В январе 2000 года в Санкт-Петербургском государственном электротехническом университете состоялся семинар «Заочная школа современного программирования» для преподавателей информатики школ Ленинградской области. Семинар был организован Малым областным университетом совместно с кабинетом информатики ЛОИРО и журналом «Компьютерные инструменты в образовании». Наряду с обсуждением методических и организационных вопросов функционирования Заочной школы участники семинара получили возможность из «первых рук» узнать мнение специалистов о роли информатики и программирования в образовании.

Перед участниками семинара выступил автор многочисленных работ по теории программирования чл.-корр. РАН С.С. Лавров, профессор математико-механического факультета, автор недавно вышедшей книги «Дискретный анализ» и ряда других книг по исследованию операций И.В. Романовский, декан факультета информатики и вычислительной техники СПбГЭТУ профессор И.В. Герасимов, профессор кафедры математического обеспечения ЭВМ А.В. Смольянинов, преподаватель ОИ и ВТ лицея № 8 города Сосновый Бор Н.Н. Паньгина.

Редакция журнала считает, что эти выступления затрагивают актуальные проблемы постановки курса информатики в школе и помещает авторские обработки выступлений в виде журнальных статей.

Лавров Святослав Сергеевич

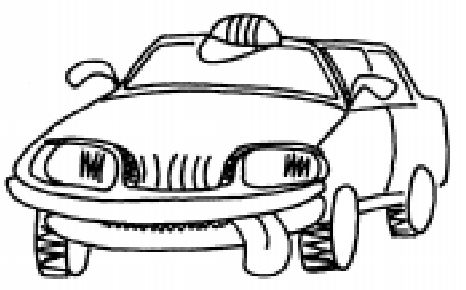
ЧТО ИЗУЧАЕТ ИНФОРМАТИКА?

В информатике, с которой я связан уже 45 лет, мне нравится все, кроме ее названия. Я считаю важным отличать содержание от формы – информацию от данных. Данные – это коды самого разнообразного вида. С ними имеет дело машина, компьютер. Человек же, получая от машины некий набор данных, придает ему определенный смысл. Только человеку данные несут информацию.

Поэтому я предпочитаю не связывать нашу деятельность с понятием информации и называть ее программированием. А его я определяю как искусство заставить машину решить стоящую перед человеком задачу. Для меня в этом определении важно каждое слово. Поговорим о них по порядку.

Человеческий фактор заключается в том, что именно человек – главный субъект всякой программистской деятельности. Машина, как бы ни была важна ее роль, лишь помогает человеку в меру его, человека, способности заставить ее это сделать. Заставить, поскольку она, как всякий бездумный и бездушный материал, тупо сопротивляется человеческому воздействию. И хорошо, что тупо. Фантасты описали множество жутких картин, связанных с проявлением у машин собственного разума и собственной воли.

Заметили ли вы, что термин «искусственный интеллект» почти вышел из употребления? Случается, что для какого-то вида деятельности, требующей от человека немалых умственных усилий, создают-



Фантасты описали множество жутких картин, связанных с проявлением у машин собственного разума и собственной воли...

ся эффективные программы. Но этот вид тут же выводится из сферы искусственного интеллекта. Начинают говорить просто о создании программы для игры в шахматы, или перевода с одного языка на другой, или автоматизации управления каким-либо процессом. Но назвать это повышением уровня машинного разума язык ни у кого не поворачивается.

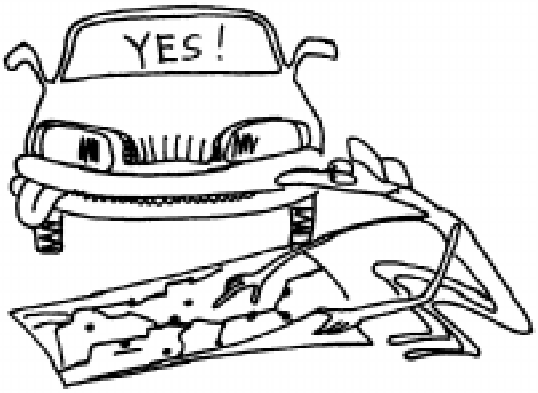
Мы заговорили о способностях человека. Все люди разные, различны их способности, их интересы и знания. В одних и тех же данных для разных людей заключена разная информация. На роль машины в его жизни каждый человек смотрит по-своему.

Что еще важнее – различны типы личности и интеллекта человека. Я никогда не стану говорить, что такой-то тип выше, а другой ниже, или слабее, или еще в чем-то уступает первому. Люди разные, поэтому и машины должны служить им по-разному. Я против агитации за какую-то универсальную информационную культуру. Программировать каждый должен по-своему. Но мне предстоит коснуться всех сторон программистской деятельности.

В какой-то мере программирование является фундаментальной наукой. Но, пожалуй, лишь в той мере, в какой оно связано с математикой. Вспомним такие математические понятия, как формальные языки, логика, множества, доказательства или вычисления в самом общем их понимании. Все они имеют для программирования важнейшее значение. В той или иной степени они должны быть освоены каждым, кто хочет докопаться в программировании до основ. И каждым, кто хочет научить программированию других.

Более специфичны, но и более существенны в практической деятельности понятия классов и структур данных. Это – тексты, числа, символьные выражения, изображения и пр. и построенные из них объекты. Можно придать какой-либо из этих структур, например, таблицам, особую роль. Программировать в какой-то одной сфере деятельности станет, может быть, и легче. Но будет ли легче сменить профессию и характер своих занятий? А ведь в наши дни эта перспектива весьма животрепещуща.

Задачи – это то, что возникает перед человеком на каждом шагу. Решая задачи, люди веками обходились без компьютеров. Но я еще помню, что в годы моего детства и самолеты, и радиоприемники были людям в диковинку, по улицам ездили извозчики и ломовики, а метрополитена не было и в столице. Но метро и самолеты – это всего лишь средства передвижения. Радио и телевидение – средства развлечения и «массовой информации» (мне трудно удержаться от кавычек). Решение задач – это нечто более существенное. Это и выделяет программирование из ряда других профессиональных знаний. В вузах всех специальностей это поняли давно. В школах общего профиля еще продолжают спорить и сомневаться. Я – человек пристрастный и от школы далекий, а вам предлагаю: «Думайте сами, решайте сами – иметь или не иметь».



Самая мощная машина с самым современным программным обеспечением никакую задачу сама не решит...

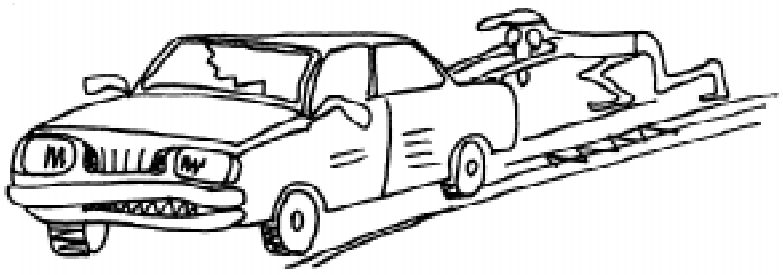
Машины бывают абстрактные и реальные...

Самая мощная машина с самым современным программным обеспечением никакую задачу сама не решит. Человек должен растолковать ей, применяясь к ее возможностям, чего он от нее хочет. Чем меньше ее возможности, тем подробнее должны быть составленные для машины предписания: алгоритмы и программы. Между прочим, самое современное программное обеспечение – это вовсе не самое совершенное. Часто в погоне за простотой использования создатели программного обеспечения выхолащивают из него некоторые принципиально важные возможности. А спохватившись, начинают вносить их заново, втискивая в уже весьма ограниченные рамки.

Поэтому знание хотя бы одного универсального языка программирования, пусть даже учебного языка, представляется мне важным. Пусть не свободное владение этим языком, а лишь освоение его основных понятий, их роли в языке, их взаимосвязей. Настаивать не могу, так как только что говорил, что из-за разнообразия типов личности некоторые люди никогда не станут пользоваться такими внутренне чуждыми им языками, не вдохновятся заложенными в них возможностями.

Алгоритм (или, по Маркову, алгорифм) и программа – это почти что синонимы. Разницу между этими понятиями можно объяснить так. Машины бывают абстрактные и реальные. Все слышали о машинах Тьюринга, многие – о машинах Поста, о нормальных алгорифмах Маркова, о других существующих лишь на бумаге средствах преобразования данных по определенным предписаниям. В этом случае и следует говорить об алгоритмах. Для реальных машин пишутся программы. Не настаиваю на этой терминологии, которая к тому же несколько принижает роль моего излюбленного термина – программирование. Но терминология складывается независимо от нашей воли. Своими волевыми усилиями мы вносим в нее невообразимый хаос, но постепенно все утрясается. Правда, каждому кажется, что утрясается не наилучшим образом.

Возможности машин велики, но не беспредельны. С ограниченностью возможностей реальных машин сталкивался почти каждый. То оперативной памяти недостает, то файл на диск не влезает, то программа работает катастрофически медленно, то погрешности вычислений начинают превышать сами вычисленные значения. Впрочем, стоп – приближенный характер всех реально выполняемых вычислений – это проблема соответствующей теории. Там уже и термины подобрались: трудно решаемые задачи, их временная и пространственная сложность, неустойчивость численного метода, некорректно поставленная задача и много чего еще.



С ограниченностью возможностей реальных машин сталкивался почти каждый...

Ограниченность возможностей абстрактных машин имеет иную природу. Есть классы задач, для решения которых составить алгоритм в принципе невозможно. Здесь можно воспользоваться такой аналогией. Правда, справедливо сказано, что всякая аналогия хромает. В печи топливо сгорает постепенно по мере его подбрасывания. Если же развести костер на торфянике или на лесной подстилке, то загорание нового материала контролю не поддается. Так и в алгоритмически неразрешимых задачах никакая организация процесса не способна справиться с появлением все новых вариантов его продолжения. Программисты называют это зацикливанием. По указанной причине зацикливание в любой программе решения неразрешимой задачи – вещь принципиально неустранимая.

Правда, в теории вычислимости популярен еще и так называемый диагональный метод. С его помощью любой вопрос, имеющий в зависимости от варианта исходных данных ответ «да» или «нет», можно превратить в неразрешимый, опираясь на предложенный вариант решения. Но этот метод я бы отнес к тем, с помощью которых изобретательные люди находят законный способ обойти любой закон. Я вижу здесь еще одно проявление человеческого фактора в программировании.

И, наконец, почему я называю программирование искусством, а не технологией? Технологиями называются методы безотказного и надежного решения задач, возникающих в определенной отрасли промышленности. Технологии существуют и в искусстве: подготовка холста, организация сценического движения, постановка голоса, версификация и т.п. Владение ими необходимо художнику. Но создание подлинного произведения искусства, решение именно художественной задачи они не обеспечивают. Сходно положение и в науке – никакой запас знаний не поможет при решении по настоящему трудной задачи. Должно наступить озарение. Помогает во многом случайная находка или интуиция – подспудное, неосознанное знание.

То же самое мы видим и в программировании. Разумеется, речь идет о роли компьютеров в решении нешаблонных задач. Сейчас компьютеры распространились так широко, что массовое их применение связано именно с решением массовых, однотипных задач. Здесь отрицать роль технологий было бы бессмысленно. Но я пытался заглянуть в душу программирования. Часть того, что я там увидел, я и предложил вашему вниманию. Спасибо.

Замечания после семинара.

Представим себе специалиста: геолога, филолога, биолога, занимающегося вычислениями разве что эпизодически. Его программистская культура достаточно высока, если он умеет:

– воспользоваться текстовым редактором, чтобы написать текст своей очередной публикации или письмо коллеге;

– обратиться с запросом к базе данных, где хранятся материалы по его специальности, и разобраться в полученных сведениях;

– добраться до калькулятора, имеющегося в используемой им оболочке, чтобы выполнить потребовавшиеся ему вычисления;

– разобраться в «подсказках», выдаваемых каждым из упомянутых программных средств.

Возможно, этот список можно слегка расширить.

Его программистская культура достаточно высока...

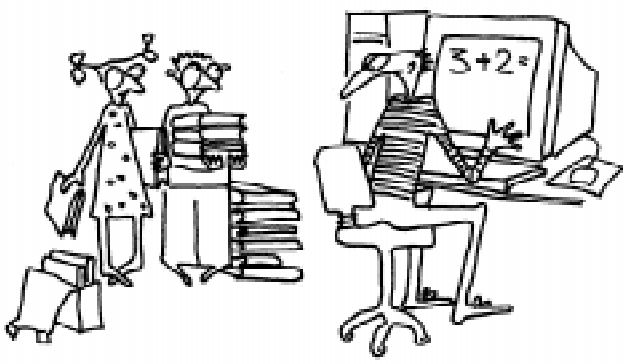
Программистская культура инженера, рассчитывающего орбиты межпланетных космических аппаратов или создающего новые программные оболочки, должна быть совершенно иной.

Уровень фундаментальности информатики за все последние годы и даже десятилетия скорее снижается, чем повышается, по причине массового распространения компьютеров и снижения среднего уровня требований к программистской культуре пользователей.

* * *

Пусть первокласснику задана задача: «У Саши в руках три книги, у Маши – две. Сколько книг у них вместе?» (именно в такой формулировке, даже если она не слишком пригодна для столь юного ученика). Пусть также в столбце примеров, заданных ему для решения, есть строчка: 3+2=. Эта строчка – в чистом виде пример задания на обработку данных. Первая задача скорее относится к сфере информатики, как я согласен принимать этот термин.

Ученик должен, прежде всего, понять, что у Маши, тоже в руках, а не в ранце, имеются две именно книги, что в этом контексте «они» – это Саша с Машей, а не те ребята во дворе школы, несущие пачки книг, за которыми он заинтересованно наблюдает, что «них» и «они» – это одно и то же местоимение (хотя он еще и слов «склонение местоимений» никогда не слышал), что спрашивается про общее количество книг у них в руках, а не дома на полках. Короче говоря, для решения задачи он должен



У Саши в руках три книги, у Маши – две....

вспомнить и использовать много сведений, приобретенных им на занятиях другими предметами, а то и вовсе почерпнутых из жизни.

Пусть все это происходит на уроке математики. Полезно, чтобы учитель понимал, что обе ситуации к математике имеют очень косвенное отношение. Я готов признать, что это понимание не поможет ему лучше вести свои уроки. Но это – элемент его информационной культуры.

* * *

Здесь говорилось, что человек, поступающий в вуз, отвечая на вопрос, заданный ему на вступительном экзамене по информатике, должен суметь следовать схеме, изложенной в учебнике («В каком учебнике?» – был общий вздох аудитории). Однако, если его ответ и не следовал ни одной из схем, содержащихся в многочисленных учебниках, но поступающий сумел убедительно отстоять свое мнение, то его надо немедленно зачислять в вуз, не обращая внимания на результаты других экзаменов (может быть – страшно сказать – даже по русскому языку).

Лавров Святослав Сергеевич,
доктор технических наук,
профессор,
чл.-корр. РАН.