

РАЗРАБОТКА УЧЕБНОГО АЛГОРИТМИЧЕСКОГО ЯЗЫКА ДЛЯ ОБУЧЕНИЯ ШКОЛЬНИКОВ ОСНОВАМ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

Е.С. Филиппов

Научный руководитель: А.И. Газейкина, к.п.н., доцент
ФГБОУ ВПО «УрГПУ», Екатеринбург, Россия

Аннотация

Статья посвящена исследованию применения алгоритмических языков в информатике: истории их развития, требованиям к учебным языкам и непосредственно к процессу разработки учебного алгоритмического языка. В статье представлен разработанный автором учебный алгоритмический язык «ПРАЯзык» и приведены примеры алгоритмов на этом языке.

Ключевые слова: *алгоритм, учебный алгоритмический язык, язык программирования, обучение основам алгоритмизации и программирования, требования к учебным алгоритмическим языкам.*

Одной из ведущих целей курса информатики является развитие алгоритмического стиля мышления у учащихся. Умение разработать оптимальный алгоритм для решения поставленной задачи требует от исполнителя определенных навыков.

Основным инструментом практического назначения при обучении учащихся основам алгоритмизации являются учебные алгоритмические языки.

Алгоритмический язык (АЯ), или Учебный алгоритмический язык (УАЯ) – формальный язык, используемый для записи, реализации и изучения алгоритмов. В общем случае алгоритмический язык реализует в себе все необходимые средства для написания программ. В этом заключается его основная цель. С педагогической точки зрения учебный алгоритмический язык является пропедевтикой языков программирования, т.е. должен обеспечить плавный переход от псевдокода к программированию.

История применения алгоритмических языков в школе начинается с 1985 года. В этом году было принято решение ввести курс информатики во все общеобразовательные школы. Алгоритмический язык с русским синтаксисом был введен в употребление академиком Андреем Петровичем Ершовым. В качестве основы для «безмашинного» курса информатики (в котором обучение школьников основам алгоритмизации и программированию занимало ведущее место) был выбран широко распространенный в 80-ые годы язык Алгол-60. Поскольку настоящий Алгол являлся сложным для его применения в школах, было выбрано очень небольшое подмножество этого языка, которое затем было русифицировано. В результате получился простой и понятный, но несуществующий язык.

Учебный алгоритмический язык активно применялся при обучении основам алгоритмизации. Впервые он был опубликован в учебнике «Основы информатики и вычислительной техники» в 1985 году. В первых школьных учебниках этот язык, впрочем, и не считался языком программирования. Он рассматривался как некоторая нотация для записи алгоритмов, причем сами алгоритмы могли быть совершенно неформальными, формулы можно было записывать как в математике (без всякой линейной записи) и пр. Считалось, что алгоритмический язык – это язык для записи алгоритмов, и что при работе на ЭВМ алгоритм надо будет перевести (переписать) на один из языков программирования, как-то Паскаль, Бейсик или Рапира. Таким образом, школьный алгоритмический язык противопоставлялся языкам программирования [6].

Разработка первого русского алгоритмического языка дала начало появлению редактора-компилятора «Е-практикум» («Е» – в честь Ершова) в 1985 г., системе программирования КуМир (Комплект учебных миров) в 1990 г. и многому другому.

Изучение алгоритмизации является отправной точкой при изучении программирования. Обучать конструированию алгоритмов следует методично и систематически, знакомя учащихся с задачами и приемами, типичными для программирования и не зависящими от конкретных приложений [1]. Одним из факторов, влияющих на успешность результата обучения, является выбор используемого учебного алгоритмического языка.

Важной особенностью учебного алгоритмического языка является и то, что он на основе метода “лингвизации” [4] поможет в дальнейшем освоить почти любой язык программирования. Этим обеспечивается непрерывность и “плавный переход” от алгоритмизации к программированию [5].

Учебный язык программирования, будучи первым для учащихся, должен отвечать определенным требованиям. К первой группе отнесем требования, которыми должен обладать каждый учебный язык программирования, чтобы считать его полноценным:

- язык должен быть **содержательно полноценным**, и в тоже время должен быть как можно более **простым и понятным** для пользователя;
- учебный язык программирования нельзя отрывать от материала обучения, поэтому его выбор должен основываться на программе обучения;
- язык должен отражать фундаментальные и наиболее важные концепции алгоритмов в очевидной, естественной и легко воспринимаемой форме [1];
- должен удовлетворять методологии структурного программирования;
- должен иметь все необходимые операции и конструкции для записи алгоритмов;

- обязательно наличие определенной типизации данных.

Ко второй группе отнесем характеристики, определяющие методические особенности учебного языка программирования.

Простота, ясность и удобство конструкций. Для обучения требуется достаточно простой язык, позволяющий с минимальными временными затратами решить задачу «знакомства с языком программирования» [2]. «Простота» подразумевает под собой наличие простого понятийного аппарата – простых, понятных, удобочитаемых конструкций для начинающего программиста. Процесс программирования – это сложная задача, поэтому наличие некоторой синтаксической свободы становится преимуществом, в отличие от жестких языков программирования.

В большинстве современных языков программирования можно выделить четыре основные части: описание структур данных (типизация), организация вычислений (вычисление), применение управляющих структур (операторы) и обеспечение средств расширения возможностей языка (подпрограммы).

Если говорить о приемах описания вычислений, то в них не должно быть “вседозволенности”. Некоторые синтаксические приемы могут привести к “неудобочитаемости” и снижают надежность программ.

Паскалеподобные языки требуют строгости и ясности в структуре программы, применяют управляющие конструкции и богатый набор структур данных. Эти принципы соответствуют философии структурного программирования [7]. Методология структурного программирования поможет учащимся быстро освоить и запомнить структуру программы.

Если рассматривать механизм подпрограмм как средство расширения возможностей языка, то мы получим следующую картину. Существует два типа подпрограмм (процедуры и функции), они могут быть вложенными, в них требуется строгое соответствие в количестве и типах формальных и фактических параметров, переменные могут быть описаны глобально в программе и локально в подпрограммах. Все эти отличия соответствуют требованиям структурного подхода и являются необходимыми для начального обучения программированию [5].

Структурное программирование значительно снижает сложность написания программы, процесс ее отладки, облегчает понимание исходного кода программы.

Явное объявление переменных и строгая типизация данных. Паскалеподобные языки на основе сильной типизации навязывают жесткий стиль [5]. Его суть заключается в следующем: тип каждой переменной задается в явном виде, и его невозможно изменить при дальнейшем выполнении программы. Данный стиль является наилучшим для начинающего программиста. Он увели-

чивает ясность, надежность программ, обеспечивает возможность легкого обнаружения ошибок [7].

Использование национальной лексики. Главное методическое достоинство языка – ясность для ученика решаемых задач, наглядность процесса работы в ходе выполнения программы. С точки зрения обучения, использование доступного для учащихся языка способствует росту скорости запоминания и воспроизведения материала. Поэтому синтаксис УАЯ должен основываться преимущественно на родном для учащегося языке. Большое внимание этой проблеме было уделено А.П. Ершовым в статье «Алгоритмический язык в школьном курсе основ информатики и вычислительной техники» [3].

Разработка общедоступного, оптимального и работоспособного алгоритмического языка является нетривиальной задачей. Уже более полувека, начиная с появления первого языка программирования высокого уровня (1954г., Фортран), множество исследователей занималось и занимается решением этой задачи. Так, результатом нашего исследования использования учебных языков в информатике стал алгоритмический язык «Праязык».

«Праязык» (аббревиатура от названия Простой Русский Алгоритмический Язык) – императивный, структурированный язык программирования, синтаксис которого основан на русском языке.

Особенностями языка являются строгая статическая типизация данных и наличие средств структурного (процедурного) программирования.

Ядром «Праязыка» является работа интерпретатора, лексический и синтаксический анализатор которого, а также синтаксическое дерево были разработаны при использовании генератора парсеров ANTLR. Данная программа позволяет конструировать трансляторы различного типа. Входными данными генератора явилась разработанная контекстно-свободная формальная грамматика языка.

Для УАЯ «Праязык» была создана среда программирования, основными функциями которой является запись, редактирование и исполнение алгоритмов. Исполнитель и среда программирования реализованы на языке программирования Java.

Приведем примеры некоторых алгоритмов, записанных на УАЯ «Праязык»:

Программа сортировка_массива

Цел и,й,в,ц; Цел массив[8] = [1,5,3,4,2,6,8,7];

Начало

Для и От 1 До 7 Выполнить

Для й От и+1 До 8 Выполнить

Если (массив[и]>массив[й]) Выполнить

Начало

в = массив[й];

массив[й] = массив[и];

массив[и] = в;

Конец

Для ц От 1 До 8 Выполнить

Печать(массив[ц]);

Конец

Программа алгоритм_Евклида

Функция алгоритмЕвклида(Цел а,б)

Начало

Пока (а != б) Выполнить

Если (а > б) Выполнить а = а - б;

Иначе Выполнить б = б - а;

Вернуть а;

Конец

Цел э1; Цел э2;

Начало

Ввести (э1);

Ввести (э2);

Печать (алгоритмЕвклида(э1,э2));

Конец

В процессе создания УАЯ «Праязык» были реализованы все этапы, необходимые для реализации алгоритмического языка. Была разработана контекстно-свободная формальная грамматика языка, разработан исполнитель и среда программирования. Грамматика языка соответствует всем требованиям, которым должен обладать алгоритмический язык.

Использование русского языка для записи переменных и служебных слов, применение названий команд, отражающих их непосредственную суть, являются механизмом, повышающим наглядность исходного кода.

УАЯ «Праязык» отвечает всем принципам философии структурного программирования: наличие базовых алгоритмических конструкций, возможность использования подпрограмм при реализации алгоритма, а также использования метода последовательного программирования «сверху вниз». Исходя из вышесказанного можно сделать вывод о возможности применения учебного алгоритмического языка «Праязыка» для обучения школьников основам алгоритмизации и структурному программированию. Его принципы помогут учащимся быстро освоить и запомнить структуру программы. А также, исходя из методологии структурного программирования, базовых алгоритмических конструкций достаточно, чтобы освоить логику алгоритмизации и осознать процесс про-

граммирования как некоторую дисциплину. Процесс разработки программ также облегчает наличие явной сильной статической типизации.

ЛИТЕРАТУРА

1. Вирт Н. Систематическое программирование. Введение. – М.: Мир, 1977. – С.183.
2. Глущенко Ю.В. Обоснование использования языка программирования в учебном процессе
3. Ершов А.П. Алгоритмический язык в школьном курсе основ информатики и вычислительной техники // Микропроцессорные средства и системы. 1985. №2. С. 48-51.
4. Ершов А.П. Комплексное развитие системного программного обеспечения: постановка проблемы. – Новосибирск, 1983. – С.38.- (Препр./ АН СССР Сиб. Отделение. ВЦ., №469)
5. Кобилев С.С. Образовательная информатика: Подход к обучению, выбор учебных языков и создание программных систем
6. Кушниренко А.Г., Лебедев Г.В. 12 лекций о том, для чего нужен школьный курс информатики и как его преподавать. Лаборатория Базовых Знаний, 2000.
7. Фьюер А Джехани Н Сравнение языков Си и Паскаль – В кн.: Языки программирования Ада, Си, Паскаль. Сравнение и оценка. – М.: Радио и связь, 1989, С.15-55

МЕТОДИКА ОБУЧЕНИЯ УЧАЩИХСЯ 5-6 КЛАССОВ ОСНОВАМ АЛГОРИТМИЗАЦИИ С ИСПОЛЬЗОВАНИЕМ СРЕДЫ KTURTLE

А.Е. Шитикова

Научный руководитель: А.И. Газейкина, к.п.н., доцент
ФГБОУ ВПО «УрГПУ», Екатеринбург, Россия

Аннотация

Статья посвящена обучению основам алгоритмизации учащихся 5-6-х классов: обсуждаются проблемы выбора языка программирования для школьников 10-12 лет, а также выбор методики обучения. В статье обосновано использование среды KTurtle и языка Лого, представлен разработанный автором учебный практикум и результаты его апробации.

Ключевые слова: алгоритм, алгоритмизация, исполнитель, язык программирования, обучение основам алгоритмизации и программирования, Лого, KTurtle.

На сегодняшний день курс информатики – вполне состоявшаяся учебная дисциплина. Потребность в непрерывном курсе информатики, проходящем че-