то будет выбрано дефолтное имя data.csv

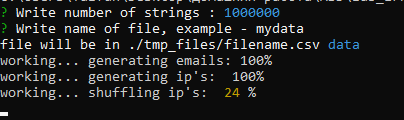


Рисунок 8.3. Выводятся названия стадий генерации и процент их завершенности

# 

Рисунок 8.4. Программа завершила работу и вывела оповещение об этом

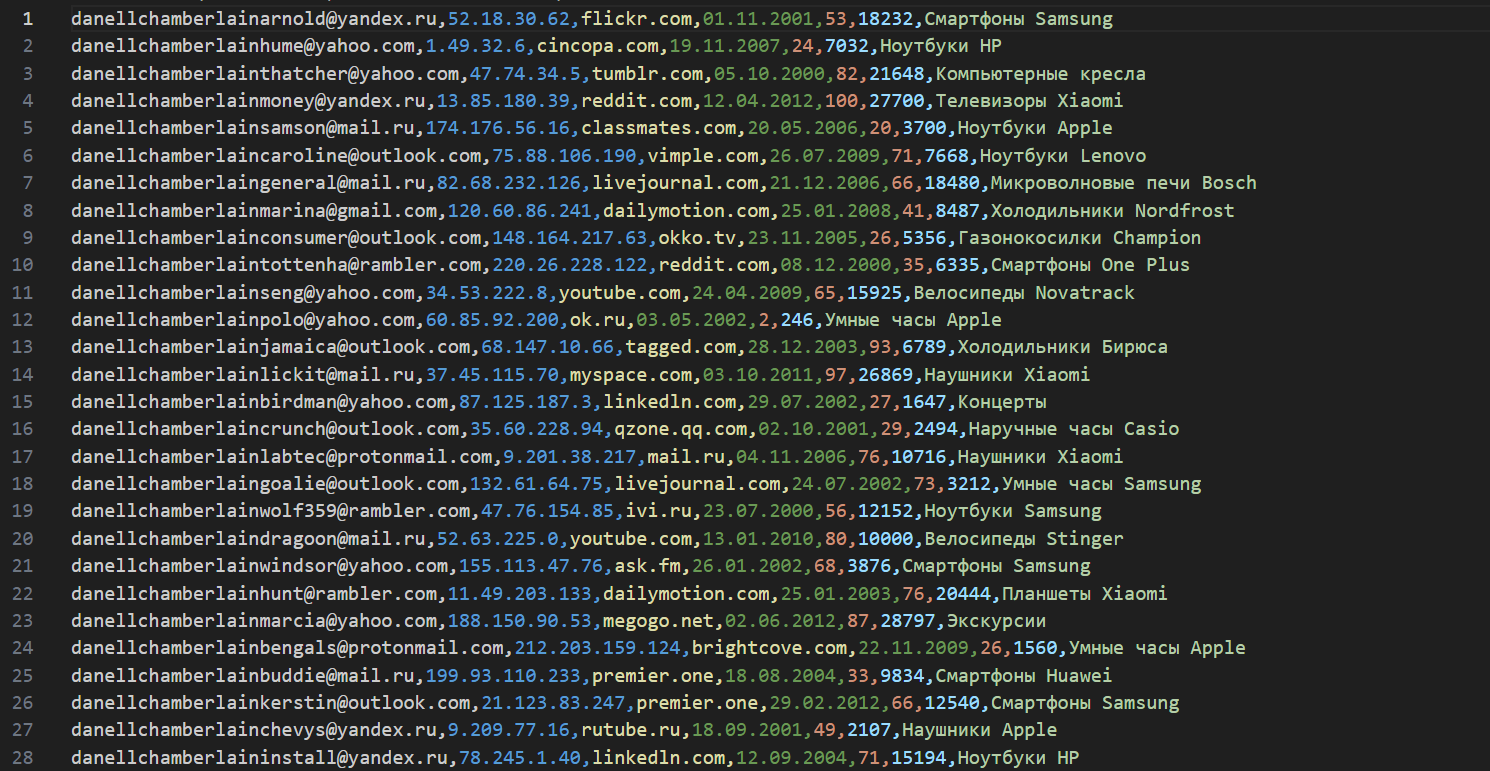


Рисунок 8.5. Результат работы программы

# **Вывод**

В ходе выполнения работы по генерации данных были достигнуты следующие результаты и выводы:

В процессе исследования и освоения различных методов и инструментов для генерации данных, были изучены способы генерации синтетических данных, создания случайных данных, а также использование специализированных библиотек и инструментов.

Выполнение работы по генерации данных способствовало улучшению навыков анализа данных, так как генерация данных требует понимания структуры и особенностей данных, которые необходимо сгенерировать.

Полученный при проведении работы опыт в генерации данных является ценным и может быть применен при решении реальных задач в будущем.