**Что делает каждый блок**

| **Путь** | **Ответственность** |
| --- | --- |
| **app.py** | запускает клиент, грузит конфиг, переходит между режимами *dialogs/chat/help* |
| **api/** | изолирует Telethon, чтобы в остальных модулях не светились TL‑типы (будет проще сменить библиотеку, если вдруг) |
| **input/** | низкоуровневое чтение клавиш через msvcrt и прокладка к ­async‑слою (aioconsole) для ввода строк — приём, который советуют для асинхронных CLI под Windows [Telethon Documentation](https://docs.telethon.dev/?utm_source=chatgpt.com) |
| **views/** | рендер с помощью Rich‑таблиц и Live‑режима; библиотека официально поддерживает раскладку экранов через Layout [Rich Documentation](https://rich.readthedocs.io/en/stable/layout.html?utm_source=chatgpt.com)[GitHub](https://github.com/Textualize/rich?utm_source=chatgpt.com) |
| **state.py** | описывает DialogState, ChatState, глобальные настройки (page\_size, theme) |
| **plugins/** | подгружается динамически (см. [rich‑cli](https://github.com/Textualize/rich-cli) и советы сообщества о плагинах) [GitHub](https://github.com/Textualize/rich-cli?utm_source=chatgpt.com)[Reddit](https://www.reddit.com/r/learnpython/comments/14tbqqm/ship_cli_programs_with_utility_package_best/?utm_source=chatgpt.com) |

**Предлагаю набросок команд / горячих клавиш**

*(все в «один символ» — чтобы печатать вслепую; можно дублировать длинным :‑командным синтаксисом для скриптового режима)*

| **Контекст** | | **Клавиша** | **Что делает** | | **Комментарий** | |
| --- | --- | --- | --- | --- | --- | --- |
| **Список диалогов** | | номер | Открыть выбранный чат | | 1‑я строка UX — быстро **1 ↵** и ты внутри чата | |
|  | | n / p | Следующая / предыдущая страница | | шаг по page\_size (по умолч. 20) | |
|  | | u | Фильтр «только непрочитанные» | | тумблер; повторно — все чаты | |
|  | | r | Mark all as read | | сразу 0 уведомлений (ℹ️ спросить подтверждение) | |
|  | | f | Быстрый поиск чатов по имени | | ввод строки → автоподсветка совпадений | |
|  | | / | Вызвать «длинную» команду | | например: /export @devchat 2024‑01‑01:2024‑03‑01 | |
|  | | h | Справка | | вывод всей таблицы клавиш | |
|  | | q | Выход | | сохранение state и graceful disconnect | |
| **Внутри чата** | **Клавиша** | | | **Что делает** | | **Комментарий** |
| ↵ (Enter) | Пометить показанное как прочитанное | | | классика «пролистал — прочитал» | |  |
| ↑ / ↓ | Пролистать по 10 сообщений | | | держим chat‑offset | |  |
| u | Загрузить «выше» ещё N сообщ. | | | lazy‑scroll из API | |  |
| b | Назад к списку диалогов | | | сохраняем позицию | |  |
| m | Ответить сообщением | | | одно‑строчный редактор, Esc — отмена | |  |
| d | Скачать вложения поста под курсором | | | если несколько — выводим выбор | |  |
| v | Открыть вложение (фото/видео/док) | | | через xdg-open / start | |  |
| p | Проиграть voice / видео | | | зовём mpv асинхронно | |  |
| e | Экспорт истории | | | диапазон дат → .md/.html | |  |
| c | Копировать текст поста в буфер | | | если OS поддерживает xclip / clip | |  |
| s | Поиск по истории чата | | | серверный search\_messages + подсветка | |  |
| h | Справка по чату | | | другая подсказка, т.к. команды отличаются | |  |

**Глобальные клавиши (работают везде)**

Ctrl‑R - Переподключиться к серверу (re‑auth)

Ctrl‑C - Корректное завершение (trap SIGINT)

Ctrl‑L - Очистить экран и перерисовать UI

**UI‑логика в чате**

| **Действие** | **Ключ** | **Что происходит** |
| --- | --- | --- |
| Переместить курсор по сообщениям | ↑ / ↓ | Выделяемое сообщение подсвечивается (rich ANSI‑цвет). |
| Ответить на выделенное | r | CLI показывает строку reply> , ждёт ввод текста и отправляет как **reply‑to**. |
| Отмена | Esc | Закрываем редактор, курсор остаётся на том же сообщении. |

*Пользователь видит визуальное подтверждение — строка «Отправлено в ответ на #123» под своим сообщением.*

**Сценарий работы консольного клиента Telegram на Python**

**Цель** — быстрая проверка «что нового» без перехода в GUI‑клиент и с минимальным отвлечением. Пользователь запускает бинарь в терминале, проходит авторизацию один раз, после чего взаимодействует с Telegram через компактное текстовое меню.

**1. Запуск и авторизация**

1. При первом запуске приложение запрашивает **api\_id / api\_hash** и номер телефона.
2. Получает код от Telegram, сохраняет сессионный файл (.session) локально.
3. Все последующие запуски используют сохранённую сессию (если файл доступен — авторизация прозрачна).

**2. Основное состояние — «Список диалогов»**

| **№** | **Тип** | **Название** | **Непрочитано** | **Последнее сообщение** |
| --- | --- | --- | --- | --- |
| 1 | 👤 | Ivan Ivanov | 3 | «Ок, договорились…» |
| 2 | 💬 | DevChat | 12 | «Pull‑request принят» |
| … | … | … | … | … |

*Отображается первая N‑ка диалогов по убыванию последней активности.*

После таблицы выводится «мини‑меню»:

[номер] – открыть чат n – следующая страница p – предыдущая

r – пометить все как прочитанные s – поиск q – выход

**Что происходит под капотом**

* Один HTTP‑запрос GetDialogs (или IterDialogs у Telethon) даёт нам и названия, и счётчик непрочитанного.
* Пагинация хранится в объекте состояния (offset/limit).

**3. Состояние «Просмотр чата»**

При выборе номера:

=== DevChat === (12 новых)

20:14 Alex ➜ Я запушил фикс

20:13 Bob ➜ Глянь MR #42

...

[Enter] – отметить просмотренное b – назад r – ответ u – вверх по истории

* Сообщения показываются в хронологическом порядке сверху вниз (можно инвертировать).
* Нажатие **Enter** помечает всё показанное как прочитанное через ReadHistoryRequest.
* Команда **u** загружает ещё N сообщений (листать выше).

**4. Дополнительные функции, которые стоит заложить**

| **Функция** | **Польза / детали реализации** |
| --- | --- |
| **Фильтр «только с непрочитанным»** | Быстро выводить список диалогов, где unread\_count > 0. |
| **Поиск (s)** | Вводим ключевые слова → search\_messages на сервере, затем постраничный вывод. |
| **Экспорт диалога** | Клавиша **e** в чате → спрашиваем диапазон дат и сохраняем в markdown/HTML. |
| **Быстрый ответ (r)** | Открывает однострочный ввод; отправляем SendMessageRequest. |
| **Скрытое обновление** | Фоновая корутина каждые X секунд обновляет счётчики; если новые сообщения появились — метка «\*». |
| **Темизация / цвет** | ANSI‑цвета: разный оттенок для своих vs чужих сообщений; выделение упоминаний. |
| **Сигналы OS** | Ctrl‑C ловим, делаем graceful shutdown (закрываем соединение). |
| **Конфиг‑файл** | ~/.tg\_cli.yaml: размер страницы, клавиши, авто‑пометка как прочитанные, прокси и т. д. |
| **Плагин‑хук** | Простая точка расширения (python‑модуль в plugins/), чтобы пользователь мог добавить собственную команду, например «скачай все фото из чата». |