

*Борис Орехов*

## РЕЦЕНЗИЯ НА КНИГУ МАРКА К. МАРИНО «КРИТИЧЕСКИЕ ИССЛЕДОВАНИЯ КОДА»

Рец. на кн.: *Mark C. Marino* Critical code studies. Cambridge, Massachusetts: The MIT Press, 2020. xii, 275 pp.

Книга Марка К. Марино существует не только в академическом контексте: хотя она издана университетским издательством MIT Press, одновременно следует учитывать, что вышла она в серии «Software Studies», редактором которой является известный медиахудожник Лев Манович, иногда мимикрирующий под представителя ученой среды, но в большинстве случаев последовательно от нее дистанцирующийся. На некоторое нескучное «хулиганство», сигнализирующее о выходе книги за рамки конвенциональных научных традиций, указывает каламбурическое преобразование названия раздела благодарностей: Hacknowledgments вместо Acknowledgments. Задача серии поставлена масштабно: отследить, как программное обеспечение интегрируется в процессы современной культуры и общества (p. ix).

Основная идея введения, задающего направление в восприятии всей книги, — код в наше время больше, чем код. Это уже не подсобный технический инструмент, а самостоятельный актер, который, если научиться его читать, объясняет, почему нечто в нашем мире устроено так, а не иначе. Про код сказано: «Он вышел за пределы сферы программирования и проник в сферу мейнстримных медиа и партизанской политической блогосферы» (p. 3). Понимание того, как написан код, не даст человеку обмануться в сомнительных и спорных вопросах (в пример приводится проблема изменения климата).

---

*Борис Валерьевич Орехов*

Институт русской литературы (Пушкинский Дом) РАН

nevmenandr@gmail.com

Вот центральное утверждение введения: «Код — это социальный текст, значение которого развивается и трансформируется по мере того, как с ним сталкиваются новые читатели, с течением времени и по мере изменения контекста» (р. 5). Утверждение, как мы видим, сделанное в рекламной интонации, и, как и всякое подобным образом организованное высказывание, опирающееся на некоторое рекламное преувеличение. С одной стороны, приведенный в книге пример того, как программиста неправильно поняли, и из этого разросся медийный скандал, пример частный, у него не так уж и много инфоповодов-братьев. Как много мы можем вспомнить случаев, когда прочтение кода становилось причиной массового бурления в СМИ? Вопрос риторический. Таких случаев практически нет. С другой стороны, код и в самом деле текст, и в самом деле текст, который может быть рассмотрен в социальном аспекте. А что не может? И инструкция к кофеварке, и коммерческое объявление на заборе, и этикетка на одежде — это социальный текст, значение которого развивается и трансформируется по мере того, как с ним сталкиваются новые читатели. Программный код ничем не хуже и не лучше этих примеров с точки зрения проявления социальности, но, конечно, более завлекателен с точки зрения интереса потенциальных читателей книги по этой теме. Вокруг инструкций к кофеваркам пока не сформировалось своей мифологии, операторы кофеварок пока не выработали яркого габитуса. Так что «Критические исследования инструкций к кофеваркам» Л. Мановичу и М. К. Марино было бы гораздо сложнее «продать» издателям и продать публике. Мой скепсис в данном случае не означает, что критических исследований кода не должно быть или что они бессмысленны. Такие исследования должны быть, и они осмысленны, но во многом безответственные и даже пустые утверждения, преувеличивающие то, что не нуждается в преувеличении, вызывают легкое раздражение. Да, давайте исследовать код, делать это критически, но стоит ли ради этого педалировать какие-то окказиональные случаи непонимания журналистами программистов и настаивать на какой-то глобальной роли программного кода в жизни человека? На мой вкус, это все же лишнее. Это ведь уже книга, а не заявка на издательский грант: можно говорить со своим читателем в более сдержанном регистре.

Надо учесть еще и то обстоятельство, что на момент издания этой книги единственным субъектом написания кода был квалифицированный программист. Теперь ситуация изменится, поскольку в этой роли теперь все увереннее выступают большие языковые

модели. Проследим, как эти изменения должны повлиять на высказанные в книге положения.

В параграфе «Собеседование» автор демонстрирует нам нетривиальные углы взгляда на код. В частности, разные программы, решающие одну и ту же задачу, написанные на одном и том же языке программирования, могут быть ранжированы по силе, и сила эта определяется не объективными критериями, а социальным контекстом. Мысль не то чтобы оригинальная, но иметь ее в виду полезно. Правда, разыграв перед читателем небольшую сценку, в которой вымышленные персонажи-программисты для того, чтобы устроиться на работу, выполняют тестовое задание, автор не доводит сюжетную ветку до конца, скрыв от нас, кто же в итоге получил работу. Это, конечно, не претензия, которую можно все-речь предъявлять книге, но вот такая эмоция неудовлетворенности читательского любопытства окрашивает восприятие. Не могу поручиться, что эффект ненамеренный, потому что так получается не только в этом параграфе, но и каждые 10–15 страниц: автор не договаривает, не отвечает на поставленные им же самим вопросы, бросает свои же размышления на середине.

Дальнейший метод изложения я бы назвал кейс-дроппингом по аналогии с нейм-дроппингом, когда говорящий заваливает свою аудиторию именами имеющих и не имеющих отношения к делу людей, не всегда успевая пояснить, зачем то или иное имя вообще прозвучало. Мы узнаем про протестный плакат с кодом на языке С, про риторический подтекст феминистского приложения на PHP, про личный исследовательский опыт автора в области изучения кода и т.д. Словом, бомбардируя нас эксцессами, автор стремится создать впечатление легитимности научного поля, в котором предметом исследования становится программный код, у которого за скобки интереса изучающего выведено то, зачем этот код написан, его функционал.

Надо сказать, что лично меня в этом убеждать и не требовалось, я верю в то, что такое научное поле имеет право на жизнь. Так что довольно серьезная и явным образом важная для автора часть книги работает на холостых оборотах.

Более того: я и сам внес некоторый вклад в наблюдение за процессами, о которых говорит М. К. Марино. В свое время меня заинтересовало, как организован код, который одновременно является и стихотворным произведением [Орехов 2015]. За исключением того, что я практически не затрагивал социальных контекстов таких программ, все это имеет к тематике рецензируемой

книги самое прямое отношение. Но М. К. Марино на мою статью не ссылается. Разумеется, статья написана по-русски, и поэтому выходит за пределы привычного автору языкового мира, но ведь в соответствии с требованиями журнала, к статье мной приложены аннотация и ключевые слова в переводе на английский. Зачем вообще журналы требуют англоязычные аннотации, если их все равно никто не читает? Еще один риторический вопрос.

К сожалению, моя статья не единственный важный для поля текст, который упустил М. К. Марино. Следовало бы включить в обзор и во многом близкую ему идейно благодаря социально-политическому и экономическому взгляду на предмет статью М. Блэка об истории кода Mozilla [Black 2015]. Кстати, М. Блэк в этом тексте, вышедшем, как и мой, в 2015 году, на более ранние работы М. К. Марино и Л. Мановича ссылается. Но обратной любезности мы не видим. Но дело, разумеется, не в любезностях, а в том, что статью М. Блэка приходится признать более содержательной, методологически выверенной и системной, чем книга М. К. Марино. Анализируя исходный код веб-браузера, автор статьи противопоставляет интерфейс программного обеспечения как то, что видимо пользователю, и исходный код, который от пользователя скрыт. Такая подспудность исходного кода — это вообще черта эпохи, ведь в период раннего распространения персональных компьютеров пользователи вынуждены были сами становиться программистами, чтобы управлять информационными процессами, а современный пользователь часто лишен понимания, как работает вызываемая им функция, потому что перед глазами у него только элемент графического управления (грубо говоря, кнопка). Честно говоря, такое описание гораздо лучше согласуется с моим пользовательским опытом, чем не слишком убедительные вбросы М. К. Марино про необычайную важность программного кода: «аудитория кода расширилась, поскольку судьи и юристы, политики и эксперты, и даже поэты включают код в свой дискурс» (р. 38). Все же этот код, если не брать во внимание специалистов, мало кто всерьез стремится изучать: графический интерфейс уверенно завоевал и не желает отдавать свои позиции, а судьи, юристы, политики и эксперты сталкиваются с кодом в исключительных ситуациях, так что сделать вывод о драматическом расширении аудитории можно только по итогам cherry picking. Другая уместная, хотя и гораздо менее текстоцентричная, чем у М. Блэка, статья [Alleyne 2011], автором книги также пропущена.

М. К. Марино стремится деконструировать, то есть разрушить и собрать заново на новых основаниях, наши представления о коде. Прямолинейный подход к коду выглядит как серия вопросов к нему типа «что делает этот код?», «как код решает свои задачи?» Это вопросы технические. А для М. К. Марино важнее социальные: «кто написал этот код?», «зачем он нужен?» В этом смысле книга «Критические исследования кода», конечно, в философском смысле постмодернистская. Но не только в философском. Автор открыто декларирует ненейтральность в отношении предмета изучения, к которому он подходит с позиций того, что Х. Плакроуз и Дж. Линдси перечисляют в своей книге как краеугольные камни постмодернизма: постколониальные исследования, критическая (наверное, отсюда и «критическое» в заглавии книги) расовая и квир теория, феминизм [Плакроуз & Линдси 2022, с. 58–59]. Само по себе это допустимо, но, к сожалению, книга М. К. Марино несет на себе все негативные отпечатки, свойственные такого рода идеологически заряженным текстам: сбивчивость аргументации, непоследовательность рассуждений (выше я уже показывал, как автор сначала рисует для нас образ двух человек, поступающих на работу, а потом про них забывает — это довольно рутинный в этой книге повествовательный прием), подмена рациональных положений эмоциями, а логических рассуждений — политическими лозунгами.

Не лишено любопытства рассуждение автора о том, можно ли читать код литературоведчески — как мы читаем поэзию, прозу и драму (р. 30–31). Ответ, как всегда у постмодернистов, не прямой. Вроде как и да, можно, но не совсем, потому что тогда код уже не совсем текст, а культурный объект, но как текст он тоже представляет интерес. Я так понимаю, читатель может выбрать ответ по своему вкусу, а М. К. Марино избегает однозначных формулировок, потому что они задают столь ненавистные постмодернистам иерархии и вообще выдают токсичную репрессивность в мышлении. Единственное, что М. К. Марино знает наверняка и что готов утверждать со всей возможной определенностью, — это то, что код важен, имеет социальную проекцию и его обязательно надо изучать критически.

Вообще сопоставление и противопоставление поэзии и кода — сквозной мотив книги. В ней есть параграф «Code Is Not Poetry (or at Least, Most of It Isn't)», где начинаются рассуждения на эту тему. В качестве того, что объединяет эти два жанра, почему-то называется игра с языком и нашими восприятиями. Сомнительно

тут то, что все это не является специфическим для поэзии: точно так же поступает и проза. А вот разводит код и поэзию функциональность. Но этим все не ограничивается, ведь в книге есть целая глава «Generative Code» с параграфом «Code and Poetry», где появляется концепция кода, специально написанного как поэтический текст, и в этом смысле противопоставленного коду, который служит для порождения поэтического текста. Но почему-то поэтическая форма ассоциируется у автора с какими-то поверхностными признаками, вроде краткости именования функций. Поэтические тексты вообще-то включают и длинные слова тоже, см. «Гиперпэон» Д. Андреева<sup>1</sup>.

М. К. Марино упорно настаивает: «Как исследователь новых медиа, получивший образование в области литературоведения, я хотел бы предложить, чтобы мы больше не говорили о коде как о тексте в метафорическом смысле, а начали анализировать и объяснять код как текст, как знаковую систему со своей собственной риторикой, как семиотическую коммуникацию, обладающую значением, выходящим за рамки ее функциональной полезности» (р 39). Да, говорить о коде только с точки зрения того, в чем его функциональная нагрузка, действительно слишком прямолинейно. Исследователь должен быть гибче, и рассматривать код как текст в самом деле может оказаться продуктивным ходом. И далее: «По сути, я предлагаю, чтобы мы могли читать и анализировать код так же, как мы могли бы анализировать литературное произведение или другие тексты». Но если с этим мы согласны, больше автору нам, к сожалению, нечего предложить. Как именно анализировать? Ведь в литературоведении сложились и алгоритмизировались устойчивые практики анализа и интерпретации. Литературовед на примерах знает, с какой стороны ему подойти к тексту. А какими путями следует идти исследователю кода? Вот у М. Блэка есть свой ответ и образец. А М. К. Марино, увы, отделяется общими фразами. Как анализировать? Как литературовед. Что анализировать? Все. Декларативности в книге больше, чем рецептурности.

Что именно попадает в сферу внимания автора? Transborder Immigrant Tool (TBIT), мобильное приложение для дешевых телефонов, предназначенное для мигрантов, пересекающих пустыню на границе США и Мексики. Оно хорошо соотносимо с постколониальными исследованиями. Важно тут не то, какой у приложения интерфейс, а то, что благодаря своей цели код звучит как политическое высказывание. При этом собственно анализа кода мы почти не видим, только замечание, что стандартная фраза из лицензии

(про то, что это ПО предоставляется без гарантий, р. 74) в приложении к ситуации мигрантов звучит трагично-иронично. Здесь подразумевается, что медитативное чтение кода позволяет мысленно пройти путь мигранта.

Нишевый язык программирования FLOW-MATIC. Как и в прочих случаях, автор анализирует не столько код, сколько идеологию, стоящую за созданием языка, максимально похожего на английский. Тут М. К. Марино тоже берет на вооружение колониальную оптику, и задается вопросом, насколько хорошо то, что функции и операторы называются словами английского языка. Идеологически, конечно, в этом будут изъяны, хотя языки программирования для того и появились, чтобы облегчать человеку работу с понятными ему словами. Есть примеры локальных языковых демаршей, среди которых особенно интересен перевод C# на гавайский язык *ʻōlelo Hawaiʻi* (р. 154), которому, к сожалению, суждено остаться местной достопримечательностью. В этом проекте не просто операторы имеют другое название, но и в основу обозначения управляющих конструкций ложатся другие, непривычные, метафоры. Например, условная конструкция *if-then-else* становится разветвляющимся ручьем (*kahawai, makawai, muliwai*).

В остальных главах идеология также заслоняет собой анализ текста. Как исследователь, применяющий цифровые методы в гуманитарных науках и получивший образование в области литературоведения, я не усматриваю у М. К. Марино литературоведческого интереса к тексту программы. Скорее, речь должна идти об интересе к социальным теориям, органически связанным с постмодернизмом (как его понимают Плакроуз и Линдси), а код в этой системе величина во многом случайная. Примерно те же выводы автор смог бы сделать на любом совершенно произвольном материале.

Возвращаясь к тому, как корректируют выводы автора большие языковые модели, приходится признать, что жесткая связь кода, данного реципиенту через чтение, и социального существенно обедняет концептуальный план книги. Ведь у больших языковых моделей нет интенциональности, а, следовательно, и их сфера социального действия значительно сужена. Пока был программист, который пишет код, он мог быть рассмотрен во всей совокупности его социальных контекстов, а в эпоху вайбкодинга эта сфера стала гораздо менее определенной.

Итог, в общем, очевиден из предыдущего изложения: несмотря на поставленную автором задачу выработки принципов критического чтения программного кода, книга проявляет себя в совсем

ином жанре, это скорее кунсткамера (или гербарий), собирающая вместе в меру интересные, в меру забавные случаи социального измерения в функционировании программного кода. Системности, последовательности и настоящей научной добросовестности в таком собрании пестрых глав не будет, чего мы и должны ожидать от книги, изданной в серии, редактируемой медиахудожником. Эта книга в меньшей степени представляет собой научное исследование, а в большей — замаскированный под академическую монографию каталог выставки современного искусства, одновременно выступающий ее экспонатом. Хорошо, что такая книга появилась, она, несомненно, должна была найти свою аудиторию.

### *Примечания*

<sup>1</sup> О сущностных признаках поэзии см. [Орехов 2026, с. 23–27].

### *Литература*

#### *Исследования*

*Орехов 2026* — Орехов Б. В. Электронная лира. Как и зачем искусственный интеллект пишет стихи? СПб.: Издательство Европейского университета в Санкт-Петербурге, 2026.

*Орехов 2015* — Орехов Б. В. Стихи в программном коде: современный опыт и методика анализа // Критика и семиотика. 2015. № 1. С. 377–389.

*Плакроуз & Линдси 2022* — Плакроуз Х., Линдси Д. Циничные теории. Как все стали спорить о расе, гендере и идентичности и что в этом плохого. М: Individuum, 2022.

*Alleyne 2011* — Alleyne B. Challenging Code: A sociological reading of the KDE Free Software project // Sociology. 2011. Т. 45. №. 3. P. 496–511.

*Black 2015* — Black M. A Textual History of Mozilla: Using Topic Modeling to Trace Sociocultural Influences on Software Development // Digital Humanities Quaterly. 2015. Volume 9. Number 3. URL: <https://dhq.digitalhumanities.org/vol/9/3/000224/000224.html>, DOI: <https://doi.org/10.63744/qhcju89xj5yf>