

В. В. Кащей,

Академия социального управления, г. Москва

РЕЛЕВАНТНОСТЬ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ В КУРСЕ ИНФОРМАТИКИ ОБЩЕОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЙ. ВЫБОР ЯЗЫКА ПРОГРАММИРОВАНИЯ

Аннотация

В статье анализируется роль алгоритмизации и программирования в курсе информатики. Рассматриваются критерии и предлагаются рекомендации по выбору языка программирования при изучении информатики.

Ключевые слова: школьная информатика, базовое образование по информатике, программирование.

Контактная информация

Кащей Владимир Васильевич, канд. пед. наук, доцент кафедры информационно-коммуникационных технологий Академии социального управления, г. Москва; *адрес:* 129281, г. Москва, Староватутинский проезд, д. 8; *телефон:* (495) 472-32-08, доб. 149; *e-mail:* wwwk54@mail.ru

V. V. Kaschei,

Academy of Public Administration, Moscow

THE RELEVANCE OF ALGORITHMIZATION AND PROGRAMMING IN THE INFORMATICS COURSE OF SECONDARY EDUCATIONAL ESTABLISHMENTS. THE CHOICE OF PROGRAMMING LANGUAGE

Abstract

The article analyzes the role of algorithmization and programming in the course of informatics. The criteria and recommendations for the choice of programming language in the study of informatics are considered.

Keywords: school informatics, basic informatics education, programming.

Необходимость и глубина изучения алгоритмизации и программирования в курсе информатики обсуждаются с момента введения данного курса в школьную программу. При этом высказываются самые различные мнения: от безусловной необходимости их изучения до полного исключения программирования из курса информатики. Вторая точка зрения обосновывается тем, что сегодня при использовании цифровых ресурсов нет необходимости уметь программировать в традиционном понимании этого термина. Наличие удобного интерфейса позволяет управлять ресурсами, не создавая программ. Попробуем выяснить, что дает изучение алгоритмизации и программирования.

Когда в школьный курс вводился новый предмет, авторы представляли его как «вторую грамотность». Введение в школьный курс дисциплины «Программирование» имело большое значение для обучения алгоритмическому образу мышления. Так же как изобретение языка и письменности обеспечило переход на качественно новый уровень развития человеческого мышления, овладение «второй грамотностью» развивает алгоритмическое мышление и вместе с ним абстрактное, логическое, критическое, т. е. научное мышление.

Первоначально предмет «Информатика» назывался «Основы программирования и вычислительной техники» и задумывался как обучение программированию. Целью было сделать программирование таким же понятным для всех, как и школьная математика. При этом предполагалось, что в будущем без знания основ программирования общаться с компьютером будет невозможно.

Программирование в то время было отраслью, в которой наиболее востребовано умение мыслить, творить. Здесь создавались новые знания, которые выражались в создании новых алгоритмов и кодировке их в программы. Проверенные алгоритмы записывались либо с помощью схем алгоритмов, либо на алгоритмическом языке высокого уровня типа Алгол. Сборники отобранных и проверенных алгоритмов периодически публиковались [1].

Разработанные алгоритмы реализуются в программах, которые не требуют материальной реализации, поэтому скорость реализации алгоритма в основном определяется только скоростью работы разработчика. Разработка алгоритма начинается с анализа условий задачи, выбора модели и метода численного решения. Все это обеспечивает выработку навыков аналитического, абстрактного, логического и алгоритмического мышления. В итоге обеспечивается выработка навыков критического мышления, которое является высшей формой мышления и относится к научному виду мышления. Программирование в принципе невозможно без творчества, поэтому научить программировать означает то же самое, что научить творчески мыслить. Выработка и развитие научного стиля мышления проявились в программировании в наиболее «чистом» виде.

Федеральные государственные образовательные стандарты определяют результаты изучения информатики. Согласно ФГОС среднего общего образования [6], изучение предметной области «Математика и информатика» должно обеспечить в том числе сформированность основ логического, алгоритмического и математического мышления; сформированность умений применять полученные знания при решении различных задач.

Есть две точки зрения на определение понятия «алгоритмическое мышление». Первая определяет алгоритмическое мышление как систему

навыков, ориентированную на заранее установленные правила, общепринятую последовательность действий, необходимых для решения типовых задач. Вторая рассматривает алгоритмическое мышление более широко. Рассмотрим более подробно эту вторую точку зрения.

Становление человеческого мышления происходило в процессе решения жизненно важных для существования человека задач. При решении некоей задачи необходимо было проанализировать сложившуюся ситуацию и либо подобрать уже существующий и опробованный алгоритм действий, либо создать новый на базе существующих путем их преобразования для использования в изменившейся ситуации.

Чем больше набор алгоритмов и чем лучше они систематизированы, тем лучше происходит адаптация объекта в системе. Необходимость анализа ситуации породила абстрактное мышление (выработка понятий), классификацию и систематизацию алгоритмов (выделение базовых конструкций) и синтез нового алгоритма на основе имеющихся абстрактных понятий и набора уже использованных алгоритмов. В этом смысле можно определять алгоритмическое мышление как систему мыслительных приемов, направленных на решение задач. Можно выделить две стадии разработки алгоритма: поиск и выделение уже известного (ранее опробованного) алгоритма и создание нового. При таком определении алгоритмического мышления мы не ограничиваемся разработкой, записью и кодированием алгоритма в программу, а рассматриваем весь процесс решения задачи, начиная с постановки задачи (анализа условий) и заканчивая получением результата и дальнейшим использованием разработанного алгоритма и программного продукта с учетом возможностей его дальнейшей модификации.

Изучение программирования способствует решению одной из главных задач образования — развитию научного (критического) мышления, без которого невозможно объективное понимание картины окружающего мира. Именно поэтому обучение алгоритмизации и программированию наиболее способствует формированию навыков такого мышления. Из этого следует, что алгоритмизация и программирование остаются одним из главных разделов информатики, и если мы хотим обучить и воспитать полноценного гражданина информационного общества, нам необходимо обеспечить их изучение.

Характерно, что представители коммерческих структур, заинтересованные в получении готовых специалистов, тоже выступают за обучение информатике в школе с большей теоретической составляющей и формированием алгоритмического мышления. Выступая на Десятой открытой Всероссийской конференции «Преподавание информационных технологий в Российской Федерации», Б. Г. Нуралиев, директор фирмы «1С», отметил: «Сегодня очевидно, что обучение современным технологиям нужно начинать не в вузе, а гораздо раньше. Необходимо правильное позиционирование школьной дисциплины “Информатика”, преподавание которой должно быть нацелено на освоение, прежде всего, ее фундаментальной составляющей, на то, чтобы привить учащимся необходимое современному человеку алгоритмическое мышление (полезное, независимо от будущей специальности), на сохранение преемственности между курсами информатики для разных ступеней образования» [5].

Президент и генеральный директор компании АБВУС. Г. Андреев, выступая на той же конференции, сказал: «Только человек, обладающий алгоритмическим мышлением, способен создавать что-либо новое, оригинальное, уникальное. Те же, кто имеет навыки пользователя, пусть и уверенного, могут лишь потреблять уже созданное кем-то. Поэтому, если мы действительно хотим выйти в мировые лидеры в сфере ИТ, нужно весь процесс обучения со школьной скамьи выстраивать с соответствующих позиций» [5].

При обучении программированию и выработке алгоритмического мышления возникает вопрос выбора языка высокого уровня, на примере которого проводится обучение. Этот вопрос постоянно обсуждается сообществом учителей информатики и заинтересованными лицами.

Определим требования, которым должен удовлетворять этот язык:

- Соответствие языка парадигме структурного программирования.
- Простота и компактность синтаксиса. Синтаксис должен быть «очевиден»: зная логику построения языка, как правило, можно «угадать» синтаксис операторов и описаний. Компактность синтаксиса обеспечивает простоту изучения языка.
- Строгость языка. Все объекты должны быть описаны и «приписаны» к соответствующим типам. Преобразования типов могут выполняться очень редко специальными функциями преобразования типов.
- Наличие интегрированной среды разработки и отладки программы с хорошей встроенной справочной системой и возможностями отладки программы.
- Высокая надежность программы за счет контроля использования типов данных и объектов программы, как на этапе трансляции программы, так и при ее выполнении.

Большое количество языков программирования, разработанных в последнее время (Java, C/C++, PHP, Python и др.), допускают более компактное и простое описание исполнения алгоритма. Однако выбор их как первого языка для обучения спорен. Их синтаксис достаточно сложен и, главное, неочевиден, есть много непонятных обучающимся конструкций. Процесс программирования облегчается за счет использования большого количества библиотек, но их использование в то же время затрудняет понимание в процессе изучения языка и методов программирования, поскольку действия, которые выполняются библиотечной программой, «невидны», и поэтому непонятно, что и, главное, как делает компьютер.

Например, на языке Python легко поменять местами значения переменных a и b . Это делается одним оператором $a, b = b, a$. Но у обучающегося не будет понимания того, как это делается на самом деле, он не узнает, что для такого обмена необходима еще одна переменная, в которой будет временно храниться промежуточное значение одной из переменных.

Еще сильнее можно запутаться при изучении структур данных в Python. Рассмотрим пример такой структуры, как списки. Списки похожи на массивы, но могут одновременно содержать данные любого типа, т. е. в одном массиве могут находиться числовые, строковые данные и данные других типов. Попробуйте это объ-

яснить обучаемым, когда большинству из них непросто понять, что такое обычный одномерный массив.

Python, конечно, мощный язык, но для начального изучения он все-таки слишком сложен. Аналогично обстоит дело и с некоторыми другими недавно разработанными языками программирования. Энтузиазм по выбору «нового и перспективного» языка для первичного обучения, проявляемый некоторыми учителями, вероятно, основан на том, что они успели забыть, как сложно разобраться с некоторыми базовыми понятиями программирования, когда встречаешься с ними впервые.

Представляется целесообразным, что при наличии времени необходимо дать возможность обучаемым основательно изучить основы программирования на базовом языке, включив сюда объектно-ориентированное программирование и базовые структуры данных (например, файлы, множества, записи, динамические переменные и структуры, отображаемые на их основе, — списки, таблицы, графы и т. д.). Лучше хорошо изучить один язык, чем бегло ознакомиться с несколькими. В этом случае при освоении нового языка можно по аналогии, в сравнении понять средства этого языка.

Изучив объектно-ориентированное программирование, можно перейти к созданию программ на основе визуальной объектно-ориентированной RAD* среды программирования, которая является развитием изученного перед этим языка. Например, Delphi или Lazarus для языка Pascal.

При последовательном изучении нескольких языков обучаемые начинают путать конструкции этих языков. В 1980-х годах в некоторых московских вузах студенты на протяжении двух—четырёх семестров изучали два—четыре языка программирования, и у них постоянно возникали проблемы с синтаксисом и структурами данных изученных языков. Это неоднократно отмечалось преподавателями вузов, проводившими занятия по программированию.

При изучении информатики не так важно, какой именно язык учить: основа всех языков идентична, а надстройка над основой изучается позже, в профильных учебных заведениях. Достаточно разобраться в способе реализации языком базового набора понятий математической логики, условий, циклов и т. п., имеющихся

практически в любом языке программирования. Задача школы состоит в ознакомлении с основами программирования и формировании приблизительного представления о методах его реализации.

Разработка программ предполагает до начала написания программы разработку алгоритма. Именно на этом этапе должны решаться основные проблемы разработки и реализации решения поставленной задачи. Написание текста программы — это перевод алгоритма на язык программирования, который является достаточно формальной задачей. Это еще раз подтверждает мысль, что выбор языка не так уж важен, если мы правильно разработаем алгоритм. При этом при выборе формы записи алгоритма рекомендуется учесть особенности языка, на который будет переведен алгоритм. В этом случае предпочтительнее язык с простым синтаксисом, максимально приближенным к естественному английскому языку.

Подведем итоги. Изучение алгоритмизации и программирования, несмотря на кажущуюся узкой область применения программирования, помогает развивать научное мышление. Поэтому изучение алгоритмизации и программирования является одной из ключевых составляющих обучения информатике. При выборе языка для обучения программированию необходимо руководствоваться пригодностью его для обучения, в первую очередь, основам программирования.

При правильной разработке и записи алгоритма выбор языка программирования не так уж и критичен, так как хорошо разработанный и правильно записанный алгоритм не сложно перевести на любой язык программирования.

Список использованных источников

1. Библиотека алгоритмов 16-50б: справочное пособие. М.: Сов. радио, 1975.
2. Бобров А. Н. Проблемы выбора языка программирования в школьном курсе информатики // Молодой ученый. 2015. № 24. <https://moluch.ru/archive/104/24471/>
3. Ворожцов А. В. Языки программирования в школе // Потенциал. 2016. № 5. https://ru.wikibooks.org/wiki/Языки_программирования_в_школе
4. Мелешко В. Алгоритмическое мышление. Без него невозможно создать что-либо новое // Учительская газета. № 22 от 31.05.2011. <http://www.ug.ru/archive/40532>
5. Преподавание информационных технологий в Российской Федерации: Материалы Десятой открытой Всероссийской конференции (16–18 мая 2012 года). М.: МГУ имени М. В. Ломоносова, 2012.
6. Федеральный государственный образовательный стандарт среднего общего образования (в ред. Приказа Минобрнауки России от 29.12.2014 № 1645). http://минобрнауки.рф/документы/2365/файл/736/12.05.17-Приказ_413.pdf

* RAD (от *англ.* rapid application development — быстрая разработка приложений) — концепция создания средств разработки программных продуктов, уделяющая особое внимание скорости и удобству программирования, созданию технологического процесса, позволяющего программисту максимально быстро создавать компьютерные программы.