

Некоторые вопросы и заблуждения по поводу программистов и программирования

А. В. Мельникова

**Эксперт в области разработки и внедрения
комплексных информационно-аналитических автоматизированных систем
(а также мама ученицы группы с углубленным изучением программирования)**

Цель статьи – рассмотрение часто встречающихся проблем, с которыми сталкиваются люди в процессе создания программного обеспечения как в своей профессиональной деятельности, так и при обучении информационным технологиям.

Речь в данном случае идет не о конкретных технических проблемах в работе с той или иной программной платформой, а об общих проблемах, препятствующих достижению качественных результатов в установленные сроки, что есть суть каждой профессии.

Статья ориентирована на людей, которые уже сделали свой выбор в пользу физико-математического профиля. В процессе обучения приходится сталкиваться со сложностями и даже разочарованиями, но у всех проблем есть решения. Для успешного решения проблемы необходимо в первую очередь понимание причин ее возникновения. Если причина определена неверно, то и выбранный путь решения не поможет. Здесь приводятся распространенные заблуждения, которые часто выдвигаются в качестве причин отсутствия успеха, анализ этих причин и некоторые подходы к преодолению препятствий на пути к цели.

1. Программированием могут заниматься только люди, обладающие большой усидчивостью.

Общее заблуждение состоит в том, что программирование требует от человека высокой способности заниматься рутинным занятием на протяжении долгого времени. Многие думают, что с этим занятием могут справиться только очень усидчивые люди, у которых к тому же нет в жизни других интересов.

В действительности именно программисты являются теми людьми, которые не любят тратить свое время на скучную однообразную работу. Это необычайно творческие люди, которые именно по причине своего неприятия рутины придумали программирование как таковое.

Таким людям очень дорого время и они предпочитают творческую работу однообразному труду. Поэтому они постоянно придумывают, как рутинные ручные методы заменить автоматическими и освободить свое время для другой интересной деятельности. Для этого, собственно, и придумывались компьютеры.

Конечно, обучение программированию требует настойчивости в достижении целей, особенно пока не освоены азы, без которых нельзя перейти к полностью творческой работе. Но это относится к программированию в той же степени, как и к любому другому обучению, или занятию: математика, музыка, спорт, искусство. Это верно и для любой серьезной работы. Выдающиеся ученые могут работать почти круглосуточно много лет подряд прежде чем сделают великое открытие – но им это нравится, это им интересно. Для них работа и хобби – одно и то же. Но чтобы достичь такого уровня, нужно сначала научиться.

2. Уроки программирования нравятся только тем, у кого хорошо и быстро получается выполнять задания.

Человеку нравится делать то, что у него хорошо получается. Но, чтобы стало хорошо получаться, нужно учиться, а для этого занятие должно человеку нравиться, иначе он учиться не станет. С чего же начать?

Если человек смог поставить себе цель, и ему действительно захотелось ее достичь, то он понимает, что для этого придется потрудиться. Это не будет в тягость, если он понимает, для чего именно нужен этот труд, и если видит, как труд приближает его к достижению цели. Он почувствует, каких знаний ему не хватает, и будет стремиться их получить. А урок и учитель – это ближайшая возможность получить недостающие знания. Для тех, кто именно так относится к урокам программирования, эти уроки интересны, и именно у этих людей результаты будут лучше. Именно ориентация на достижение конкретной цели отличает занятия проектным программированием от всех остальных уроков, где цели ставятся не учеником, а определяются стандартной программой обучения¹.

На практике, конечно, масштабы целей зависят от уровня знаний и навыков. Можно придумать и написать простую компьютерную игру, а можно создать автоматизированную систему для оптимизации расписания авиарейсов крупного аэропорта, где одновременно производят взлет и посадку десятки самолетов. Программирование позволяет сделать и то, и другое.

3. Человеку не нужно учиться программированию, потому что это лишь ремесло, и, будучи физиком, он сам сможет этому научиться.

Теоретически человек может научиться самостоятельно абсолютно всему, так как в наше время доступно множество ресурсов для получения знаний. Но важна также скорость получения этих знаний, а также своевременное их применение. Самая высокая скорость получения знаний достигается у людей на стадии вузовского и особенно школьного обучения, так как в этот период, как правило, они освобождены от необходимости зарабатывать себе на жизнь и кормить семью. К тому же они имеют возможность получения консультаций у своих наставников без существенных финансовых затрат.

У людей, которые не используют эти возможности активно, накопление знаний пойдет с отставанием, и они всегда будут проигрывать тем, кто успел больше за тот же срок. Чтобы в будущем получить интересную работу, придется вступить в конкуренцию с множеством других таких желающих, а для этого нужны конкурентные преимущества. Чем их будет больше, тем выше будут шансы.

В ряде вузов (МФТИ, МГУ и др.) сейчас открыты учебные центры и курсы при крупных успешных компаниях, куда берут студентов 1-2 курсов для дополнительного обучения и последующего возможного устройства на хорошо оплачиваемую интересную работу. Однако набор в такие центры производится по конкурсу, в который включены достаточно серьезные задания по программированию, значительно превышающие как уровень ЕГЭ, так и уровень обучения в самих вузах. Попадают в такие центры студенты, уже умеющие программировать, а это означает, что они должны научиться этому заранее, до поступления в вуз.

Навыки программирования – это серьезное конкурентное преимущество при выборе подавляющего большинства профессий в современном мире.

4. Надо ли учиться серьезно, или достаточно понимания самых общих теоретических принципов плюс умение написать простейшую программу в пять строчек?

Как и в любом деле, отличие между глубоким и поверхностным подходом состоит в том, что поверхностное изучение предмета дает человеку скромную сумму знаний, которая может и не пригодиться в будущей профессии, а глубокое изучение дает навыки, применимые для решения более широкого круга задач. Программирование дополнительно развивает мышление, аналитические способности, навыки самоорга-

¹ Именно по такой методике достижения цели и построен курс обучения в группах с углубленным изучением программирования, которые ведет преподаватель И.Р. Дединский (см. <http://ded32.net.ru/news/2011-04-03-58>).

низации и учит работать на результат. Конечно, есть и другие способы развития этих качеств, но если применять их одновременно, то результаты будут выше. Тогда зачем себя лишать такой возможности?

5. Человеку не нужно учиться программированию, если он не собирается быть программистом.

В современном мире, если человек не является производителем программного обеспечения, то он все равно является его потребителем. Почти все отрасли современной экономики работают с применением сложнейших программных комплексов. Если не понимать технологий такой работы, то сложно добиться успехов в любой из ее областей. Понимание этих технологий и владение ими дает большое конкурентное преимущество человеку независимо от того, в какой области он хочет себя проявить.

В качестве примера можно привести финансовую сферу. Так, раньше в банках работали в основном бухгалтеры, а сейчас, при переводе клиентов на дистанционное обслуживание через Интернет и мобильных операторов, в банках преобладают специалисты по информационным технологиям. А «люди в окошках» все менее и менее востребованы. На практике оказывается, что программист гораздо быстрее обучается управлению финансовыми потоками, чем экономист – программированию. Ту же картину мы видим и в промышленном секторе, где роботы постепенно заменяют рабочих, и во многих других областях. Знание информационных технологий становится в той же степени необходимым навыком для современного человека, как знание иностранных языков. Никто же не говорит, что не будет учить английский потому, что не собирается быть переводчиком!

6. Всегда ли нужна ли программисту четко поставленная задача?

И в школах, и в информационно-технологических компаниях часто можно услышать жалобы программистов на то, что учитель (в будущем – заказчик) недостаточно четко поставил задачу, и поэтому человек не смог ее выполнить так, чтобы заказчик был доволен результатом.

При этом есть и другой тип людей, которые, которые как раз очень рады возможности самостоятельно принимать решения о том, как выполнить задачу, и не любят навязанных решений.

В программировании обеим группам найдется место. Первые могут стать кодировщиками (программистами-исполнителями), а вторые – системными архитекторами. Кодировщику нужна четко поставленная задача, а архитектору достаточно понимать цель. Поэтому системный архитектор ставит задачи, а кодировщик лишь исполняет. И те, и другие весьма востребованы и быстро найдут себе работу. Но уровень квалификации и зарплаты будут отличаться в несколько раз. Выбор каждый делает сам. И сейчас, в процессе обучения, самое время его сделать.

7. Интересно создавать программу, но не интересно искать ошибки, особенно если в программе тысячи строк и сотни переменных. На творческую работу не остается времени из-за долгой отладки.

Залог успеха при создании программы в качестве ее проектирования (не путать с кодированием). Чем меньше уделяется времени этому этапу, тем больше потом времени уходит на поиск и исправление ошибок. А ведь проектирование и является самой креативной частью работы из всех. Это как раз работа архитектора. Пропуская ее, человек сам себе создает скучную работу на последующих этапах. Если сравнить эту область со строительством, то легко представить, какого качества получится здание, если перед его возведением не будет четко определенного проекта. Ломать и перестраивать дом по несколько раз гораздо сложнее, чем переписывать программу. Поэтому никому даже в голову не приходит поручить строителям начинать класть кирпичи прежде, чем архитектор завершит проектирование. Программирование в этом смысле ничем не отличается от строительства. Если лишить себя творческой работы, связанной с продумыванием архитектуры программы, то на последующих этапах будет и сложно, и скучно. Да и в любой другой области человеческой деятельности это правило действует абсолютно так же.

8. Не очень-то радостно идти на урок (а в будущем – на работу), если не получилось добиться работоспособности программы. Это подрывает настрой и мотивацию.

В отличие от коротких задач, разработка программы является проектом, который требует не только постановки цели, но и планирования, оценки трудоемкости и контроля промежуточных результатов. Если не

отнестись к этой деятельности, как к проекту, то работа не будет выполнена в срок и с нужным уровнем качества. Это отличает проекты по программированию от задач по большинству других предметов, где выполнение задания состоит в написании упражнения, решения примера или чтении параграфа из учебника, которое можно выполнить гарантированно за конкретное, и как правило небольшое, время.

Если до сдачи задания есть, скажем, три дня, то лучше начать работу именно в первый день. Это позволит сразу же выявить сложности, препятствующие достижению цели, и сформулировать вопросы, ответы на которые позволят решить проблемы (лучше письменно, чтобы не забыть). Часто эти вопросы можно задать сразу по электронной почте или другому средству коммуникации (скайп, TeamSpeak, ICQ) своему руководителю или учителю. Второй день можно использовать для получения ответов в личной беседе – с руководителем или коллегой, учителем, другом или в сети. На третий день можно завершить работу или развить ее. Планирование в четыре-пять дней с запасными (резервными) днями в конце периода даст дополнительный запас по времени и придаст уверенность. Если же начать работать в последний-предпоследний день, то трудности возникнут обязательно. Программирование позволит научиться не только писать программы, но и управлять проектами и своим временем, что также является очень востребованным навыком для специалиста любой отрасли. Не научившись выполнять в сроки маленькие проекты на три дня, в будущем не получится управлять большими проектами на месяц или год. Навык управления своим временем дает возможность не только вовремя справиться с задачами, но и позволяет человеку чувствовать уверенность в себе и устойчивость в жизни, что сильно повышает настроение и интерес к тому делу, которым человек занимается.

9. Какой практический результат я получу, если буду заниматься программированием?

Есть области, которые дают человеку возможность заниматься очень творческой работой, но при этом не позволяют найти мгновенное практическое применение результатам труда. Так часто бывает в фундаментальной науке. Например, открытие в математике обычно не может использоваться людьми непосредственно, а может использоваться лишь другими науками, которые позволят найти ему практическое применение. И наоборот, есть работа, в которой сразу виден результат, но мала творческая составляющая – например, уборка снега. Программирование – это уникальная область, в которой люди могут заниматься творчеством 100% времени и создавать свой продукт исключительно умственным трудом, как в математике, и в то же время сразу могут увидеть практические результаты своего труда. Это редкая и удачная комбинация теории и практики.