

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
Антонян А.А.**

**Преподаватель
Дик А.Г.**

Санкт-Петербург

2024 г.

Оглавление

1. Цель работы.....	3
2. Описание задачи (формализация задачи).....	3
3. Теоретическая часть	4
4. Основные шаги программы	5
5. Блок схема программы	6
6. Описание программы	8
7. Рекомендации пользователя	11
8. Рекомендации программиста.....	12
9. Исходный код программы.....	12
10. Контрольный пример.....	12
11. Вывод	15
12. Источники.....	15

Цель работы

Целью лабораторной работы является разработка системы генерации датасета покупок в магазинах с учетом определенных требований и условий. Датасет должен включать данные покупки, такие как категория товара, название магазина, название бренда, стоимость, количество товаров, координаты магазина и время покупки, а также платежные данные покупателя.

Описание задачи (формализация задачи)

Задача состоит в создании датасета, содержащего данные о покупке со следующими требованиями:

1. **Название магазина:** Не менее 30 уникальных магазинов.
2. **Координаты и время:** Координаты должны соответствовать реальному местоположению магазина, а время покупки не должно выходить за пределы времени работы магазина.
3. **Категорий:** Категорий должен соответствовать тематике магазина. Не менее 50 категорий.
4. **Бренд:** Не менее 500 уникальных брендов.
5. **Номер карты:** Генерация карт с возможностью многократного использования с повторением не больше пяти раз и возможностью настраивать вероятность к какому банку и платежной системе принадлежит карта.
6. **Количество товаров:** Минимум 5 товаров должно быть в каждой покупке.
7. **Стоимость:** Должна соответствовать средней стоимости товара.
8. **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.
9. **Количество строк датасета:** Не меньше 50000.

Теоретическая часть

Для создания датасета использованы несколько программных модулей:

1. `generate.py`: Генерация времени покупки, случайная генерация тематики магазина, генерация названия магазина, генерация категория, генерация бренда, генерация количества товаров, генерация стоимости товара, функция взвешенного выбора для настройки вероятностей генерации содержимого датасета.
2. `data.py`: Модуль, содержащий словари, используемые для случайной генерации данных.
3. `buy.py`: Модуль содержащий классы для связывания сгенерированной информации.
4. `main.py`: Основной скрипт для генерации датасета, который объединяет данные из вышеуказанных модулей и записывает их в XML файл.

Ограничения:

- Количество строк в датасете ограничивается вводом пользователя, но минимальное количество сгенерированных строк будет 50000.
- Количество магазинов – минимум 30.
- Координаты должны соответствовать реальному местоположению магазина, а время покупки не должно выходить за пределы времени работы магазина.
- Количество брендов – минимум 500.
- Количество категорий – минимум 50.
- Количество товаров – минимум 5.
- Веса банков и платежных систем определяются пользователем и должны в сумме давать число больше нуля.
- Логика выбора карт оплаты с ограничением на 5 повторов.

Основные шаги программы

- 1) Запуск программы (main.py):
- 2) Пользователь вводит веса банков и платежных систем.
- 3) Запускается функция генерации датасета, внутри которой запускается цикл по количеству строк датасета. Внутри цикла создается объект класса `Vu`. Генерируются поля объекта класса `Vu` и ими заполняется датасет.
 - а) Поля объекта класса `Vu` взаимосвязаны (одни данные генерируются на основе других), таким образом товары соответствуют категории магазина, а бренды соответствуют категории товара.
- 4) Данные собираются и записываются в файл `purchases.xml`.

Блок схема программы

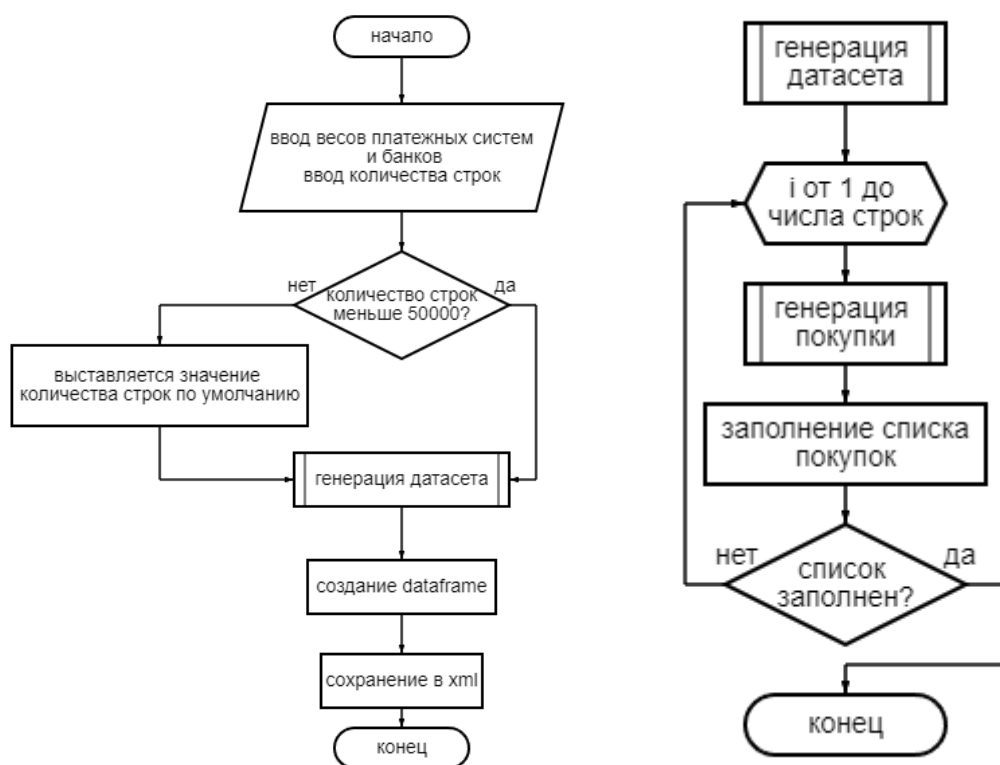


Рис 1. Блок-схема main.py

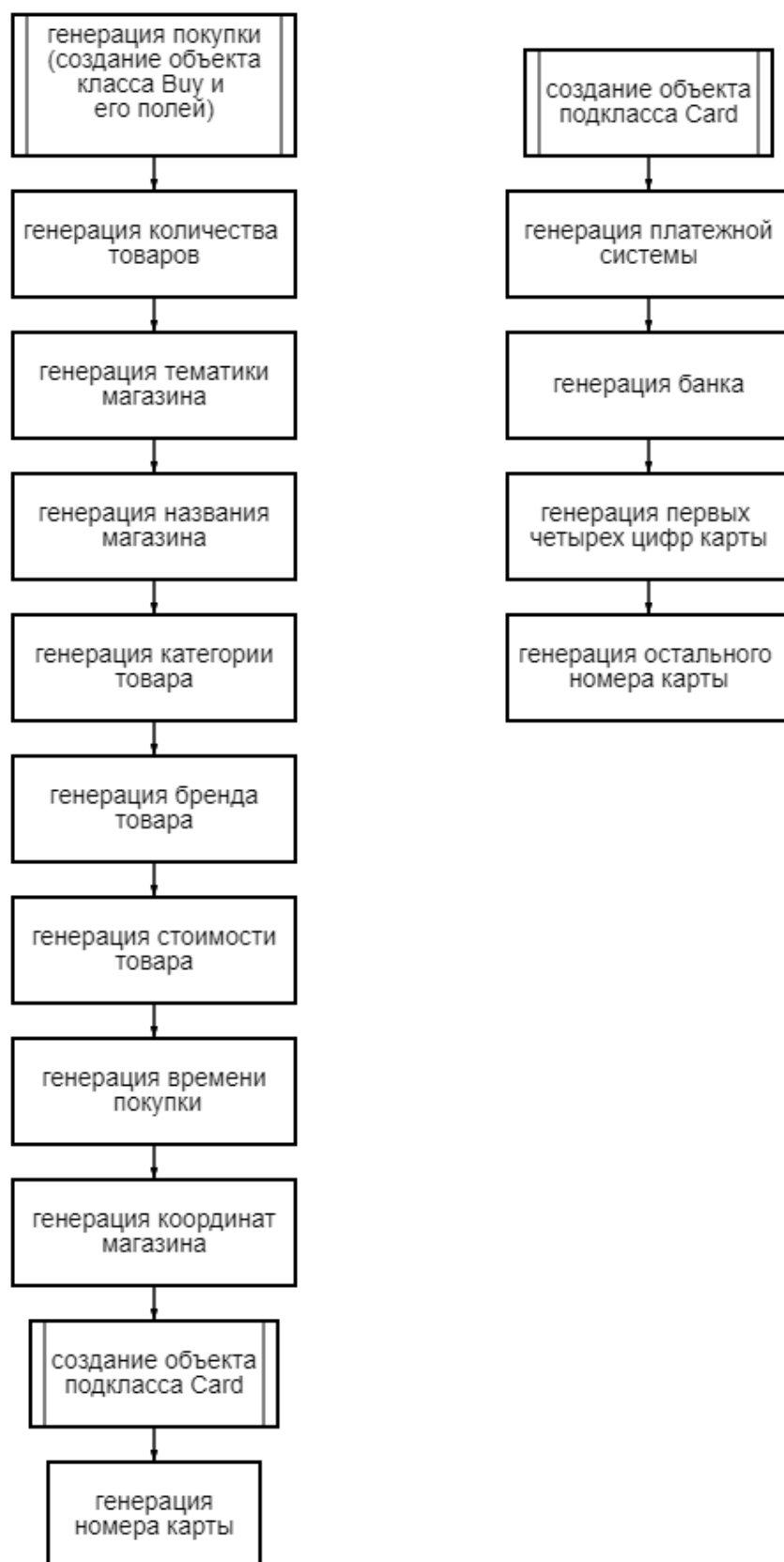


Рис 2. Блок-схемы подпрограмм

Описание программы

Программная реализация написана на языке Python 3.12.2 с использованием следующих библиотек: pandas [\[1\]](#), random [\[2\]](#), и datetime [\[3\]](#). Программа организована через модульную структуру с акцентом на генерацию данных о покупках. В процессе разработки программы использовались 8 функций, каждая из которых имеет чётко определённое назначение и 4 основных модуля:

Таблица 1. generate.py

Функция	Описание	Возвращаемое значение
generate_random_datetime	Генерация случайной даты и времени покупки.	str
generate_department	Случайная генерация отдела магазина.	str
generate_shop	Генерация случайного магазина на основании его тематики.	str
generate_categories	Генерация случайной категории товара.	str
generate_brands	Генерация случайного бренда на основании категории его товара и тематики магазина.	str
generate_quantity	Случайная генерация количества товаров	int
generate_cost	Случайная генерация стоимости товара на основании его ценовой категории.	int

weighted_choice	Функция взвешенного выбора.	Any
-----------------	-----------------------------	-----

Таблица 2. Класс Buy

Поле	Описание	Тип
quantity	Случайно генерируется количество товаров	int
department	Случайно сгенерированная тематика магазина	str
shop	Случайно сгенерированное название магазина в зависимости от его тематики	str
category	Случайно сгенерированная категория товара в зависимости от тематики магазина	str
brand	Случайно сгенерированный бренд в зависимости от категории товара	str
cost	Случайно сгенерированная стоимость в зависимости от ценовой категории товара	int
time	Случайно сгенерированное время покупки	str
coordinates	Случайно сгенерированные координаты, соответствующие определенному магазину	tuple
card_number	Случайно сгенерированный номер карты, в зависимости от платежной системы и банка (которые генерируются в подклассе Card)	str

Таблица 3. Подкласс Card

Метод/Поле	Описание	Возвращаемое значение/Тип
generate_bank_card_number (метод)	Генерация номера карты.	str
payment_system	Случайно сгенерированная, с учетом весов, платежная система	str
bank	Случайно сгенерированный, с учетом весов, банк	str
pre	Первые четыре цифры карты, выбранные в зависимости от банка и платежной системы	str
bank_card_number	Номер карты, сгенерированный с учетом первых четырех цифр	str

Таблица 4. main.py

Функция	Описание	Возвращаемое значение
generate_data	Генерация датасета.	list

Таблица 5. data.py

Структура данных	Описание	Тип
shops	Двухуровневый словарь, содержащий тематики магазинов, которым соответствуют определенные магазины с их координатами.	dict
low_price_category	Двухуровневый словарь, содержащий товары низкой ценовой категории, которым соответствуют возможные бренды.	dict
medium_price_category	Двухуровневый словарь, содержащий товары средней ценовой категории, которым соответствуют возможные бренды.	dict
high_price_category	Двухуровневый словарь, содержащий товары высокой ценовой категории, которым соответствуют возможные бренды.	dict
pm_sys	Двухуровневый словарь, ключи первого уровня – платежные системы, им соответствуют банки, которым в свою очередь соответствуют первые четыре цифры номера карты в определенной платежной системе.	dict

Рекомендации пользователя

Для запуска программы убедитесь, что у вас установлен Python и необходимые библиотеки, такие как pandas [\[1\]](#). Код можно запустить в среде разработки или через командную строку, используя консоль для настройки параметров и генерации данных. Также убедитесь, что все модули программы находятся в одной директории для корректного выполнения. Запуск программы производится через файл main.py, который автоматически генерирует датасет в файл purchases.xml. Также настройте веса для платежных

систем и банков согласно вашим требованиям, убедившись, что веса больше нуля.

Рекомендации программиста

Поддерживайте актуальные версии используемых библиотек и Python для обеспечения работоспособности программы на современных системах. Следуйте передовым практикам разработки (best practices), 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/ArseniiAntonian/spbu-algorithms-and-data-structures>

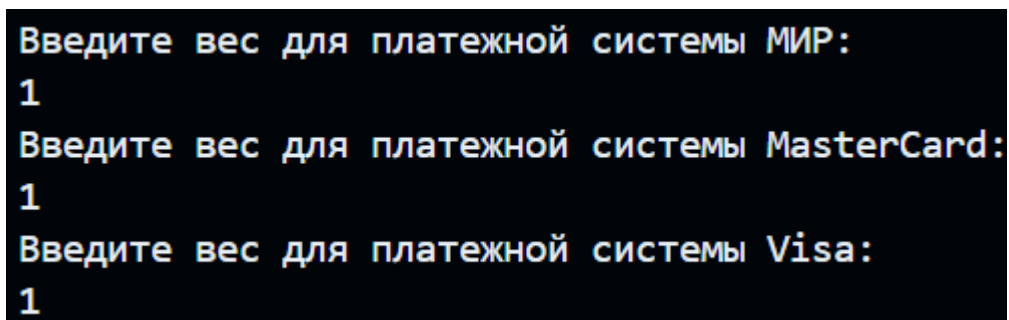
Контрольный пример

1. Запуск программы

Для запуска программы используйте файл `main.py`. Программа будет отвечать за генерацию датасета с покупками на основе заданных данных о количестве строк, платежных системах и банках.

2. Ввод весов платежных систем и банков

После запуска программы пользователю предложено ввести веса для платежных систем (Рис. 3) и веса для банков (Рис. 4). Веса определяют вероятность выбора той или иной платежной системы или банка при генерации билета.



```
Введите вес для платежной системы МИР:
1
Введите вес для платежной системы MasterCard:
1
Введите вес для платежной системы Visa:
1
```

Рис 3. Пример ввода весов платежных систем

```
Введите вес для банка Сбер:
1
Введите вес для банка Т-банк:
1
Введите вес для банка ВТБ:
1
Введите вес для банка Альфа банк:
1
```

Рис 4. Пример ввода весов банков

3. Ввод количества строк в датасете

Пользователю предлагается ввести количество строк для генерации (Рис. 5). Минимальное количество билетов, которое можно сгенерировать, составляет **50 000** (Рис. 6).

```
Введите количество строк (минимум 50000) :100000
```

Рис 5. Пример ввода количества строк

```
Введите количество строк (минимум 50000) :1
Количество строк не может быть менее 50000. Установлено значение 50000
```

Рис 6. Пример ввода количества строк меньше 50000

4. Генерация датасета

После ввода количества строк программа приступает к генерации датасета, используя введенные параметры, затем сгенерированный датасет сохраняется в purchases.xml.

```

assignment1 > purchases.xml
1  <?xml version='1.0' encoding='utf-8'?>
2  <purchases>
3    <purchase>
4      <shop>Животный мир</shop>
5      <coordinates>(59.8962, 30.3621)</coordinates>
6      <time>2021-07-28T15:41:50+3:00</time>
7      <category>корм для кошек</category>
8      <brand>IAMS</brand>
9      <card_number>2201425639391281</card_number>
10     <quantity>74</quantity>
11     <cost>72742</cost>
12   </purchase>
13   <purchase>
14     <shop>Магнит</shop>
15     <coordinates>(60.016238, 30.311033)</coordinates>
16     <time>2023-04-30T15:45:32+3:00</time>
17     <category>слабоалкогольные напитки</category>
18     <brand>Desperados</brand>
19     <card_number>4377964285348183</card_number>
20     <quantity>69</quantity>
21     <cost>20424</cost>
22   </purchase>
23   <purchase>
24     <shop>Максидом</shop>
25     <coordinates>(60.002133, 30.384205)</coordinates>
26     <time>2022-09-10T14:52:47+3:00</time>
27     <category>мышь</category>
28     <brand>Cougar</brand>
29     <card_number>2201385806073784</card_number>
30     <quantity>93</quantity>
31     <cost>84165</cost>
32   </purchase>

```

Рис 7. пример датасета

Вывод

В рамках данной работы были исследованы принципы генерации синтетических данных, применительно к моделированию покупок в магазинах. Разработан алгоритм, который учитывает особенности категорий товаров, тематик магазина, ценовой категории товаров, брендов. Было реализовано программное обеспечение для автоматической генерации датасета, включающего такие данные, категории товара, бренд товара, название магазина, время покупки и координаты магазина, информация о платежных системах и банках. Программа позволяет настраивать параметры генерации банковских карт оплаты, обеспечивая соответствие заданным требованиям и реалистичность получаемого датасета.

Источники

1. Pandas' documentation // Pandas URL: <https://pandas.pydata.org/docs/> (дата обращения: 19.09.2024).
2. random — Generate pseudo-random numbers // Python URL: <https://docs.python.org/3/library/random.html> (дата обращения: 19.09.2024).
3. Datetime's documentation // Python URL: <https://docs.python.org/3/library/datetime.html> (дата обращения: 19.09.2024).