

ГИПОТЕЗА ЛИНГВИСТИЧЕСКОЙ ОТНОСИТЕЛЬНОСТИ (ГИПОТЕЗА СЕПИРА-УОРФА) В ПРОГРАММИРОВАНИИ

Попов А.А., студент

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Мозалевская Д.А. – канд. фил. наук, преподаватель

В данной работе исследуется влияние парадигм программирования на мышление разработчика и архитектуру цифровых систем через призму лингвистической гипотезы Сепира-Уорфа. Рассматривается язык программирования не просто как утилитарный инструмент, а как когнитивная среда, задающая онтологические законы. Выявляется антропологический аспект взаимодействия человека и программного кода.

Традиционно языковая система рассматривается исследователями не только как базовое средство коммуникации, но и как фундаментальный инструмент, который напрямую формирует структуру человеческого сознания и способы познания окружающего мира. Проблема взаимосвязи языка, мышления и объективной реальности является одной из ключевых в современной философии. В середине XX века американские исследователи-этнолингвисты Э. Сепир и Б. Ли Уорф, опираясь на глубокие исследования языков коренных народов Северной Америки (в частности, анализируя специфику нелинейного восприятия времени у племени хопи), сформулировали знаменитую гипотезу лингвистической относительности. Согласно данному подходу, грамматическая и семантическая структура естественного языка не просто пассивно отражает существующую реальность, но и активно конструирует ее в сознании говорящего субъекта. Язык выступает в роли своеобразной когнитивной матрицы, которая задает рамки мышления, детерминирует поведение и диктует способы категоризации повседневного опыта [1].

В современной когнитивной науке и философии для четкости методологических границ исследования принято выделять две основные версии гипотезы Сепира-Уорфа: «мягкую» и «строгую». «Мягкая» версия предполагает, что язык оказывает существенное влияние на направленность мышления, однако оставляет субъекту потенциальную возможность выйти за его концептуальные рамки. «Строгая» же версия утверждает, что язык абсолютно и бескомпромиссно определяет все когнитивные процессы: человек в принципе не способен помыслить, осознать или сформулировать то явление, для которого в его родной знаковой системе отсутствуют соответствующие лексические единицы или грамматические конструкции [2].

Долгое время концепция Сепира-Уорфа применялась в гуманитарных науках исключительно к естественным человеческим языкам. При этом «строгая» версия лингвистического детерминизма часто и вполне обоснованно подвергалась критике со стороны исследователей за излишнюю категоричность, поскольку естественная речь обладает гибкостью и позволяет преодолевать понятийные границы. Однако, как отмечает академик В. С. Стёпин, исследуя актуальные проблемы философии науки и техники, в современных условиях тотальной глобальной цифровизации фокус философского осмысления закономерно и неизбежно смещается на языки формальные и искусственные [2]. И именно в этой новой цифровой среде, при непосредственном переходе к языкам программирования, строгая версия гипотезы Сепира-Уорфа лишается некоторых противоречий и обретает более полное воплощение.

Если естественный человеческий язык служит преимущественно для фиксации и описания уже существующей, объективно данной физической реальности, то язык программирования выступает принципиально иным инструментом - средством её непосредственного, демиургического создания. Программист, оперируя синтаксическими конструкциями, не просто передает информацию, а буквально конструирует новые онтологические уровни бытия. Современный философ сознания Д. Чалмерс в своих исследованиях аргументированно доказывает, что виртуальные миры, спроектированные с помощью программного кода, обладают собственной, независимой онтологической полноценностью. Они являются подлинной, структурно сложной, а не просто иллюзорной или вторичной реальностью, поскольку в них присутствуют строгие причинно-следственные связи и реальные объекты взаимодействия [3]. В этой автономной реальности синтаксис и семантика выбранного языка программирования перестают быть просто правилами написания текста и трансформируются в фундаментальные «законы физики» для создаваемой цифровой среды. Именно они определяют границы возможного и невозможного внутри системы, задавая базовую механику взаимодействия виртуальных сущностей так же жестко, как гравитация или термодинамика определяют процессы в материальном мире.

Следовательно, осознанный или вынужденный выбор конкретного языка программирования диктует разработчику не просто алгоритм решения узкой утилитарной задачи, но и саму фундаментальную архитектуру его технического и творческого мышления. Философ Л. Витгенштейн в

своем фундаментальном труде «Логико-философский трактат», исследуя изоморфизм языка и реальности, вывел знаменитый постулат: «Границы моего языка означают границы моего мира» [4]. В контексте современной IT-индустрии этот глубокий философский тезис полностью лишен поэтической метафоричности и работает абсолютно буквально. То, что невозможно или крайне затруднительно выразить средствами конкретного языка программирования ввиду отсутствия в его ядре нужных встроенных абстракций или структур данных, становится невозможным ни помыслить технически, ни эффективно реализовать в рамках создаваемой этим языком программной системы. Разработчик оказывается в строгих когнитивных рамках доступного ему синтаксического инструментария. В этой парадигме программный компилятор выступает не просто как утилитарный транслятор кода в машинные команды, а как строгий эпистемологический фильтр – он становится абсолютной, непреодолимой границей объективно выразимого и, как следствие, границей самого конструируемого виртуального мира.

Наиболее ярко и наглядно действие принципа лингвистической относительности проявляется при глубоком анализе и столкновении различных базовых парадигм программирования. Каждая из таких парадигм предлагает разработчику не просто новый набор технических команд, а свою собственную, уникальную онтологию – полноценное философское учение о бытии конструируемого цифрового мира. В качестве фундаментального примера можно рассмотреть объектно-ориентированное программирование (ООП), реализуемое в таких популярных языках, как C++, Java или C#. Данная парадигма концептуально приучает специалиста воспринимать любую сложную систему и окружающий мир в целом как жестко структурированную иерархию взаимодействующих дискретных сущностей (объектов). Каждый такой объект наделяется автономным внутренним состоянием и заранее определенным набором поведенческих реакций. В рамках этой когнитивной модели на первый план в мышлении неизбежно выходят «существительные» и строгая таксономия (систематизация и классификация). Непрерывная действительность искусственно дробится на атомарные составляющие, а мир предстает как гигантский механистический конструктор – набор деталей и модулей. Подобный подход концептуально отсылает к классической аристотелевской логике и субстанциальной картине мира, где первична сама вещь, а не происходящий с ней процесс.

Диаметрально противоположную картину мира формирует декларативная парадигма функционального программирования, классическими представителями которой выступают языки Haskell, Lisp и F#. В рамках этой математически выверенной концепции фундаментальной основой цифрового бытия становятся не статичные дискретные объекты, а динамические процессы, выступающие в роли смысловых «глаголов». Реальность здесь воспринимается через призму чистых функций и непрерывных трансформаций данных, при которых принципиально не происходит изменения их исходного состояния. Это концептуально отсылает к, так называемой, процессной философии, где фокус внимания смещается с фиксации изолированного сущего на его непрерывное становление. Разработчик, мыслящий абстрактными категориями функционального языка, видит проектируемую систему не как пространственный «склад» автономных объектов, а как направленный, логически детерминированный поток вычислений. Внимание переносится с того, какими сущностями наполнена система, на то, как именно сквозь нее протекают и эволюционируют данные. Именно поэтому переход программиста от одной парадигмы к другой требует не просто механического заучивания нового программного синтаксиса. Он требует полноценной, зачастую сложной перестройки всего когнитивного аппарата – радикального изменения самого способа концептуализации проблемы. Этот интеллектуальный барьер служит блестящим практическим подтверждением гипотезы Сепира-Уорфа: чтобы эффективно созидать на концептуально новом языке, разработчик вынужден научиться совершенно иначе мыслить.

Особый научно-исследовательский интерес для социально-гуманитарных дисциплин представляет антропологический аспект длительного взаимодействия человека и программного кода. В этом контексте искусственный язык выступает не просто как внешняя рабочая среда, а как активный участник когнитивной трансформации субъекта. Выдающийся ученый в области теоретической информатики Э. Дейкстра неоднократно подчеркивал фундаментальную мысль: интеллектуальные инструменты, которые мы выбираем и используем, отнюдь не нейтральны – они оказывают глубокое, подспудное и зачастую необратимое влияние на наши базовые мыслительные привычки [5]. Постоянная, многочасовая аналитическая работа в предельно жестких рамках формальных, математически детерминированных логических систем неизбежно запускает процесс глубокой профессиональной когнитивной адаптации. Человеческое сознание, вынужденное регулярно подстраиваться под бескомпромиссные требования абсолютной однозначности, строгой типизации и алгоритмичности программной среды, начинает постепенно перестраивать собственные внутренние механизмы анализа информации и принятия решений.

Формируется специфический алгоритмический тип мышления. Программист начинает бессознательно переносить логику искусственного языка на повседневную жизнь, социальные взаимодействия и решение бытовых экзистенциальных проблем. Для наглядности можно привести несколько характерных примеров подобного когнитивного переноса. Во-первых, в сфере межличностной

коммуникации: специалист может подсознательно воспринимать диалог как выполнение программной функции, где на валидный «ввод» (логичный, структурированный аргумент) всегда должен следовать строго детерминированный «вывод» (ожидаемая рациональная реакция собеседника). Сталкиваясь с непредсказуемой человеческой эмоциональностью, такое мышление интерпретирует её как «необработанное исключение» (exception) или сбой системы. Во-вторых, в решении банальных повседневных задач часто проявляется склонность к гипероптимизации: например, подсознательное применение алгоритмов поиска кратчайшего пути при планировании маршрута по супермаркету или попытка жестко регламентировать спонтанные личные отношения. Человек, привыкший к строгой типизации, предсказуемости кода и поиску оптимальных паттернов, начинает ожидать аналогичной предсказуемости и рациональности от человеческого общества, которое по своей природе нелинейно и иррационально. Любая неэффективность реального мира начинает восприниматься как системный «баг», требующий немедленного рефакторинга. Это неизбежное столкновение строгой машинной логики с хаотичной человеческой природой зачастую становится причиной экзистенциальных кризисов и социальных фрустраций.

Таким образом, возникает феномен технологической козволюции. Изначально разработчик создает искусственный язык, чтобы формализовать свои команды и управлять вычислительной машиной. Однако со временем структура, парадигма и ограничения этого языка начинают форматировать сознание самого создателя. Применение гипотезы Сепира-Уорфа к сфере информационных технологий убедительно доказывает, что язык программирования выступает в роли мощного когнитивного фильтра и инструмента конструирования реальности. Осознание этой глубокой лингвистической и онтологической обусловленности является критически важным шагом для современного IT-специалиста, стремящегося сохранить творческую субъектность и широту мышления в условиях тотально алгоритмизированного мира.

Список использованных источников:

- 1 Уорф, Б. Л. *Отношение норм поведения и мышления к языку* [Электронный ресурс] / Б. Л. Уорф // *Культурология.Ру*. – Режим доступа: <https://culturolog.ru/content/view/252/9/>. – Дата доступа: 10.03.2026.
- 2 Сепир, Э. *Избранные труды по языкознанию и культурологии* / Э. Сепир. – М. : Прогресс, 1993. – 656 с.
- 3 Стёпин, В. С. *Философия науки и техники* [Электронный ресурс] / В. С. Стёпин // *Электронная библиотека e-Reading*. – Режим доступа: <https://www.e-reading.life/book.php?book=15814>. – Дата доступа: 10.03.2026.
- 4 Чалмерс, Д. *Реальность+: виртуальные миры и проблемы философии* [Электронный ресурс] / Д. Чалмерс. – Режим доступа: https://totenburg.org/chalmers-reality-plus-1_-_PREdmarket.pdf. – Дата доступа: 11.03.2026.
- 5 Витгенштейн, Л. *Логико-философский трактат* [Электронный ресурс] / Л. Витгенштейн // *Онлайн-библиотека Djvu*. – Режим доступа: <https://djvu.online/file/Rz032Zw3wOjRE>. – Дата доступа: 11.03.2026.
- 6 Макконнелл, С. *Совершенный код* [Электронный ресурс] / С. Макконнелл // *Онлайн-библиотека Djvu*. – Режим доступа: <https://djvu.online/file/12izBScOYDZZQ>. – Дата доступа: 11.03.2026.
- 7 Дейкстра, Э. *Дисциплина программирования* / Э. Дейкстра. – М. : Мир, 1978. – 275 с.