

ПСИХОЛОГИЯ ОБРАЗОВАНИЯ EDUCATIONAL PSYCHOLOGY

Научная статья | Original paper

Преодоление знаковой натурализации при обучении школьников программированию и работе в информационных системах

А.И. Федосеев¹ 

¹ Ассоциация участников технологических кружков, г. Москва, Российская Федерация
 fedoseev@kruzhok.org

Резюме

Контекст и актуальность. Предложенный С. Пейпертом конструкционистский подход в настоящее время является доминирующим при обучении школьников программированию, в том числе реализован в таких образовательных средах, как Scratch и LEGO Mindstorms. Это соответствует эволюции цифровых человеко-машинных систем, которые вместе с популярностью и ориентацией на неподготовленных пользователей реализуют в качестве ведущего метод проб и ошибок и не направлены на развитие учащихся. **Цель.** Рассмотреть явление знаковой натурализации при обучении работе в информационных системах и возможность его преодоления. **Гипотеза.** Подходом к преодолению знаковой натурализации учащимися может стать моделирование исходного отношения осваиваемой системы программируемых вычислений на основе конкретного языка программирования. **Методы и материалы.** Данная теоретическая статья опирается на принципы мыследеятельностной педагогики и идеи развивающего образования В.В. Давыдова, В.В. Рубцова, Ю.В. Громыко. В статье рассматриваются история вопроса и современные научные представления о преодолении знаковой натурализации при работе в информационных системах, в том числе посредством организации деятельности моделирования на материале игровых и учебных ситуаций с использованием современных киберфизических систем и сред. **Результаты.** Предметом моделирования становится отношение идеально-го компьютера (notional machine) к программируемой, регулируемой системе автоматизированных материальных процессов в разных практических контекстах. **Выводы.** Рассмотренный в статье подход при последующем исследовании, реализации и внедрении в образовательный процесс дает возможность выстроить альтернативную среду развития мышления учащихся при работе в цифровых средах и обучении программированию.

Ключевые слова: знаковая натурализация, конструкционизм, моделирование, мыследеятельност-ный подход, мышление, операции в цифровой системе, программирование, развивающее образова-ние, рефлексия, человеко-машинные системы

Для цитирования: Федосеев А.И. (2025). Преодоление ограничений конструкционизма и знаковой натурализации при обучении школьников программированию. *Культурно-историческая психология*, 21(4), 59–69. <https://doi.org/10.17759/chp.2025210406>

Overcoming the sign naturalization in learning programming and working in information systems

A.I. Fedoseev¹ 

¹ Kruzhok Association, Moscow, Russia
 fedoseev@kruzhok.org

Abstract

Context and relevance. The constructionist approach proposed by S. Papert is currently dominant in teaching students programming, including implemented in educational environments such as Scratch and LEGO Mindstorms. This corresponds to the evolution of digital human-machine systems, which, along with their popularity and focus on untrained users, implement trial and error as a leading method and are not aimed at student development. **Objective.** The goal is to consider the phenomenon of sign naturalization in learning programming. **Hypothesis.** An approach to overcoming the phenomenon of sign naturalization involves solving practical problems while modeling the original relations of the mastered computational system based on a specific programming language. **Methods and materials.** This theoretical study is based on the principles of cognitive and activity-based approach, thinking and the ideas of developmental education by V.V. Davydov, V.V. Rubtsov, and Y.V. Gromyko. The article examines the history of the issue and modern scientific ideas about overcoming sign naturalization when working within information systems, including the organization of modeling activities based on game- and educational-based situations using modern cyber-physical systems and environments. **Results.** The subject of modeling is the relationship of a notional machine to a programmable, regulated system of automated material processes in various practical contexts. **Conclusions.** The approach considered in the study being researched and implemented in the educational process makes it possible to build an alternative environment for the development of students' thinking when working in digital environments and learning programming.

Keywords: sign naturalization, constructionism, modeling, cognitive and activity-based approach, thinking, operations in digital system, programming, developmental education, reflection, human-machine systems

For citation: Fedoseev, A.I. (2025). Overcoming the sign naturalization in learning programming and working in information systems. *Cultural-Historical Psychology*, 21(4), 59–69. <https://doi.org/10.17759/chp.2025210406>

Введение

Обучению программированию в школе уже больше полувека. Информатика, программирование, информационные технологии, цифровая грамотность — акцент менялся, но всякий раз речь шла о необходимости познакомить школьника с непрерывно растущим и усложняющимся миром цифровых систем, научить его применять цифровые технологии и включать их в осмысленную жизнь людей. Академик А.П. Ершов, один из отцов-основателей урока информатики, писал в своем манифесте начала 1980-х о программировании как о *всеобщей второй грамотности* (Ершов, 1983) — уже тогда компьютеры и программирование виделись как новый способ познания и среда развития ребенка. Всемирно известный педагог и энтузиаст обучения детей программированию Сэймур Пейперт сформулировал следующую педагогическую максиму (Papert, 1980): сам способ обучения не только программированию, но и другим наукам должен измениться — на смену готовому знанию должна прийти работа по самостоятельному конструированию знания ребенком, и незаменимым инструментом в этом становится программируемый компьютер.

Педагогическая инновация Пейперта — *конструкционизм* — стала ответом на традиционный подход к обучению школьников программированию, который С. Пейперт называл *инструкционизмом*, подразумевая обучение посредством инструкций учителя. Вместо этого Пейперт предложил создавать ситуации, в которых ребенок будет самостоятельно *конструировать* новые знания. Для того, чтобы на практике исследовать процессы развития ребенка, Пейперт с коллегами из Массачусетского технологического института создал игровой язык программирования LOGO, позволяющий рисовать фигуры на экране компьютера при помощи программируемого движения маленькой черепашки.

Еще в 1980-е и 90-е ряд исследователей за рубежом и в нашей стране показали ограничения подхода Пейперта при обучении программированию и информатике (Реа и др., 1985), а также в более широком контексте развития ребенка (Давыдов, Рубцов, 1990). Это не умаляет революции, которую данный подход совершил в развитии педагогической психологии, дав уникальный пример формирования условий для *самостоятельных* учебных действий ребенка. Предложенный Пейпертом способ работы и обучения за компьютером был больше похож не на системное обучение инструменту, но на игру или *брюкколаж* (Turkle, Papert, 1990).

Несмотря на скептическое отношение многих исследователей и педагогов, годы спустя этот прием был тиражирован в рамках школьной LEGO-робототехники (Resnik и др., 1988) и в программировании в визуальных средах, наиболее популярной из которых является Scratch¹. Сейчас эти инструменты стали повсеместными: зачастую первые шаги младших школьников в программировании начинаются в Scratch или вместе с роботами LEGO Mindstorms, а с 2022 года в России — с продолжающими эти идеи аналогами (Овсянников и др., 2023). Педагогика конструкционизма находит также продолжение в «фаблабах» — средах цифрового производства — через создание собственными руками материальных воплощений цифровых моделей (LaParde, Lassiter, 2023).

Популярность конструкционизма не случайна и опирается на эволюцию компьютеров и ведущего способа работы пользователей за компьютером. Вместе с формированием массовой культуры общества потребления и ростом популярности цифровых устройств все меньшие требования выдвигались к профильным знаниям и культуре пользователя. Не без влияния идей Пейперта самостоятельное исследование пользователем доступного интерфейса, действие методом «проб и ошибок» стали основным способом работы. «Не заставляйте пользователя думать» — широко распространенный девиз дизайнеров интерфейсов цифровых систем в течение более чем 30 лет (Krug, 2000).

Анализ эволюции операционных систем показывает, что пользователю современного компьютера или смартфона стал предлагаться ограниченный набор конкретных операций или *процедур*, что позволяет совершать только заложенные разработчиком действия в системе (Курячий, 2004). Такие *процедурные* человеко-машинные системы опираются на самостоятельные пробы в опоре на графический интерфейс и подкрепление правильного поведения пользователя, манипулирующего виртуальными объектами. Благодаря этому требования к предварительным знаниям, необходимым для работы, снижаются, а самообучение базируется на конструкционистском подходе.

С психологической точки зрения такой тип систем предлагает пользователю ограниченный набор операций со знаками в неизменных знаковых системах, предназначенных для решения конкретных практических задач. При этом за рамками работы пользователя остается восстановление оснований знаковых систем и предлагаемых операций, пользователь зачастую лишается необходимости моделировать предметное содержание объектов, с которыми осуществляется работа (Давыдов, Рубцов, 1990).

Сама идея того, что школа должна готовить грамотных потребителей, не раз обсуждалась в педагогических кругах и шире. Такой школе современные цифровые технологии только помогают, ведь необходимость специально учить детей пользоваться подобными программами с каждым годом снижается. Дети

с рождения находятся в цифровом окружении и как «аборигены» осваивают новые интерфейсы естественно, перебирая доступные операции (Palfrey, Gasser, 2011). При этом им все сложнее избегать тех шаблонов и ограничений развития мышления, которые таят в себе современные процедурные системы (Федосеев, 2013). Следствиями этого являются и эпистемологический релятивизм, когда место *ошибок* занимает *обратная связь* (Turkle, Pargert, 1990), и распространение среди непрофессионалов явлений вроде «техномагии» — форм невежества, при котором цифровые системы перестают восприниматься как познаваемые инженерные устройства управления.

В оппозицию к потребительскому подходу выдающиеся педагоги — от Пейперта и Ершова до В.В. Давыдова — утверждали, что *школа должна развивать ребенка и учить его мыслить*. Этот подход нашел свое отражение в реализации мыследеяностной педагогики и школы будущего (Громыко, Рубцов, Марголис, 2020). В продолжение идей В.В. Давыдова отечественными педагогами и психологами уже формулировались требования к компьютерным развивающим технологиям обучения, в основу которых могут быть положены *принцип компьютерного моделирования деятельности* (Давыдов, Рубцов, 1990) и обеспечение освоения учащимися культуры мышления (Жегалин, 2007).

Можно ли сохранить предложенную в рамках педагогики конструкционизма учебную субъектность школьника, но преодолеть при этом ограничения современных цифровых систем? Выти из ловушки противопоставления формально-логического «инструкционистского» способа обучения программированию и ограничений конструкционистского самостоятельного движения в опоре на рассудочно-эмпирическое мышление, например через работу с идеальным? Можно ли предложить переход от операций без понимания их содержания к моделированию на материале программирования и работы с цифровыми системами — по аналогии с работами В.В. Давыдова на примере формирования понятия числа или А.А. Устиловской на примере работы с геометрическим объектом (Устиловская, 2008)?

В статье рассматриваются теоретические основания данной проблемы и предлагается возможное направление преодоления ограничений конструкционизма в условиях цифрового образования школьников.

Педагогический подход Сеймура Пейпера

В 1980-е основным способом работы на компьютере было программирование с использованием текстовых языков, в том числе выполнение отдельных операций в командной строке. Такой способ программирования требовал серьезной математической подготовки: основ алгебры (переменные и операции), функционального анализа (функции и рекурсия), формальной логики

¹ <https://scratch.mit.edu>

доказательства теорем (логические условия, последовательное выполнение программы) и др. Это не могло не сказаться на том, что процент школьников, успешно осваивающих информатику, был в значительной степени ограничен. Реализуя гуманитарный проект *массовой компьютерной грамотности*, Сеймур Пейперт представил книге «Mindstorms» альтернативный подход под названием *конструкционизм*² (Papert, 1980). Развивая идеи конструктивизма Жана Пиаже, Пейперт поставил на первое место простые операции учащихся с виртуальным объектом и конструирование ими новых объектов, что позволяло школьникам самостоятельно исследовать доступную виртуальную среду и создавать собственные более сложные операции, тем самым конструировать предмет своей работы и знания о нем.

Это стало возможно благодаря ключевому педагогическому приему Пейперта, реализованному в визуальной среде программирования LOGO — визуальному конструированию — обеспечению прямого соответствия между командами на языке программирования и результатом их выполнения (рис. 1) в виде траектории, которую черепашка рисует на экране, следя командам программы.

Пользователь мог последовательно выполнять команды и одновременно видеть на экране изменения, которые являлись следствием их выполнения. Гипотетически такой подход позволял ребенку строить собственный процесс учения, со временем самостоятельно складывать более сложные операции из элементарных, тем самым формировать различные концепции структурного *императивного программирования* и даже обучаться основам алгоритмического, конструирующего, геометрического мышления. Аскетичные по нынешним временам возможности языка LOGO, предложенного Пейпертом, давали новые возможности педагогу — дать ребенку возможность строить собственный виртуальный мир, обозначая *словами* в программе конкретные действия черепашки. Несмотря на вполне материальное представление объекта на экране компьютера, такой подход позволял выходить на идеальные представления о предмете. Созданная ребенком

команда «ТРЕУГОЛЬНИК», состоящая из более простых команд-операций рисования отрезков, воспроизводила на экране компьютера треугольник согласно прозрачным и строгим математическим правилам. Тем самым знаковые представления в тексте программы и на экране начинали выражать ключевые характеристики идеальных объектов (рис. 2). В опоре на рассмотренный *элементарный шаг* конструирования и самостоятельного именования объекта Пейперт предложил по индукции строить более сложные идеальные объекты.

Реализованный в LOGO конструкционистский подход получил свое развитие в последующих экспериментальных педагогических практиках. Так, М. Бейнон предложил расширить работу с графическими объектами до более сложных операций конструирования динамических объектов с целью формирования идеального представления о процессах (Веупон, 2017). В рамках создания материальных объектов в цифровых производственных пространствах «фаблабов» исследователи из Массачусетского технологического института предложили формировать у школьников первичные представления об онтологиях (LaParde, Lassiter, 2023).

Тем не менее массовое применение LOGO в школе вызвало не только энтузиазм педагогов, но и критику, которая указывала в том числе на ограничения конструкционистского подхода в обучении программированию и формировании связанных с программированием способностей (Реа и др., 1985). К основным направлениям критики можно отнести отсутствие влияния LOGO на формирование таких базовых способностей программиста, как планирование и решение задач, а также сложности с переносом знаний и навыков из области программирования на другие учебные ситуации, в том числе при погружении в профессиональный контекст. Но к наиболее важным замечаниям стоит отнести критическую значимость педагога в проведении таких занятий. Еще Пейперт отмечал важность подготовки педагога, который сможет сопровождать детей, вставших на путь конструирования нового знания. На первый план выходят такие аспекты работы педагога, как сопоставление идеаль-



Рис. 1. Связь, лежащая в основе конструкционистского подхода С. Пейперта



Fig. 1. The main relation of the constructionist approach by S. Papert

² Конструкционизм (constructionism) С. Пейперта важно не путать с конструктивизмом (constructivism) Ж. Пиаже.

РИСУЕМ ТРЕУГОЛЬНИК
ВПЕРЕД 100
НАПРАВО 120
ВПЕРЕД 100
НАПРАВО 120
ВПЕРЕД 100
КОНЕЦ

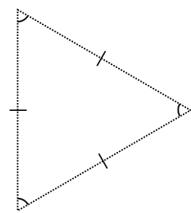
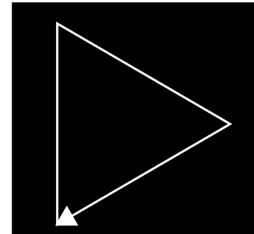


Рис. 2. Конструирование треугольника программой и идеальные характеристики геометрического объекта

TO TRIANGLE
FORWARD 100
RIGHT 120
FORWARD 100
RIGHT 120
FORWARD 100
END

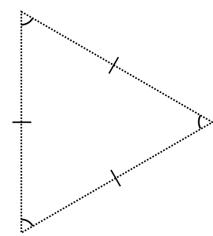
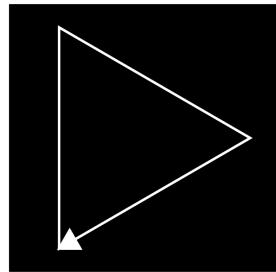


Fig. 2. The construction of a triangle and the ideal characteristics of the geometrical object

ного содержания с контекстом визуальной среды, работа с идеальным представлением о вычислительной системе. Конструирование собственных действий из элементарных операций при работе с определенной предметной областью и практической задачей предполагает выход учащихся в процесс моделирования, что выдвигает высокие требования к цифровой системе и педагогу (Рубцов, Марголис, Пажитнов, 1987).

Сегодня идеи Пейперта нашли свое воплощение не только в начальной, но и в средней школе — в виде среды визуального программирования Scratch, в которой школьники переходят к созданию мультфильмов и простых видеоигр, и в пластиковых роботах LEGO Mindstorms (Resnick и др., 1988) и их современных аналогах, которые школьники программируют для езды по линии и выполнения других простых действий. Среда визуального программирования Scratch (рис. 3)

реализует тот же принцип связи программы и управляемого ей объекта, только язык и сама среда, реализованная в виде современного графического интерфейса, куда более разнообразна, чем текстовый интерфейс и черепашка в LOGO. При этом *написание* программ сменилось их *составлением*, ведь учащимся не нужно именовать какие-то новые сущности, достаточно лишь выбирать и комбинировать имеющиеся характерные блоки при составлении программы. С психологической точки зрения это изменение фундаментально: вместо продумывания и ввода новых понятий в опоре на существующие (как в примере про рисование треугольника, которое состояло из трех перемещений и поворотов), учащиеся стали визуально комбинировать знаки, за которыми скрываются отдельные операции над объектами, в определенной знаковой системе из слов, иконок и числовых параметров.



Рис. 3. Интерфейс Scratch: та же связь между программой и результатом ее выполнения, но в куда более богатом окружении

Fig. 3. The user interface of Scratch: we see similar connection between the program and its result in a rich visual environment

Такой интерфейс выполнен в традициях современного дизайна цифровых систем: пользователь работает с графическими объектами, выбирает и перемещает их с помощью мыши. Для детей, с детства привыкающих к графическим и тактильным интерфейсам на экране компьютера и смартфона, такой способ создания программ может показаться более привлекательным.

Однако требует более подробного рассмотрения вопрос о *знаковой натурализации*, возникающей вследствие оперирования объектами без выделения лежащей за операциями структуры действий и сущностных характеристик рассматриваемых объектов (Устиловская, 2008), к чему подталкивает в том числе визуально богатый интерфейс. Рассмотрим подробнее явление знаковой натурализации, в том числе при работе в современных информационных системах.

Знаковая натурализация в информационных системах

Явление знаковой натурализации рассматривается в педагогической психологии в контексте различения освоения конкретных операций в ходе решения практических задач и восстановления обучающимся исходных отношений, лежащих в основе данных операций, и связанных с ними идеальных характеристик (Давыдов, 1996; Медведев, 2010). Натурализация (овеществление) сопровождает практическую работу в заданной системе операций, если для работы нет необходимости восстанавливать ее устройство, заложенные в работу системы законы и идеальные представления. При работе в знаковых системах натурализация означает оперирование знаками без восстановления их исходного содержания.

Сама по себе натурализация в педагогическом контексте может быть необходима как первый шаг в преодолении *вербализма* (т. е. воспроизведения текста без его понимания), что было показано на примере изучения школьниками геометрии (Устиловская, 2008): учащиеся сперва должны начать совершать

предметные действия с чертежом, чтобы вырваться за рамки простого проговаривания, но затем педагогу в целях обеспечения освоения обобщенных способов действия необходимо специально организовать процесс преодоления знаковой натурализации, чтобы выйти к исходным идеальным отношениям, характеризующим геометрические объекты.

В настоящий момент при работе с информационными системами проблема вербализации не стоит так остро³ ведь даже начинающий пользователь совершают простые операции в системе, которые приводят к ее видимым изменениям. Но за богатством интерфейса и возможными сюжетами (например, в Scratch или LEGO) ребенок попадает в ловушку знаковой натурализации: объекты на экране *овеществляются* ребенком — начинают восприниматься непосредственно так, как выглядят, без восстановления лежащего за ними идеального содержания.

Знаковая натурализация — общее явление для современных информационных систем, поскольку со времен появления LOGO произошла революция взаимодействия пользователя с системой. Большинство систем для конечного пользователя, будь то персональный компьютер или смартфон, можно охарактеризовать как *процедурные* (Курячий, 2008). Это касается самого принципа работы пользователя (рис. 4). Если для профессионального программиста компьютер и набор программ являются инструментами, в опоре на которые он сам конструирует необходимую для решения конкретной задачи среду, то пользователь в процедурной системе может решать только те задачи, которые покрываются выбором из списка предлагаемых процедур — возможных операций в системе, встроенных в интерфейс, и их комбинацию. Устройство такой системы не прозрачно для пользователя, поэтому ее поведение предсказуемо только при совершении стандартных действий.

Интересный эффект применения таких систем состоит в том, что заданный список процедур зачастую определяет не только способ решения задачи, но и сильно влияет на саму постановку задач пользователем — внешняя задача трансформируется до возмож-

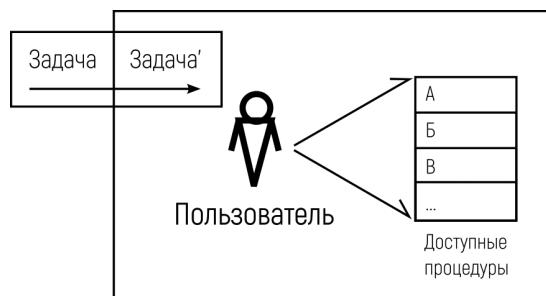


Рис. 4. Особенности работы пользователя в процедурной информационной системе

³ В связи с ростом популярности диалоговых интерфейсов, опирающихся на инструменты искусственного интеллекта, и частичного вытеснения ими графических интерфейсов, проблема вербализма в информационных системах может стать актуальной.

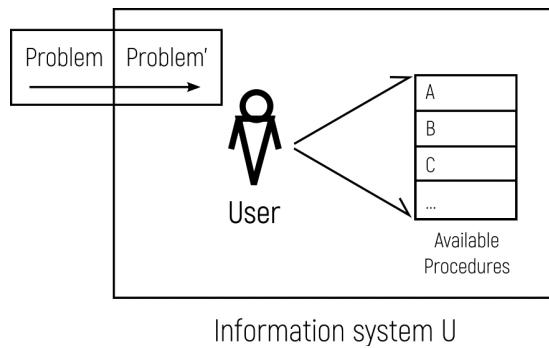


Fig. 4. The characteristics of user experience in a procedural information system

ностей системы. Например, работая в популярном текстовом редакторе, пользователь редактирует документ на уровне его *внешнего представления*, как будто документ создается непосредственно на бумаге. При этом зачастую пользователем упускается множество уникальных особенностей цифровых документов: разделение информации и формы ее представления, использование инструментов автоматизации, например нумерации библиографии, работа со словарями.

Ярким признаком знаковой натурализации в современных информационных системах с графическим интерфейсом стало появление особых знаков («иконок» или «виртуальных объектов»), напоминающих пользователю объекты реального мира, операции над которыми пользователь может выстраивать исходя не из имеющихся знаний об устройстве системы, а из соответствующих *овеществленных представлений* об этих объектах. На экране предстает «рабочий стол», на нем лежат «папки», в них хранятся «листочки-файлы», которые можно отправить в «облако» или в «корзину», — пользователь видит в этих действиях с информацией привычные или легко воображаемые для себя операции с реальными и воображаемыми объектами.

Натурализация действительно позволяет самостоятельно выполнять доступные операции и формировать из них простые предметные действия, видеть непосредственный результат своих шагов, трансформации объектов и т. п. Это хорошо работает в случае простых операций, которые должны привести к понятным и однозначным результатам, а также для самостоятельного индуктивного овладения доступными операциями в системе, что соответствует идеям конструкционизма. Неудивительно, что дизайнеры интерфейсов коммерческих продуктов, по сути своей процедурных систем с четким и ограниченным функционалом, ухватились за этот прием и осознанно формируют знаковую натурализацию у пользователей, что позволяет быстро и без лишних раздумий освоить и использовать продукт, ускорить действия пользователя, довести их до автоматизма (Krug, 2000).

С психологической точки зрения можно отметить важные ограничения процедурных систем, основанных на механизмах знаковой натурализации. Это отказ пользователя от ответственности за свои действия — плохо работать может все что угодно: неисправное оборудование, программы с ошибками или

неумелые программисты. Массовый переход к процедурным системам связан с коммерциализацией и превращением пользователя в *потребителя* — потребитель получает готовый продукт и не должен разбираться с проблемами, вызванными в том числе его собственными некорректными действиями.

Другое ограничение связано с потенциалом развития мышления пользователя. Работая с готовыми процедурами, шаблонами и мастерами («wizards»), пользователи не выходят на понимание ограничений системы, на деятельность моделирования или конструирования, которые характерны для инструментального использования систем программистами. Еще М. Вейтгеймер отмечал, насколько привычка действовать только последовательно, шаг за шагом, согласно выученному шаблону препятствует развитию мышления (Вейтгеймер, 1987). Можно обнаружить ограничения в обнаружении пользователями идеальных представлений и удержания этих представлений при использовании интерфейсов, характерных для современных процедурных систем (Федосеев, 2013).

Знаковая натурализация является принципиальным ограничением при обучении программированию. Неизменность такой системы, ее процедурный характер, невозможность переконфигурирования или смены используемой модели деятельности ограничивают возможности пользователя и начинающего программиста в выделении *идеального компьютера* (notional machine), т. е. вычислительной модели или идеального представления о программируемой системе (Sorva, 2013). Так, используемая в языке LOGO Сеймуром Пейпертом «черепашка» неизменна, язык управления ею, как и более современный язык Scratch, могут расширяться пользователем, но эти системы не предназначены для того, чтобы пользователь «заглядывал им под капот», восстанавливая заложенные в них принципы управления или модели, на которые системы опираются, тем более меняя внутренние правила работы этих систем. Именно выделение *идеального компьютера* (notional machine) при освоении способов программирования фиксируется большинством исследователей как ключевая педагогическая задача (Munasinghe, Bell, Robins, 2023; Papert, 1980; Sorva, 2013).

В.В. Давыдов, В.В. Рубцов и другие (Давыдов, Рубцов, 1990), проанализировав подход С. Пейпера, установили, что в педагогической системе этого

выдающегося исследователя основное осваиваемое учащимся содержание сводится к выполнению операций, заданных механизмом работы электронного устройства. В системе развивающего образования Давыдова это соответствует пооперационному контролю при освоении *учебных действий контроля*. Другие учебные действия, связанные с выделением исходного отношения изучаемой системы, действия моделирования, действия преобразования модели значительно слабее представлены в системе Пейперта. Более того, следуя подходу Ж. Пиаже, С. Пейперт отождествляет (реализуя принцип изоморфизма) операциональную форму осуществления действия и само ситуативное действие, имеющее определенный смысл для учащегося. Как утверждают авторы (Давыдов, Рубцов, 1990): «Если учесть взаимопревращаемость учебных действий и операций, то определить сколько-нибудь однозначно состав операций очень трудно».

Игрово-педагогический подход С. Пейперта критикуется и исследователями и разработчиками образовательных видеоигр, например Я. Богостом (Bogost, 2010). Учащийся, работая по системе С. Пейперта, принимает за данность представленное изображение предметов и набор операций в конкретной компьютерной системе. Учащийся в данной системе обучения не ставится в ситуацию построения знакового отношения и создания моделей существенных характеристик используемой системы вычислений в заданном языке программирования. А именно этот тип моделей может быть наиболее принципиальным и важным, поскольку он позволяет ответить учащемуся на вопросы, зачем он использует системы программируемых вычислений, как он вмешивается в регулирование автоматизированных физических процессов. То есть системы программируемых вычислений в компьютере могли бы стать средством организации деятельности учащегося.

Таким образом, при работе с информационными системами, в том числе при обучении программированию, необходимы инструменты и педагогические практики, направленные на преодоление знаковой натурализации и создающие ситуации, способствующие развитию мышления учащихся. Рассмотрим подробнее возможный поход, направленный на решение поставленной проблемы, в опоре на работы В.В. Давыдова, В.В. Рубцова и Ю.В. Громыко (Громыко, Просекин, 2024; Громыко, Рубцов, Марголис, 2020).

Преодоление знаковой натурализации

В работах отечественных авторов В.В. Давыдова, В.В. Рубцова, Ю.В. Громыко преодоление ограничений конструкционизма и знаковой натурализации связано с действиями моделирования учащегося. Компьютерное моделирование деятельности может создавать условия для поиска и выделения существенных характеристик объектов (Давыдов, Рубцов, 1990). Преодоление знаковой натурализации с помощью специальных действий и инструментов явля-

ется принципиальным шагом на пути формирования мышления ребенка (Устиловская, 2008). Для этого необходимо выйти за ограничения избыточной иллюстративной наглядности к выделению мысленных структур, встроенных в язык (Громыко, 2023).

В.В. Давыдовым была разработана образовательная система, которая также исключает получение учащимся знания «в готовой форме», на чем настаивал и С. Пейперт. Но, в отличие от конструкционизма Пейперта, основу системы Давыдова образует технология представления теоретического понятия в виде специфических предметных действий, позволяющих прослеживать условия происхождения знания (понятие числа, понятие фонемы). Для освоения понятия В.В. Давыдовым была разработана методическая система формирования учебной деятельности младшего школьника, состоящая из самостоятельных учебных действий учащегося: специфических предметных действий по выделению исходного отношения изучаемой системы, действий моделирования отношения, действий преобразования модели для изучения этого отношения, решения частных задач на основе освоения общего способа решения, действий контроля предыдущих действий и действий оценки освоения общего способа решения учебной задачи. Осуществляя данные учебные действия, младший школьник и осваивает содержание понятия.

Преодоление знаковой натурализации возможно через выделение существенных характеристик объекта и построение *собственных знаковых систем*, задающих его идеальное представление, в рамках деятельности человека-машинной системы. Этот процесс исследовался отечественными психологами в контексте формирования у школьников обобщенного способа действия через самостоятельные построения учащимися моделей в результате циклического движения от предметных действий к идеальным представлениям через введение новых символов, слов (Давыдов, Андронов, 1997). В экспериментах Л.В. Берцфай (Давыдов, 1996) и А.М. Медведева (Медведев, 2010) рассматривались процессы перехода от освоения отдельных операций к освоению обобщенных способов через формирование школьниками собственных знаковых представлений. При переносе этого подхода на работу в информационной системе учащиеся могут восстановить идеальную действительность ситуации управления, в рамках которой используется программируемая система, тем самым обнаружить ограничения используемых моделей или предложить собственные.

С точки зрения мыследеятельностной педагогики, теории и практики развивающего образования перед учащимся должна быть поставлена задача моделирования исходного отношения осваиваемой системы программируемых вычислений. В этом случае предметом такого моделирования становится отношение *идеального компьютера (notional machine)* к программируемой, регулируемой системе автоматизированных материальных процессов в разных практических контекстах. Именно эта форма моделирования позво-

лит учащемуся превратить *вычислительное мышление* (*computational thinking*) (Grover, Pea, 2013) в средство работы с системой деятельности, в которую включено компьютерное устройство. Подобное моделирование позволяет превратить знаки в целенаправленно создаваемые средства выражения понятого. Например, предметом работы в игре, обучающей программированию, должно стать не столько конструирование программ из готовых блоков или элементов языка программирования, сколько восстановление принципов, заложенных разработчиком в такой конструктор, обнаружение моделирующего отношения в данном языке программирования и его ограничений, а также выход за границы этих принципов при решении более сложных практических задач, в том числе построение новых более сложных моделей.

Такой подход также может найти опору на специальные инструменты, такие как среды *имитационного моделирования* (Жегалин, 2007), моделирующие агентные системы, предложенные в рамках *цифрокогнитивного подхода* (Громыко, 2023), а также системы искусственного интеллекта и *киберфизические системы*, обладающие большим потенциалом для обучения моделированию, поскольку связаны с решением задач с неопределенностью на стыке детерминированных цифровых систем и стохастической физической реальности. Работа с киберфизическими системами и лежащими за ними физическими, а не только цифровыми процессами обладает куда большим потенциалом для освоения деятельности моделирования. Киберфизические системы составляют основу для запуска новой метапредметной дисциплины деятельности киберфизики (Громыко, Просекин, 2024), которая в России уже развивается в рамках Национальной киберфизической платформы (Федосеев, 2023) на базе сети технологических кружков (Андрюшков, 2023). Работа в таких системах предполагает непосредственное участие школьников в построении модели целевой системы и только затем в создании программы (управлении процессом или его цифровой симуляции) в опоре на предложенную модель.

Говоря об исключительной важности предложенных еще в рамках конструкционизма принципов формирования учебной субъектности школьников и их самостоятельных учебных действий в органи-

зованных педагогом ситуациях, следует отметить большой потенциал игровых форм обучения (Федосеев и др., 2025; Bogost, 2010), в том числе при вовлечении участников в такие сложные формы деятельности, как исследование (Fedoseev, Vdovenko, 2014). Совмещение игровых, моделирующих и организующих рефлексивную коммуникацию и деятельность учащихся инструментов может стать важным шагом к реализации означенных в статье педагогических подходов.

Заключение

В статье была рассмотрена проблема знаковой натурализации в обучении программированию и при работе в информационных системах в целом, ставшая особенно актуальной вследствие массового перехода от традиционных методов обучения к конструкционизму, предложенному С. Пейпертом и реализованному как в современных программно-методических средах, таких как Scratch, так и в широко популярных информационных системах.

Рассмотренный в статье подход, направленный на преодоление знаковой натурализации и связанный с восстановлением исходного моделирующего отношения в системе и его преодолением в ходе решения сложных практических задач, дает возможность при реализации и внедрении в образовательный процесс выстроить альтернативную среду развития мышления учащихся при работе в цифровых средах и обучении программированию. Это позволит на практике осуществить идеи школы будущего, построенной на освоении учащимися способов мышления, коммуникации и действия в опоре на погружение их в конструкторскую, исследовательскую, проектную и другие ведущие культурные формы деятельности.

Для выявления психологических механизмов преодоления знаковой натурализации в современных цифровых системах и средах и создания условий для реализации соответствующей педагогической практики в младшей и, особенно, средней школе необходимо проведение дальнейших психолого-педагогических исследований, в том числе экспериментальных подтверждений отдельных означенных в статье подходов.

Список источников / References

1. Андрушков, А.А., Просекин, М.Ю., Старостинская, А.С., Федосеев, А.И. (2025). *Как стать технологическим кружком Национальной киберфизической платформы? Методическое пособие*. М.: Ассоциация участников технологических кружков. URL: <https://platform.kruzhok.org/toolkit> (дата обращения: 08.10.2025). Andryushkov, A.A., Prosekin, M.Y., Starostinskaya, A.S., Fedoseev, A.I. (2025). *How to Start a Technology Circle of the National Cyber-physical Platform. Methodical Manual*. Moscow. Kruzhok Association. (In Russ.). URL: <https://platform.kruzhok.org/toolkit> (viewed: 08.10.2025).
2. Вейтгеймер, М. (1987). *Продуктивное мышление*. М.: Прогресс.
3. Wertheimer, M. (2020). *Productive thinking*. Springer Nature.
3. Громыко, Ю.В. (2023). Культурно-историческая психология овладения деятельностью и альтернативы цифровизации. *Культурно-историческая психология*, 19(2), 27–40. <https://doi.org/10.17759/chp.2023190204>
3. Громыко, Ю.В. (2023). Cultural-Historical Psychology of Mastering Activity and Alternatives to Digitalization. *Cultural-Historical Psychology*, 19(2), 27–40. (In Russ.). <https://doi.org/10.17759/chp.2023190204>

4. Громыко, Ю.В., Просекин, М.Ю. (2024). Деятельностная когнитивистика в условиях цифровизации. *Вопросы философии*, 4, 45–52. <https://doi.org/10.21146/0042-8744-2024-4-42-52>
- Gromyko Y.V., Prosekin M.Y. (2024). Activity Cognitivistics under the Conditions of Digitalization. *Problems of Philosophy*, 4, 45–52. (In Russ.). <https://doi.org/10.21146/0042-8744-2024-4-42-52>
5. Громыко, Ю.В., Рубцов, В.В., Марголис, А.А. (2020). Школа как экосистема развивающихся детско-взрослых сообществ: деятельностный подход к проектированию школы будущего. *Культурно-историческая психология*, 16(1), 57–67. <https://doi.org/10.17759/chp.2020160106>
- Gromyko, Y.V., Rubtsov, V.V., Margolis, A.A. (2020). The School as Ecosystem of Developing Child-Adult Communities: Activity Approach to Designing the School of the Future. *Cultural-Historical Psychology*, 16(1), 57–67. (In Russ.). <https://doi.org/10.17759/chp.2020160106>
6. Давыдов, В.В. (1996). *Теория развивающего обучения*. М.: Интор.
- Davidov, V.V. (1996). *The concept of developmental teaching*. Moscow. Intor. (In Russ.).
7. Давыдов, В.В., Андронов, В.П. (1997). Психологические условия происхождения идеальных действий. *Психологическая наука и образование*, 2(3), Статья 3. URL: https://psyjournals.ru/journals/pse/archive/1997_n3/Davydov (дата обращения: 08.10.2025).
- Davidov, V.V. Andronov, V.P. (1997). The psychological conditions of the origin of ideal actions. *Psychological Science and Education*, 2(3), Article 3. (In Russ.). URL: https://psyjournals.ru/journals/pse/archive/1997_n3/Davydov (viewed: 08.10.2025).
8. Давыдов, В.В., Рубцов, В.В. (1990). Тенденции информатизации советского образования. *Советская педагогика*, 2, 50–55.
- Davidov, V.V., Rubtsov, V.V. (1990). Trends in the informatization of Soviet education. *Soviet Pedagogy*, 2, 50–55. (In Russ.).
9. Ершов, А.П. (1983). Программирование — вторая грамотность. *Квант*, 2, 2–7.
- Ershov, A.P. (1983). Programming is the Second Literacy. *Kvant*, 2, 2–7. (In Russ.).
10. Жегалин, В.А. (2007). *Школа мышления: Академия практического образования*. М.: ЭКАР.
- Zhegalin, V.A. (2007). *The School of Thinking. Academy of Practical Education*. Moscow. Ecar. (In Russ.).
11. Курячий, Г.В. (2004). *Операционная система UNIX: Курс лекций*. Учебное пособие. М.: ИНТУИТ.РУ.
- Kouryachy, G.V. (2004). *Operating System UNIX. A Course of Lectures. Tutorial*. Moscow. Intuit.Ru. (In Russ.).
12. Медведев, А.М. (2010). Генетико-моделирующий метод и диагностика новообразований учебной деятельности. *Психологическая наука и образование*, 15(4), 91–99.
- Medvedev, A.M. (2010). Genetic modeling method and diagnosis of new formations of educational activity. *Psychological Science and Education*, 15(4), 91–99. (In Russ.).
13. Овсянников, А.Ю., Дятлова, П.А., Федосеев, А.И. (2023). *Исследование технологического обеспечения кружков НТИ: робототехника*. М.: Ассоциация участников технологических кружков.
- Ovsyannikov, A.Y., Dyatlova, P.A., Fedoseev, A.I. (2023). *Research of the Technological and Methodological Support of the Circles of National Technology Initiative: Robotics*. Moscow. Kruzhok Association. (In Russ.).
14. Рубцов, В.В., Марголис, А.А., Пажитнов, А. (1987). Компьютер как средство учебного моделирования. *Информатика и образование*, 5, 8–13.
- Rubtsov, V.V. Margolis, A.A. Pazhitnov. A. (1987). Computer as a means of educational modeling. *Informatics and Education*, 5, 8–13. (In Russ.).
15. Устиловская, А.А. (2008). *Психологические механизмы преодоления знаковой натурализации идеального содержания геометрических понятий*: дис. ... канд. психол. наук. Психологический институт РАО. М.
- Ustilovskaya, A.A. (2008). *Psychological Mechanisms for Overcoming the Sign Naturalization of the Ideal Content of Geometric Concepts*: Dis. kand. psikhol. Nauk. Institute of Psychology of the Russian Academy of Education. Moscow. (In Russ.).
16. Федосеев, А.И. (2013). Диагностика ограничения развития мышления учащихся при работе в современных информационных системах на примере задачи классификации. *Информатика и образование*, 3, 22–32.
- Fedoseev, A.I. (2013). Study of the Children Thought Development Limitations in the Modern Information Systems Concerning the Case of a Classification Problem Solving. *Informatics and Education*, 3, 22–32. (In Russ.).
17. Федосеев, А.И. (2023). Концепция Национальной киберфизической платформы. В: *Доклады IV Всесоюзного Конгресса по сенсорике и экономике «Сенсорное слияние-2023»*. (с. 13–19). Санкт-Петербург.
- Fedoseev, A.I. (2023). The Concept of the National Cyber-physical Platform. In: *Proceedings of the IV Congress on Sensory Science and Economics «Sensors Fusion-2023»* (pp. 13–19). Saint-Peterburg. (In Russ.).
18. Федосеев, А.И., Андриушков, А.А., Горбунов, Г.Г., Зарипов, Н.А., Земцов, Д.И., Мумина, М.М., Пашкова, Д.В., Пигуз, А.А., Старостинская, А.С., Фахретдинов, А.Р., Яськов, И.О. (2025). *Полезные игры и польза игр*. М.: Издательство НИУ ВШЭ.
- Fedoseev, A.I., Andryushkov, A.A., Gorbunov, G.G., Zaripov, N.A., Zemtsov, D.I., Mumina, M.M., Pashkova, D.V., Piguz, A.A., Starostinskaya, A.S., Fakhretdinov, A.R., Yaskov, I.O. (2025). *Serious Games and Games for Good*. Moscow. HSE Publishing House. (In Russ.).
19. Beynon, M., (2017). Mindstorms revisited: Making new construals of Seymour Papert's legacy. In: *Educational Robotics in the Makers Era* (pp. 13–19). Springer International Publishing.
20. Bogost, I. (2010). *Persuasive games: The expressive power of videogames*. mit Press.
21. Fedoseev, A.I., Vdovenko, D. (2014). Playing Science: Role-Playing Games as a Way to Enter Scientific Activity. In: *Frontiers in Gaming Simulation* (pp. 3–12). <https://doi.org/10.1007/978-3-319-04954-0>
22. Grover, S., Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, 42(1), 38–43. <https://doi.org/10.3102/0013189X12463051>
23. Krug, S. (2000) *Don't make me think!: a common sense approach to Web usability*. Pearson Education India.
24. LaPrade, M., Lassiter, S. (2023). From LOGO programming to Fab Labs: the legacy of constructionist learning in K 12 education technology. *International Journal on Integrating Technology in Education*, 12(2), 1–18. <https://doi.org/10.5121/ijite.2023.12201s>
25. Munasinghe, B., Bell, T., Robins, A. (2023). Computational thinking and notional machines: The missing link. *ACM*

- Transactions on Computing Education*, 23(4), 1–27. <https://doi.org/10.1145/3627829>
26. Palfrey, J., Gasser, U. (2011). *Born digital: Understanding the first generation of digital natives*. ReadHowYouWant.com.
27. Papert, S. (1980). *Mindstorms. Children, Computers, and Powerful Ideas*. New York: Basic Books, Inc.
28. Pea, R., Kurland, M., Hawkins, J. (1985). LOGO and the Development of Thinking Skills. In: *Children and Microcomputers: Research on the Newest Medium* (pp. 193–317). Beverly Hills, Calif. : Sage Publications.
29. Resnick, M., Ocko, S., Papert, S. (1988). LEGO, Logo, and design. *Children's Environments Quarterly*, 5(4), 14–18.
30. Sorva, J. (2013). Notional machines and introductory programming education. *ACM Transactions on Computing Education*, 13(2), 1–31. <https://doi.org/10.1145/2490822>
31. Tenenberg, J. (2024). Notional machines: Retrieving background practices of perception and action. *ACM Transactions on Computing Education*. <https://doi.org/10.1145/3688390> (viewed: 08.10.2025).
32. Turkle, S., Papert, S. (1990). Epistemological pluralism: Styles and voices within the computer culture. *Signs: Journal of women in culture and society*, 16(1), 128–157. <https://doi.org/10.1086/494648>

Информация об авторе

Федосеев Алексей Игоревич, президент Ассоциации участников технологических кружков, Москва, Российская Федерация, ORCID: ORCID: <https://orcid.org/0000-0003-1652-6982>, e-mail: fedoseev@kruzhok.org

Information about the author

Alexey I. Fedoseev, President of Kruzhok Association, Moscow, Russian Federation, ORCID: <https://orcid.org/0000-0003-1652-6982>, e-mail: fedoseev@kruzhok.org

Конфликт интересов

Автор заявляет об отсутствии конфликта интересов.

Conflict of interest

The author declares no conflict of interest.

Декларация об этике

Исследование соответствует Кодексу этики научных публикаций. https://psyjournals.ru/info/code_of_ethics

Ethics statement

The study research complies with the Scientific Publications Ethic Policy. https://psyjournals.ru/en/info/code_of_ethics.

Поступила в редакцию 28.01.2025

Received 2025.01.08

Поступила после рецензирования 10.10.2025

Revised 2025.10.10

Принята к публикации 10.12.2025

Accepted 2025.12.10

Опубликована 29.12.2025

Published 2025.12.29