

Анализ языков программирования и выбор оптимального для изучения в школе

Бельченков Алексей Николаевич

*Приамурский государственный университет им. Шолом-Алейхема
студент*

Симдянкина Елена Константиновна

*Приамурский государственный университет им. Шолом-Алейхема
студент*

Лучанинов Дмитрий Васильевич

*Приамурский государственный университет им. Шолом-Алейхема
старший преподаватель кафедры информационных систем, математики и
правовой информатики*

Аннотация

В статье представлен обзор и анализ языков и систем программирования, изучаемых в школах. Приведены требования, предъявляемые к первому изучаемому языку программирования, рассмотрены особенности каждого языка и выбран оптимальный вариант изучения программирования в общеобразовательной школе.

Ключевые слова: язык программирования, программа, основа программирования, алгоритмическое мышление.

Analysis of programming languages and a selection of the best to study at school

Belchenkov Alexey Nikolaevich

*Sholom-Aleichem Priamursky State University
student*

Simdyankina Elena Konstantinovna

*Sholom-Aleichem Priamursky State University
student*

Luchaninov Dmitry Vasilyevich

*Sholom-Aleichem Priamursky State University
Senior lecturer of the Department of Information Systems, Mathematics and Legal Informatics*

Abstract

The article presents an overview and analysis of languages and programming systems studied in schools. The requirements for the first studied programming language are given, the features of each language are considered, and the optimal variant of studying programming in a comprehensive school is chosen.

Keywords: programming language, program, programming basis, algorithmic thinking.

Мнения об изучении языка программирования в школе, расходятся: некоторые считают, что программирование изучать не обязательно, а нужно просто поднимать компьютерную грамотность и осваивать основные офисные программы (как Западе), другие отмечают необходимость изучения операционных систем и нескольких языков программирования с различными парадигмами. Это крайние случаи, но зачастую бывает совсем непросто найти золотую середину. В первую очередь, необходимо определить цель:

- обучение школьников логическому и алгоритмическому мышлению;
- знакомство с компьютерами на бытовом уровне, чтобы школьники умели пользоваться Интернетом, электронной почтой, текстовым и графическим редакторами;
- закладка базовых знаний, необходимых для будущих инженеров, математиков, физиков и специалистов по информационным технологиям;
- знакомство с программированием как явлением, чтобы он представлял потенциал компьютерных систем.

Из общего количества школьников программистами станут единицы, но, несмотря на это, необходимо развивать алгоритмическое мышление и на примерах знакомиться с принципами построения современных компьютерных систем.

Каждый преподаватель предъявляет свои требования к учебному языку программирования. К таким требованиям может относиться, например: наличие высокоуровневых инструментов для обнаружения и недопущения ошибок и для отладки программ, простой, интуитивный синтаксис, наличие качественной документации с примерами, кроссплатформенность (наличие версий под различные платформы), наличие дружелюбной среды разработки и многое другое.

С одной стороны, обучение языкам программирования в школе не направлено на подготовку специалистов и базовые понятия алгоритмизации можно традиционно давать на Паскале. Для школьного уровня нет особой разницы между различными языками, база у всех идентична, а более углубленное изучение происходит уже в профильных учебных заведениях. Учащемуся достаточно знать базовые понятия бинарной логики, циклов, условий, ит.п. а их присутствие отмечается практически в любом языке программирования. Проблема заключается в том, что в школе важной задачей является знакомство с тем, что такое программирование и примерном представлении о том, как программировать.

С другой стороны, очень быстрое развитие IT-технологий способствует появлению новых платформ, языков программирования (Java, C#, Python и др.), технологий. Современному программисту приходится постоянно следить за текущими изменениями. Исходя из этого, программу обучения школьников программированию необходимо тщательно обдумывать, заглядывая в будущее [3].

Принципиальное значение имеет выбор языка и системы программирования, т.к. во многом от этого зависит методика освоения курса, последовательность и содержание предъявления учебного материала, система учебных заданий и, в общем, вся дальнейшая работа, нацеленная на овладение программированием для решения реальных практических задач на компьютере. Этот выбор определяет доступность восприятия, изучения и овладения учащимся приемами и методами программирования.

Программирование преимущественно подразумевает написание программы, а для ученика это означает то, с чем он работает на компьютере каждый день, например, текстовый процессор или Интернет-обозреватель. Т.е. для учащегося программой считается приложение с графическим оконно-кнопочным интерфейсом. Отсюда следует, что программа, составляемая им, должна быть такой же. Здесь идут в ход визуальные среды программирования, т.к. они предоставляют широкие возможности школьнику реализации такого приложения.

В настоящее время происходит переход от алгоритмических к объектно-ориентированным языкам с использованием визуальных систем программирования (Visual Basic, Delphi, Visual C++ и т.п.). Часто используются ранее Turbo Pascal, конечно, достаточно хорошо подходит для овладения принципами программирования, но его терминальный интерфейс достаточно тяжело воспринимается современными обучающимися.

Поскольку объектно-ориентированное программирование является основой всех нынешних языков программирования, то становится необходимым его изучение ещё в школе, чтобы учащиеся, приходя в высшие учебные заведения, уже обладали правильным представлением о программировании. Это утверждение, конечно, не бесспорно, так как для большинства учащихся их дальнейшее образование не связано с разработкой IT технологий.

Первый язык программирования должен обладать рядом требований:

- чёткое представление о том, что программа делает на каждом шаге;
- строгий формальный язык, без лишних «вольностей»;
- строгая типизированность, ибо смешение целых чисел, вещественных чисел и текстовых переменных приводит у начинающих программистов к неправильному представлению о методах хранения данных в памяти компьютера [4].

Чем больше учащиеся видят сообщений от компилятора об ошибках, и чем больше из этих сообщений они поймут, тем больше фундаментальных знаний о программировании они получают. Достаточно неплохой в этом

смысле язык – Паскаль, но т.к. он достаточно редко применяется на практике, специалисты по Паскалю оказываются мало востребованными на рынке труда.

Возможно, это и приводит к основной «неудовлетворенности» учащихся, решивших совершенствоваться в программировании. Для реальной работы им приходится самостоятельно изучать более популярные языки (Java, C/C++, PHP, Python и т.д.). Однако использование Си, как первого языка программирования имеет некоторые проблемы: в нём достаточно много пугающих конструкций даже для интересующихся программированием учащихся. Но, с другой стороны, совсем не обязательно показывать все глубины Си. С ним можно работать и на том же уровне, что с Паскалем, используя соответствующие конструкции.

Одним из лучших языков программирования для начинающих всегда считался и считается Pascal, потому что он был придуман специально для обучения основам программирования. Но он уже устарел и не применяется в коммерческих целях. Если начинать изучение программирования с Паскаля, то после основ станет необходимым изучение другого языка с другим синтаксисом и правилами. На Паскале, возможно, и стоит показать основные понятия, но обучать ему полностью не имеет смысла. Наиболее целесообразно рассматривать языки и технологии, которые активно используются в данный момент и будут использоваться в ближайшем будущем.

Паскаль считается более упрощенным, чем C, и он может научить «идеологически» подходить к программированию. Почему сначала нужно объявлять типы переменных, что операция присваивания значения и логическое равенство – это разные вещи. Когда делаешь элементарные ошибки в Паскале, то программа не будет работать. Он считается обучающей базой [2]. К альтернативам для Turbo Pascal, имеющим место в учебном процессе относятся:

1. Free Pascal – бесплатный компилятор Паскаля, который отвечает современным требованиям и обладает средой разработки достаточно схожей с Turbo Pascal. Это дает возможность использования большей части методической литературы, которая написана для Turbo Pascal. Работает на операционных системах Windows и Linux.

2. Pascal ABC.NET – удобная среда разработки, которая работает под Windows. Система была создана в качестве учебной среды программирования.

3. Lazarus – среда разработки, которая использует компилятор Free Pascal, но несмотря на это поддерживает разработку современных оконных приложений. Позиционируется как, совместимая с Delphi, но учебный материал, написанный для последней, применим не всегда. Интерфейс более сложен по сравнению с IDE Free Pascal. Рекомендуется использовать при достаточной квалификации преподавателя.

При выборе современных систем визуального проектирования (Lazarus, Delphi, Visual Studio ит.д.) в начале обучения программирования порождает

некоторые проблемы. Создавая проект, система автоматически генерирует большой объем кода, который связан с работой визуальных компонентов и не относится к решению поставленной задачи [2].

Сегодня достаточно удачным выбором, был бы Python. Он был создан более 20 лет назад и до сих пор используется и в обучении основам программирования, и в коммерческих целях. На рынке труда Python-программисты широко востребованы, и эта востребованность будет расти еще много лет, т.к. язык Python используют крупные корпорации в своих коммерческих проектах.

Python является полноценным языком программирования высокого уровня. Он обладает целым рядом парадигм: процедурная парадигма, модульная парадигма, функциональная парадигма, объектно-ориентированная парадигма.

Программу, написанную на Python, чаще всего не требуется изменять для запуска на другой операционной системе. Программы успешно работают как под Linux, так и под Windows.

Python является интерпретируемым языком. Это достаточно хорошо, потому что можно пробовать идеи «на ходу», не вводя ненужный код. Python характеризуется понятным синтаксисом, т.к. в нем мало вспомогательных элементов, а правила языка вынуждают программистов делать отступы. Python причисляется к языкам с динамической типизацией. В таком случае нагрузка на программиста меньше из-за того, что не требуется самостоятельно следить за типами данных.

В некоторых школах проводят интересные эксперименты по изучению программирования на языке Python. Учителя и школьники очень положительно отзываются об этом языке. Большой опыт преподавания информатики в школах с практикой на языке Python накоплен и за рубежом.

Несмотря на многие положительные моменты, связанные с языком Python, существует и ряд трудностей полного перехода обучения этому языку на уроках информатики:

- отсутствие, в достаточной мере, учебно-методического материала;
- все учебники по информатике написаны на примере Pascal
- успешное обучение школьников языку Python требует достаточной квалификации учителей в этом направлении[1].

Целью обучения программированию в школе является не изучение конкретного языка программирования, а развитие алгоритмического мышления, знакомство с различными стилями мышления и методами, которые применимы при решении различных задач.

Таким образом, среди рассмотренных языков наиболее подходящим для изучения в школах можно считать язык программирования Python. Но при существовании определенных трудностей для его изучения одним из вариантов является знакомство (без детального изучения) с несколькими языками программирования.

Библиографический список

1. Бобров А.Н. Проблемы выбора языка программирования в школьном курсе информатики // Молодой ученый. 2015. №24. С. 61-64. URL: <https://moluch.ru/archive/104/24471/> (дата обращения: 02.11.2018).
2. Лапчик М.П., Семакин И.Г., Хеннер Е.Г. Методика преподавания информатики: Учебное пособие для студентов педвузов. М.: Издательский центр «Академия», 2003. 624 с.
3. Захарова И.Г. Информационные технологии в образовании: Учебное пособие для студ. высш. учеб. Заведений. М.: Издательский центр «Академия», 2005. 192 с.
4. Авдошин С.М., Ахметсафина Р.З., Максименкова О.В. Информатика и ИКТ. Учебно-справочные материалы. М.: Просвещение, 2012. 252 с.