

Разработка кроссплатформенной учебной среды для обучения программированию школьников и студентов

А.А. Плеханов, Р.В. Кусмамбетов, В.В. Лаптев

Астраханский государственный технический университет, Астрахань, Россия

Аннотация – Обучение программированию сегодня – это не просто изучение синтаксиса языка, это в первую очередь формирование алгоритмического мышления. Современные профессиональные инструменты слишком сложны. Синтаксис английского языка создает лингвистический барьер, а доступность готовых решений в интернете провоцирует к простому бездумному копированию кода. Кроме того, в условиях перехода образовательных учреждений на отечественное программное обеспечение остро встает вопрос кроссплатформенности учебных сред. В этой статье рассматривается опыт внедрения и планируемой модификации специализированной учебной среды программирования *Semantic IDE*, разработанной в Астраханском государственном техническом университете. Рассказано о ключевых принципах, заложенных в основу среды, приведён анализ существующих аналогов и описаны основные технические аспекты реализации.

Ключевые слова – учебная среда программирования, кроссплатформенность, *Semantic IDE*, русскоязычный синтаксис, транслятор кода.

I. ВВЕДЕНИЕ

Информационные технологии не стоят на месте, каждый год появляются новые инструменты для разработки программного обеспечения. В то же время такие инструменты становятся сложными, их функционал становится перегруженным, как и языки программирования. Как следствие, обучение становится сложнее, и, согласно исследованию компании “Яков и Партнёры”, только половина студентов ИТ-специальностей становятся выпускниками [1].

II. ПРОБЛЕМЫ СОВРЕМЕННОГО НАЧАЛЬНОГО ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ

Погружение в программирование для начинающего обучающегося сопряжено с рядом проблем. Первая и самая очевидная – это дефицит квалифицированных преподавателей информатики [2], которые могут на достаточном уровне обучить основным базовым

концепциям программирования (ветвление, циклы, функции и т.п.), не говоря уже про сложные языки такие как C++, Java и Go.

Второй проблемой является языковой барьер. Для продуктивной работы с большинством современных языков программирования и, что важно, с сообщениями об ошибках в средах разработки, необходимо владеть английским на среднем уровне. Школьнику приходится одновременно учиться думать как программист и запоминать десятки иностранных ключевых слов. Это создает избыточную когнитивную нагрузку и отвлекает от главного – понимания логики построения алгоритма. Преподаватели Пензенского государственного университета отмечают, что обучение на родном языке делает программирование доступнее для тех, кто не владеет английским языком [3].

Третья проблема – это такие среды разработки, как Visual Studio Code или IDE от JetBrains – мощные IDE для опытных разработчиков. Для новичка же их интерфейс представляет собой лабиринт из кнопок, меню и настроек, в которых начинающему легко запутаться. Идеальная учебная среда должна быть простой, интуитивной и направленной на решение конкретных учебных задач, а не на поддержку всевозможных технологий.

Четвёртая проблема связана с популярными учебными языками, такими как Python или JavaScript [4]. Будучи нетипизированными, они прощают многие ошибки на этапе написания, но могут выдать неочевидные сбои при выполнении из-за несоответствия типов данных. Это формирует небрежное отношение к структуре данных. При последующем переходе на строго типизированные языки (C++, Java) студенты сталкиваются с резким скачком сложности, так как фундамент для понимания типов не был заложен [5].

Пятой, но не менее важной проблемой является доступность готовых решений. Если выданное задание требует выполнения на Python, то ученик с высокой вероятностью найдет его готовое решение в интернете. Отсюда обучение сводится к бездумному поиску и копированию, а не к самостоятельному размышлению и поиску ошибок. В этой ситуации требуется

уникальная среда, код для которой нельзя просто скопировать из сети, заставляя студента “включать” голову.

Именно совокупность этих проблем и определяет требования к современной учебной среде программирования, которую не удастся в полной мере удовлетворить существующими массовыми продуктами.

III. АНАЛИЗ СУЩЕСТВУЮЩИХ УЧЕБНЫХ СРЕД

Чтобы понять, какие инструменты уже есть на рынке и в чём заключаются их недостатки, мы провели сравнительный анализ нескольких популярных сред, используемых в образовании: Semantic IDE, КуМира, ПиктоМира и Scratch. Сравнение проводилось по ключевым для нашей задачи критериям: способность работать на разных операционных системах (кроссплатформенность), невозможность копирования кода, отсутствие решений в открытом доступе, наличие встроенных обучающих материалов и возможность показать код учебного языка на другом промышленном языке. Результаты анализа показали, что полностью подходящего решения нет (см. Табл. I).

ТАБЛИЦА I.
АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ

Критерий	Semantic IDE	КуМир	ПиктоМир	Scratch
Кроссплатформенность	-	+	+	+
Запрет на копирование кода	+	-	-	-
Отсутствие готовых решений в интернете	+	-	+	-
Есть обучение программированию	+	+	+	+
Перевод кода с учебного языка в промышленные языки программирования	+	-	-	-

Например, Scratch, КуМир и ПиктоМир работают на разных платформах, но в них легко скопировать код и, ключевой момент, у них отсутствует “Перевод кода с учебного языка в промышленные язык программирования”. А также для КуМир и Scratch можно найти готовые проекты и код на просторах Интернета. Среда Semantic IDE, разработанная в АГТУ (эксплуатируется с 2012 года), является самой подходящей разработкой. В ней нельзя скопировать код, нет готовых решений в сети для её уникального языка Slang, присутствует основная теория по темам, и в первой версии Semantic уже присутствовал перевод в другой язык программирования (Python, C, Pascal, Basic) [6]. Однако её главным недостатком стала привязка к операционной системе Windows. Во время

активного импортозамещения и перехода школ и вузов на отечественные ОС на базе Linux (такие как «Альт» или «РОСА») [7] среда, которая не поддерживает эти платформы, теряет значимость. Этот анализ показал, что необходим не поиск нового инструмента, а глубокая модернизация существующего – Semantic IDE, с целью устранения его ключевого технического ограничения и добавления конвертора, который будет переводить код на современные востребованные языки программирования.

IV. КОНЦЕПЦИЯ SEMANTIC IDE V2

Концепция новой версии, Semantic IDE v2 строится на трех основных идеях, призванных устранить старые недостатки и добавить новые возможности.

Первая идея – переход на кроссплатформенную веб-технологическую основу. В качестве платформы выбран фреймворк Electron, который позволяет создавать десктопные приложения с использованием веб-технологий (JavaScript/TypeScript, HTML, CSS) [8]. Это решение устраняет проблему платформенной зависимости. Одно и то же приложение будет работать под Windows, macOS и, что критически важно, под любыми дистрибутивами Linux, включая все отечественные.

Второй идеей является создание целостного учебного курса внутри среды. Среда будет содержать структурированный теоретический материал по темам, сразу после изучения которого ученик перейдет к выполнению практических заданий в том же интерфейсе, используя тот же редактор. Затем последуют более сложные лабораторные работы. Это позволит сконцентрироваться на предмете, минимизировав посторонние переключения контекста и бесполезный поиск информации.

Третья идея – развитие и усиление функции трансляции кода. Это уникальная особенность, унаследованная от первой версии. Планируется поддержка преобразования кода из учебного Slang в пять современных промышленных языков: C++, Java, Python, JavaScript и Go. Цель такого преобразования – наглядная демонстрация кода: слева – оригинальный код на русском Slang, справа – его эквивалент на выбранном языке программирования. Это позволяет ученику увидеть, что логика и алгоритм, которые он построил, универсальны. Он понимает, что, выучив основы на родном языке, он уже фактически умеет программировать, и ему остается лишь освоить новый синтаксис. Это снимает страх перед языками программирования с иностранной лексикой и делает последующий переход осознанным и плавным.

V. ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ: РОЛЬ АБСТРАКТНОГО СИНТАКСИЧЕСКОГО ДЕРЕВА

Для реализации описанных возможностей необходимо использовать абстрактное синтаксическое дерево.

Абстрактное синтаксическое дерево (англ. Abstract Syntax Tree (AST)) – это не просто техническая деталь, а сердцевина всей системы. Это древовидная модель, которая представляет структуру программы. Каждый элемент кода – оператор, переменная, вызов функции, – становится узлом в этом дереве с определенными связями. Ключевая особенность подхода в Semantic IDE состоит в том, что это дерево строится динамически, прямо во время набора кода пользователем (Рис. 1).

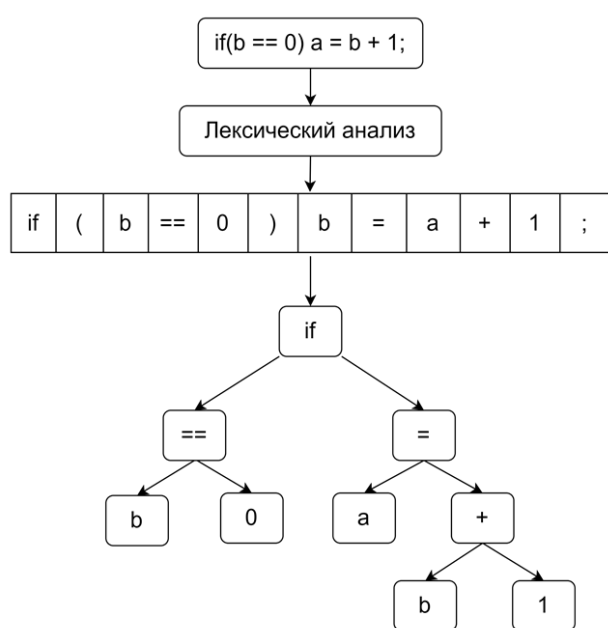


Рис. 1. Наглядное представление абстрактного синтаксического дерева

Это поможет решить несколько задач одновременно. Во-первых, на его основе работает мгновенная проверка кода: анализатор проходит по дереву и выявляет семантические ошибки (например, использование необъявленной переменной) ещё до попытки запуска.

Во-вторых, именно это дерево является объектом для манипуляций в семантическом редакторе. Когда студент добавляет блок «цикла» в коде, он фактически добавляет новый узел в AST. Представление кода на экране – это просто удобное для чтения отображение этого дерева.

В-третьих, AST служит идеальным промежуточным звеном для транслятора. Конвертору не нужно анализировать текстовый код Slang. Вместо этого он получает на вход уже готовое, проверенное и структурированное абстрактное синтаксическое дерево. Его задача сводится к обходу этого дерева и

генерации на его основе эквивалентного кода на целевом языке (C++, Python и т.д.). Такой подход делает систему трансляции надежной, предсказуемой и легко расширяемой для добавления новых целевых языков.

Можно провести аналогию с архитектурным паттерном «Модель-Представление-Контроллер» (MVC). В этой аналогии абстрактное синтаксическое дерево и есть Модель (Model) – центральная структура данных. Представление (View) – это интерфейс редактора, который отображает эту модель в виде текста (выбранного языка программирования). А Контроллер (Controller) – это все обработчики действий пользователя (ввод или изменение текста), которые изменяют AST, а те изменения, в свою очередь, автоматически обновляют View и запускают процессы проверки и трансляции.

VI. ВЫВОДЫ И ЗАКЛЮЧЕНИЕ

Разработка Semantic IDE v2 – это не создание проекта с нуля, а модернизация и развитие успешного отечественного образовательного продукта с более чем десятилетней историей.

Перенос среды на современную веб-технологическую платформу (Electron, TypeScript) гарантирует её жизнеспособность и доступность в любой образовательной организации, независимо от используемой операционной системы. А функция конвертации кода в пять промышленных языков построит прочный мост между учебной средой и реальным миром разработки, помогая студенту преодолеть синтаксический барьер и увидеть общность фундаментальных принципов программирования.

Таким образом, Semantic IDE v2 имеет все шансы стать современным, технологически актуальным и педагогически выверенным инструментом для массового обучения основам программирования и алгоритмического мышления в школах и учреждениях дополнительного образования.

ЛИТЕРАТУРА

- [1] Выпускники по ИТ-специальностям: от количества к качеству [Электронный ресурс] // Яков и Партнёры. — URL: <https://yakovpartners.ru/publications/from-quantity-to-quality/> (дата обращения: 17.12.2025)
- [2] Сколько учителей не хватает в российских школах на самом деле. [Электронный ресурс] URL: <https://vk.cc/cSyccn> (дата обращения: 18.12.2025)
- [3] Кодить по-русски: студентов научат программировать на кириллице [Электронный ресурс] // Известия. — URL: <https://iz.ru/1980660/anton-belyi/kodit-po-russki-studentov-nauchat-programmirovat-na-kirillice> (дата обращения: 16.12.2025)
- [4] Федеральная рабочая программа | Информатика. 7–9 классы (базовый уровень) [Электронный ресурс] URL: <https://vk.cc/cDP1ur> (дата обращения: 19.12.2025).
- [5] Барышева И.В., Козлов О.А., Малкина Е.В., Шестакова Н.В. Проблемы программирования в рамках школьного образования // Вестник Нижегородского университета им. Н. И. Лобачевского. Серия: Социальные науки. – 2023. – № 3(71). – [Электронный

ресурс] URL: <https://cyberleninka.ru/article/n/problem-programmirovaniya-v-ramkah-shkolnogo-obrazovaniya> (дата обращения: 19.12.2025).

- [6] Лаптев В.В., Грачев Д.А. Интегрированная среда для обучения программированию // Объектные системы – 2013: материалы VII Международной научно-практической конференции (Ростов-на-Дону, 10-12 мая 2013 г.) / Под общ. ред. П.П. Олейника. – Ростов-на-Дону: ШИ (Ф) ЮРГТУ (НПИ), 2013. – с. 17-24.
- [7] Топ-10 российских операционных систем 2025: перспективы и особенности [Электронный ресурс] // ABC-Labs. — URL: <https://www.abc-labs.ru/blog/top-10-rossiyskikh-operatsionnykh-sistem-2025-perspektivy-i-osobennosti/> (дата обращения: 16.12.2025)
- [8] Янг А., Мек Б., Кантелон М. Node.js в действии. 2-е изд. – СПб.: Питер, 2018. – с. 339–359.

Информация об авторах

Плеханов Алексей Алексеевич, бакалавр направления «Информатика и вычислительная техника» Астраханского государственного технического университета, г. Астрахань, Россия.

Кусмамбетов Рамиль Викторович, бакалавр направления «Информатика и вычислительная техника» Астраханского государственного технического университета, г. Астрахань, Россия, finn_nite_m@mail.ru

Лаптев Валерий Викторович, к.т.н., доцент кафедры "Автоматизированные системы обработки информации и управления" Астраханского государственного технического университета, г. Астрахань, Россия, WLaptew@yandex.ru

Development of a cross-platform learning environment for teaching programming to schoolchildren and students

A.A. Plekhanov, R.V. Kusmambetov, V.V. Laptev
Astrakhan State Technical University, Astrakhan, Russia

Abstract – Today learning to program is not just learning the syntax of a language, it is primarily the formation of an image of algorithmic thinking. Modern professional tools are too complex. The English syntax of languages creates a linguistic barrier, and the availability of ready-made solutions on the Internet provokes simple mindless copying of code. In addition, in the context of the transition of educational institutions to domestic software, the issue of cross-platform learning environments is becoming acute. This article examines the experience of implementing and planned modification of the specialized Semantic IDE programming environment developed at Astrakhan State Technical University. It describes the key principles underlying the environment, provides an analysis of existing analogues, and describes the main technical aspects of implementation.

Keywords – educational programming environment, cross-platform, Semantic IDE, Russian syntax, code translator.

References

- [1] Graduates in IT specialties: from quantity to quality [Electronic resource]. – URL: <https://yakovpartners.ru/publications/from-quantity-to-quality>. (date of access: 17.12.2025)
- [2] How many teachers are actually missing in Russian schools. [Electronic resource] URL: <https://vk.cc/cSyccn> (date of access: 18.12.2025)
- [3] Coding in Russian: students will be taught how to program in Cyrillic. [Electronic resource] – URL: <https://iz.ru/1980660/anton-belyi/kodit-po-russki-studentov-nauchat-programmirovat-na-kirillice> (date of access: 16.12.2025)
- [4] Federal Work Program | Computer Science. Grades 7-9 (basic level) [Electronic resource] URL: <https://vk.cc/cDP1up> (date of access: 19.12.2025).
- [5] Barysheva I.V., Kozlov O.A., Malkina E.V., Shestakova N.V. Problems of programming in school education // Bulletin of the Nizhny Novgorod University. Series: Social Sciences. – 2023. – No. 3 (71). – [Electronic resource] URL: <https://cyberleninka.ru/article/n/problem-programmirovaniya-v-ramkah-shkolnogo-obrazovaniya> (date of access: 19.12.2025).
- [6] Laptev V.V., Grachev D.A. Integrated environment for teaching programming. // Object systems – 2013: Proceedings of the VII International Scientific and Practical Conference (Rostov-on-Don, May 10-12, 2013) / Ed. P.P. Oleinik. – Rostov-on-Don: SRSPU (NPI), 2013. – pp. 17-24.
- [7] Top 10 Russian operating systems in 2025: Prospects and features [Electronic resource] // ABC-Labs. – URL: <https://www.abc-labs.ru/blog/top-10-rossiyskikh-operatsionnykh-sistem-2025-perspektivy-i-osobennosti/> (date of access: 16.12.2025)
- [8] Young, A., McHale, B., Cantelon, M.. Node.js in action. – 2nd ed. – St. Petersburg.: Piter, 2018. – Pp. 339–359.

Information about the authors

Alexey A. Plekhanov, bachelor of Science in Computer Science and Engineering, Astrakhan State Technical University, Astrakhan, Russia.

Ramil V. Kusmambetov, bachelor of Science in Computer Science and Engineering, Astrakhan State Technical University, Astrakhan, Russia, finn_nite_m@mail.ru.

Valery V. Laptev, cand. of tech, sci., asst. prof., Automation Systems of Information Processing and Control Dept., Astrakhan State Technical University, Astrakhan, Russia, WLaptew@yandex.ru.