# Документация к файлу `password\_generator\_v2\_comm.py`

## Общее описание

Файл `password\_generator\_v2\_comm.py` содержит код графического приложения для генерации паролей на основе заданных параметров. Приложение предоставляет пользователю возможность настраивать длину, сложность и другие характеристики паролей, а также сохранять, копировать или печатать результаты.

---

## Используемые модули

- \*\*`tkinter`\*\* — для создания графического интерфейса.

- \*\*`random`\*\* — для генерации случайных паролей.

- \*\*`string`\*\* — для работы с символами.

- \*\*`tempfile`\*\* — для создания временных файлов.

- \*\*`os`\*\* — для работы с операционной системой (печать).

- \*\*`csv`\*\* — для сохранения паролей в формате CSV.

---

## Основные функции и классы

### Класс `PasswordGeneratorApp`

Главный класс приложения, управляющий интерфейсом и логикой генерации паролей.

#### Методы:

1. \*\*`\_\_init\_\_(self, root)`\*\*

Инициализирует приложение, создаёт виджеты и применяет тему по умолчанию.

2. \*\*`show\_help\_message(self)`\*\*

Отображает вспомогательное окно с информацией о функционале приложения.

3. \*\*`apply\_theme(self, theme)`\*\*

Применяет выбранную тему (светлую или тёмную) ко всем элементам интерфейса.

4. \*\*`toggle\_theme(self)`\*\*

Переключает между светлой и тёмной темами.

5. \*\*`generate\_password(self, length, level)`\*\*

Генерирует пароль на основе заданных длины и уровня сложности.

6. \*\*`auto\_level(self, length)`\*\*

Автоматически определяет уровень сложности пароля в зависимости от его длины.

7. \*\*`on\_generate(self)`\*\*

Обрабатывает нажатие кнопки "Сгенерировать", создаёт пароли и выводит их в текстовое поле.

8. \*\*`on\_copy(self)`\*\*

Копирует сгенерированные пароли в буфер обмена.

9. \*\*`on\_save(self)`\*\*

Сохраняет пароли в файл (форматы: TXT или CSV).

10. \*\*`on\_print(self)`\*\*

Отправляет пароли на печать через временный файл.

11. \*\*`reset\_settings(self)`\*\*

Сбрасывает все настройки к значениям по умолчанию.

12. \*\*`create\_widgets(self)`\*\*

Создаёт и размещает все элементы интерфейса.

---

## Переменные класса `PasswordGeneratorApp`

- \*\*`THEMES`\*\*

Словарь с цветовыми схемами для светлой и тёмной тем.

- \*\*`current\_theme`\*\*

Текущая тема интерфейса (по умолчанию `"dark"`).

- \*\*`all\_widgets`\*\*

Список всех виджетов для удобного управления темой.

- \*\*`difficulty\_var`\*\*, \*\*`auto\_level\_var`\*\*, \*\*`unique\_var`\*\*, \*\*`exclude\_similar\_var`\*\*, \*\*`excluded\_chars\_var`\*\*

Переменные для хранения состояния чекбоксов и выпадающих списков.

- \*\*`length\_entry`\*\*, \*\*`count\_entry`\*\*, \*\*`result\_text`\*\*

Виджеты для ввода длины, количества паролей и вывода результатов.

---

## Горячие клавиши

Приложение поддерживает следующие комбинации клавиш:

- \*\*Ctrl+G\*\* — сгенерировать пароли.

- \*\*Ctrl+C\*\* — скопировать пароли в буфер обмена.

- \*\*Ctrl+S\*\* — сохранить пароли в файл.

- \*\*Ctrl+P\*\* — отправить пароли на печать.

- \*\*Ctrl+R\*\* — сбросить настройки.

---

## Пример использования

1. Запустите файл `password\_generator\_v2\_comm.py`.

2. Задайте параметры паролей (длину, сложность, количество).

3. Нажмите "Сгенерировать" или используйте горячие клавиши.

4. Сохраните, скопируйте или напечатайте результаты.

---

## Примечания

- Приложение поддерживает русский язык интерфейса.

- Для корректной работы на Windows требуется настроенный принтер (для функции печати).