

Комикс о том, как гуманитарии учатся программировать

ГУМАНИТАРИЙ

.exe

Из гуманитария в digital.
Миссия: не сойти с ума! 🐍

МОЗГ:
гуманитарный ❤️

```
def solve(problem):  
    if not understand:  
        ask(ishka)  
        try_again()  
    return success
```

Всё просто.
Главное - отступы!

ПЛАН:

- ☑️ Выучить Python
- ☑️ не сойти с ума
- ☑️ создать что-нибудь крутое ☆



ERROR:
humanitarian
обнаружен



SyntaxError:
invalid syntax

ЦЕЛЬ:
не сойти
с ума ☆

ОРУЖИЕ:
любопытство
и упорство
❤️ ❤️

НЕ
ПАНИКОВАТЬ!
❤️

humanitarian.exe -- загрузка...



%%

❤️ юмор • самоирония • упорство • код • прогресс ❤️

Элина Третьякова
Гуманитарий.exe

«Автор»

2026

Третьякова Э.

Гуманитарий.exe / Э. Третьякова — «Автор», 2026

«Гуманитарий.exe» — ироничный комикс о реальном пути человека из мира текстов и смыслов в мир Python, нейросетей и разработки. История о том, что войти в IT можно, даже если сначала ты не понимаешь вообще ничего. Главное — вовремя нажать «Запуск».

© Третьякова Э., 2026

© Автор, 2026

Элина Третьякова

Гуманитарий.exe

Глава 1

Глава 1. Всё началось довольно невинно

Всё началось довольно невинно.

Пару лет назад, когда нейросети уже перестали казаться чем-то совсем фантастическим, но ещё не успели превратиться для меня в привычный инструмент, я решила с ними познакомиться. До этого я, конечно, слышала про искусственный интеллект, читала новости, видела сгенерированные картинки и примерно представляла, на что способны современные модели. Но всё это существовало где-то рядом и никак особенно не вмешивалось в мою жизнь. Преимущественно это были генераторы контента — изображений, гифок, инструменты для обработки фотографий.

А потом я открыла Claude.

Не помню уже, что именно спросила в первый раз. Наверняка ничего судьбоносного. Скорее всего, это был обычный вопрос, который раньше я без раздумий отправила бы в поисковик. Но вместо списка ссылок получила ответ. Задала следующий вопрос — получила следующий. Уточнила, возразила, попросила объяснить иначе — и постепенно обнаружила, что уже не проверяю возможности новой игрушки, а разговариваю.

Это было странное ощущение. Я прекрасно понимала, что по ту сторону экрана нет человека, и всё же человеческий мозг оказался удивительно падок на собеседника, который внимательно следует за ходом разговора, не перебивает, не раздражается, не устает от уточнений и способен помнить, о чём шла речь несколькими сообщениями раньше. Через какое-то время я уже тащила в чат всё подряд: тексты, идеи, вопросы, которые давно собиралась кому-нибудь задать, и задачи, до которых обычно не доходили руки.

Кажется, тогда я впервые поняла, насколько опасно давать гуманитарию технологию, которая умеет поддерживать разговор.

Я чуть было не влюбилась.



Разумеется, не в буквальном смысле. Скорее в самую возможность. В ощущение, что рядом внезапно появился инструмент, с которым можно не только искать готовые ответы, но и думать вслух. Я приходила с одной мыслью, уходила с пятью новыми, потом возвращалась проверить шестую — и всё это довольно быстро превратилось в привычку.

До программирования в тот момент было ещё далеко.

Идея Telegram-бота появилась позже и по гораздо более прозаической причине. Мне нужен был помощник, который мог бы выполнять конкретные задачи прямо в Telegram. Никакой революции в области искусственного интеллекта я устраивать не собиралась. Мне хотелось всего лишь удобного бота-помощника.

Словосочетание «всего лишь» я теперь вспоминаю с некоторой нежностью.

Со стороны задача выглядела вполне безобидно. Telegram-ботов вокруг было полно. Они присылали сообщения, принимали команды, публиковали посты, отвечали пользователям. Очевидно же, что кто-то их делает. Значит, где-нибудь должна существовать инструкция, по которой один такой бот собирается.

Я нашла инструкцию.

Первые несколько шагов даже не вызвали тревоги. Нужно было создать бота в Telegram, получить токен, затем каким-то образом связать всё это с нужным сервисом. Я переходила по пунктам, копировала то, что предлагалось скопировать, вставляла туда, куда предлагалось вставить, и довольно быстро добралась до момента, когда вся конструкция должна была наконец ожить.

Не ожила.

Вместо результата я получила ошибку.

Я перечитала сообщение. Потом инструкцию. Вернулась на несколько пунктов назад, всё проверила и повторила попытку. Ошибка осталась.

Вот здесь закончилась понятная часть инструкции. Пока всё шло по плану, я чувствовала себя почти уверенно: нажмите сюда, скопируйте это, вставьте туда. Но стоило появиться ошибке, как выяснилось, что дальше нужно уже не просто повторять чужие действия, а понимать, что именно происходит.

Я полезла искать причину и почти сразу столкнулась с первым словом, которое раньше встречала разве что мельком: API. Ладно. Что такое API?

Открыла новую вкладку. Через несколько минут рядом с API появился API key, потом token, request и endpoint. Я переходила из одной вкладки в другую, пытаясь собрать из этих понятий хоть какую-нибудь связную картину.

API в одном из объяснений сравнивали с официантом, который принимает заказ и несёт его на кухню. Эта версия показалась мне вполне дружелюбной, но через пару абзацев официант обзавёлся ключами, токенами, запросами и конечными точками, и ресторан снова превратился во что-то подозрительно техническое.

Особенно впечатлил endpoint — «конечная точка». Я перечитала объяснение и подумала, что конечная точка сейчас нужна в первую очередь мне самой.



На столе остывал кофе. В браузере множились вкладки. В одной объясняли API, в другой спорили про токены, в третьей кто-то на форуме советовал проверить заголовки запроса. Я честно пыталась разобраться, но попытка вернуться к первоначальной задаче показала, что бот за время моего интеллектуального путешествия работать сам по себе почему-то не начал.

Я откинулась на спинку кресла и посмотрела на экран. Вообще-то мне нужен был всего лишь небольшой и удобный Telegram-бот-помощник. Я не собиралась запускать спутник, строить ракету или изобретать новый интернет. Мне хотелось, чтобы бот выполнял несколько полезных действий, а вместо этого я уже довольно долго выясняла, что такое API и куда отправляется request.

Где-то здесь здравый смысл предложил закончить эксперимент. Я почти согласилась, потянулась к чашке, сделала глоток давно остывшего кофе и машинально посмотрела на соседнюю вкладку.

Там был открыт Claude.

Несколько секунд я просто смотрела на экран, а потом до меня дошло.

— Стоп. У меня же есть Claude.

Я вернулась в чат и описала задачу. Рассказала, что хочу сделать Telegram-бота-помощника, объяснила, что он должен уметь, и честно призналась, что совершенно не понимаю, как всё это собрать.

Claude начал объяснять, и ситуация перестала выглядеть безнадёжной. Я могла остановиться на любом месте, переспросить, попросить объяснить проще, уточнить, зачем нужен конкретный шаг. Вместо десятка вкладок у меня снова появился один собеседник и одна задача.

А потом Claude написал код.

Я долго смотрела на экран. Там были строки, скобки, кавычки, какие-то английские слова и отступы. По отдельности почти ничего из этого не выглядело пугающе. Но вместе всё превращалось в язык, на котором я не говорила.

Я прокрутила код вниз, потом обратно вверх. Он выглядел убедительно. Это было всё, что я могла сказать о нём с уверенностью. Если бы в тот момент кто-нибудь спросил меня, хороший ли это код, я, наверное, оценила бы его исключительно эстетически: строчки аккуратные, отступы ровные, скобки вроде на месте.

Значит, оставалась сушая мелочь — запустить.

И тут выяснилось, что между «у меня есть код» и «у меня работает бот» располагается довольно значительная территория. Коду требовалось место, где он будет выполняться. Что-то нужно было установить, что-то настроить, что-то подключить. Потом снова появился API, понадобился ключ, и выяснилось, что его надо куда-то правильно поместить.

Я шла за Claude практически вслепую: он писал, что нужно сделать, я делала. Наконец наступил момент запуска. Я нажала — и получила ошибку.

Теперь хотя бы было понятно, кому её нести. Я скопировала сообщение и отправила Claude. Он объяснил причину и предложил исправление. Я исправила, запустила снова и получила новую ошибку. Скопировала её, получила новый ответ, снова исправила. Через некоторое время процесс приобрёл устойчивый ритм: я запускала код, получала ошибку, несла её Claude, возвращалась с исправлением и через пару минут приносила следующую. Это уже начинало напоминать не разработку бота, а странную эстафету.

Я посмотрела на часы, потом на ноутбук и обнаружила, что вечер каким-то образом успел закончиться. На экране оставался код, который я не умела читать. Рядом был чат с длинной историей наших попыток, а в браузере теснились вкладки с документацией и форумами. Бот всё ещё не работал. Зато теперь я знала, что существуют API, токены, запросы и endpoint. Не понимала толком ни одного из них, но знала. Тоже результат.

В голове возникла очень точная характеристика моего состояния:

humanitarian.exe — not responding

Я закрыла глаза и решила, что на сегодня достаточно.

Телефон подал сигнал. На экране появилось уведомление: «4-я Битва роботов в России». Я прочитала и уже собиралась убрать телефон, когда взгляд зацепился за цифру. Четвёртая. Я вернулась к уведомлению и перечитала ещё раз.

— Четвёртая?!



Больше всего меня поразил даже не сам факт существования Битвы роботов, а её порядковый номер. Получалось, что первая, вторая и третья каким-то образом успели состояться без моего ведома. Люди уже не первый год собирали роботов, выводили их на арену, устраивали соревнования, приходили смотреть на бои — а я только сейчас обнаружила, что всё это вообще происходит.

Я открыла информацию о соревновании, потом ещё одну страницу. Роботы были настоящими. Команды — тоже. Машины строили, готовили к боям, привозили на соревнования. Всё это не выглядело как анонс далёкого будущего.

Я сидела с телефоном в руке и смотрела на ноутбук, где всё ещё был открыт код моего неработающего Telegram-бота. Ещё несколько часов назад всё это казалось набором отдельных вещей: нейросеть, бот, какие-то непонятные технические слова, ошибка, которую я никак не могла победить. Теперь они вдруг сложились в одну картину.

Мысль оказалась настолько простой, что я даже удивилась, почему не замечала этого раньше: будущее уже наступило. То самое будущее, которое когда-то представлялось чем-то далёким и почти фантастическим, давно стало обыкновенным настоящим. Пока я воспринимала нейросети и роботов как отдельные любопытные новости технологического мира, где-то уже успели провести три Битвы роботов и готовились к четвёртой. А на моём ноутбуке лежал код Telegram-бота, который ещё утром в моей жизни вообще не существовал.

Можно было закрыть ноутбук и больше к этому не возвращаться. В конце концов, бот был нужен мне не настолько, чтобы ради него осваивать совершенно незнакомую область. Можно было найти другой способ, попросить кого-нибудь помочь или вовсе отказаться от идеи. Но теперь мне уже хотелось разобраться. Несколько часов назад я просто пыталась заставить работать Telegram-бота, а к концу вечера обнаружила целый технологический мир, о котором почти ничего не знала. И этого оказалось достаточно, чтобы вместо закрытого ноутбука передо мной снова появилась открытая вкладка браузера.

Тогда я ещё понятия не имела, с чего именно начинают люди, которые хотят разобраться в программировании. Не знала, что мне понадобится, сколько времени это займёт и чем вообще закончится. Я знала только одно: мне хотелось понять чуть больше, чем вчера.

Так началась история, к которой я совершенно не готовилась.



Где-то внутри системы запустился новый процесс.

