

Профессиональный путь программиста

Компьютеры и Интернет широко вошли во все сферы жизнедеятельности современного человека. Информационные технологии (ИТ, от англ. information technology, IT) – быстро развивающаяся отрасль мировой экономики. Огромное количество людей так или иначе работают в этой сфере. Это инженеры и рабочие, обеспечивающие проектирование и производство систем, компьютеров, средств цифровой связи и другой электроники (так называемого «железа»).

Любой современный компьютер или мобильный телефон по своей структуре является настолько сложным устройством, что не может функционировать самостоятельно, как, скажем, утюг или двигатель внутреннего сгорания. **Необходима программа**, которая бы всем этим руководила.

Программный код, который управляет компьютерами и другими устройствами, создают программисты. Этот код не является чем-то материальным, как электроника, – напротив, подобно книге, созданной писателем, **продукт деятельности программиста является абстрактным** (идеальным), не существующим в реальном мире. Программы «рождаются» в сознании их создателей (инженеров-программистов) и далее «живут» в памяти компьютеров, коими, собственно, и управляют.

Программа, с которой работает пользователь (программа высокого уровня), управляет не самим компьютером, а другой программой (системного уровня), которая в свою очередь может управлять какой-то третьей (низкого уровня). В конечном счете, эта иерархическая цепочка завершится в самом низу на какой-то конкретной аппаратной схеме (железе), состоящей из процессора, памяти, устройств ввода-вывода и других элементов.

Такие промежуточные программы называются драйверами (от англ. drive – управлять), а программный код, который находится в самом низу – контроллером (от англ. control – контролировать, в данном случае конкретную часть электроники компьютера). Программы самого высокого уровня называют прикладными, потому что они выполняют какую-то конкретную задачу предметной области и непосредственно взаимодействуют с пользователем (человеком).

На каждом уровне такой иерархии могут существовать разные программы, написанные на разных языках программирования, как правило, разными программистами. В создании любого крупного программного комплекса задействован не один программист, а целый коллектив. Часто над одной и той же программой работают сразу несколько специалистов.

Как же так получается, что компьютеры способны понимать программы, написанные программистами? Что для этого нужно и как вообще можно стать программистом?

Путь, который может пройти программист как профессионал, можно условно разделить на **четыре уровня**:

- Пользовательский уровень.
- Знание основ программирования.
- Углубленное изучение языка программирования; работа в реальных коммерческих проектах.
- Работа в команде программистов над сложными системами.

Давайте рассмотрим особенности каждого из этих четырех этапов.

Пользовательский уровень

В третьем тысячелетии основы компьютерной грамотности и использования сети Интернет стали такими же атрибутами современного человека, как умение писать и читать. Даже гуманитарии, которые далеки от точных наук, используют компьютер как один из основных инструментов для работы с информацией. Программисты не являются исключением. Они не обязаны знать все программы в мире и уметь ими пользоваться. Это то же самое, что пытаться прочитать все написанные на сегодня книги.

С другой стороны, хороший программист никогда не останавливается на каком-то узком наборе стандартных программ. Обладая широким пользовательским кругозором, он всегда пробует новые программные инструменты, ищет альтернативные, а иногда и нестандартные решения. Именно такой исследовательский подход к решению задач и желание понимать происходящие в компьютере информационные процессы и побуждают обычного пользователя заняться программированием.

Получить базовые навыки пользования компьютером можно на уроках информатики, во внешкольном кружке или на специализированных курсах. Можно освоить компьютер и самостоятельно, но на это уйдет больше времени.

Как правило, пользователи работают с программами, запускаемыми под операционной системой Windows (OS, operation system, можно сказать, что это самая главная программа в любом компьютере). Существует огромное множество, как других операционных систем, так и других типов компьютеров. Самые популярные из альтернатив – это OSX (MacOS) и Linux.

Вне зависимости от того, чем именно вы решите заниматься в жизни, вам всегда пригодится умение искать нужную информацию в сети Интернет, пользоваться различными сетевыми сервисами — от электронной почты до платежей онлайн. Прошли те времена, когда для решения какой-то одной конкретной задачи нужно было штудировать тома учебной и справочной литературы. Кроме поисковых сервисов (google.com, yandex.ru) в Интернет существует масса сайтов, построенных по типу вопрос-ответ (лучший для программистов — stackoverflow.com). Это революционный и самый эффективный способ найти решение конкретной проблемы, потратив минимум времени.

Львиная доля самой актуальной информации в сети, и по программированию в том числе, публикуется на английском языке. Поэтому, если вы хотите достигнуть профессионального уровня, то **английский язык для вас крайне важен**.

Знание основ программирования

Как начать программировать? Лучший способ – это занятия в кружке юных программистов при школе или в каком-нибудь внешкольном учебном центре, как станции юных техников (Москва: informatics.mcsme.ru/circles, Петербург: spbtc.ru/information). Там не ставят оценок, как в школе. Есть время попробовать, подумать, стоит ли вообще серьезно заниматься программированием. Поверьте, это очень интересное и увлекательное занятие само по себе, хотя и далеко не для всех. За год интенсивных занятий в кружке или на курсах вы можете вполне реально освоить школьную программу по информатике в сфере программирования. Важно отметить, что в школе не только изучают теоретические основы программирования на примере стандартного набора задач, но и дают другие полезные для программиста знания и навыки: логика, системы счисления, представление информации, работа с программами пакета MS Office.

Какой язык программирования выбрать? Старая шутка гласит: «Выбирай тот язык программирования, на котором практикует ближайший к тебе гуру (авторитетный опытный профессионал), который всегда может помочь советом». Другой совет состоит в том, чтобы выбрать язык программирования, используя который, вы будете чувствовать себя счастливым. А если серьезно, то не важно, какой именно язык вы выбираете, чтобы освоить азы структурного программирования. Основная ваша задача на этом этапе – это понимание таких вещей как алгоритм, код, данные. Вы должны уметь работать с числовыми и строковыми переменными, массивами, использовать циклы, условный оператор, подпрограммы в формате процедур или функций, корректно программировать ввод и вывод данных. Судя по демонстрационным вариантам ЕГЭ по информатике 2012 года, основными учебными языками программирования остаются **Бейсик (BASIC), Паскаль (Pascal) и Си (C)**. С таким же успехом в этот список можно добавить Питон (Python) и Руби (Ruby). Вы не выбираете один язык программирования раз и навсегда. Напротив, плох тот программист, который знает только один язык программирования. Со временем к вам придет осознание того факта, что такое разнообразие языков и сопутствующих им технологий существует не случайно. Дело в том, что в реальной жизни выбирать тот или иной язык программирования нужно в зависимости от поставленной задачи.

Можно изучить основы программирования самостоятельно. В этом случае вам понадобятся учебно-методические материалы. Это может быть самоучитель или набор вводных статей в

электронном виде (tutorial), которых в сети сейчас существует огромное множество даже на русском языке. Из программ вам потребуется компилятор или интерпретатор языка, который вы выбрали, и редактор исходного кода. Желательно иметь пакет программ разработчика с интегрированной средой разработки (IDE). Это специальный набор программ, который совмещает в себе и удобный редактор кода и подключенный к нему компилятор или интерпретатор. Подобные пакеты можно найти даже для очень старых компьютеров на базе операционной системы MS-DOS.

Интерпретатор – это программа, которая транслирует код вашей программы в машинный язык (самый низкоуровневый язык программирования; компьютер “железо” понимает только этот язык программирования) и тут же его выполняет шаг за шагом (интерпретирует). Компилятор – это программа, которая транслирует (компилирует) код вашей программы в специальный исполняемый формат (exe-файл под Windows), по сути, в отдельную программу, которую позже можно самостоятельно запустить даже на другом компьютере под управлением той же или совместимой с ней операционной системы.

Стоит также отметить, что даже базовых знаний и навыков программирования уже достаточно для решения пусть и учебных, но довольно сложных алгоритмических задач. Участие в олимпиадах по программированию и творческих конкурсах разработки программного обеспечения может вылиться в дополнительные преимущества при поступлении в вуз.

Профессиональное программирование

Между начинающим программистом, который с интересом впитывает всё новое, и специалистом, способным реализовать программное решение для какой-то реальной задачи, лежит, что называется, пропасть. Это выражается в уровне понимания происходящих процессов как создания программного обеспечения (ПО), так и его последующего внедрения и использования. Дорогу осилит идущий. Рано или поздно количество часов, потраченных на изучение учебных материалов и просмотр чужих и написание своих кодов, выльется в качественные изменения в понимании будущего специалиста. Лучшее средство для этого – **разработка своего собственного программного проекта**. Пусть даже на некоммерческой основе. В мире существует огромное множество подобных «открытых» проектов, библиотек, исходных кодов.

Термин открытое программное обеспечение (англ. open-source software) обозначает статус некоторого ПО (библиотека компонентов, прикладная программа), доступного в Сети для свободного скачивания, изучения и использования. Такие программы, в зависимости от приложенной к ним лицензии, могут быть использованы любым человеком или организацией в таком же open-source проекте, в коммерческой системе или модифицированы под нужды конкретной задачи.

Работая над своим реальным, пусть пока и не коммерческим, проектом, вы гораздо быстрее получите крайне важный профессиональный опыт: научитесь использовать дополнительные инструменты программиста (отладчик, программную документацию, системные утилиты, различные настройки, всевозможные форматы представления данных и библиотеки модули для работы с ними), искать в сети готовые решения проблем, выбирать оптимальные из них, общаться с другими людьми из мира IT. Одно дело – решать теоретические олимпиадные задачи, и совсем другое – создание полезного ПО, у которого есть конкретная предметная область.

При сегодняшнем огромном числе программистов нередко можно встретить талантливых самоучек, которые самостоятельно дошли до вершин профессии без высшего образования. С другой стороны, это, как правило, сотрудники очень узкой специализации. Достигнув своего локального потолка, такой человек уже не способен расти в профессиональном и карьерном плане. При первой же нестандартной задаче он просто растеряется. Привязанность и к месту работы, и к конкретной функции является весьма уязвимым местом в социальном аспекте.

Вот почему в таких наукоемких сферах как IT **высшее техническое образование играет одну из ключевых ролей**. Оно дает вам как перспективу, так и безопасность в будущем. И речь сейчас вовсе не о формальном дипломе. Обладая широкой теоретической базой, полученной в вузе, вы сможете легко осваивать новые инструменты, языки программирования и даже целые парадигмы (функциональное программирование, логическое программирование, ООП, DSL, AI, Machine Learning). Вам будет гораздо легче найти новую работу. Богатый математический аппарат даст

возможность решать сложные задачи, принимать нестандартные решения, видеть шире и дальше, со временем повести за собой людей, возглавив коллектив от нескольких программистов до мультинациональных групп разработчиков, в качестве системного архитектора (Architect), старшего программиста (Lead Developer), руководителя IT-направления (англ. Chief Information Officer, CIO) или исполнительного директора (англ. Chief Executive Officer, CEO).

Не нужно далеко ходить за примерами. Любой из выдающихся IT-людей современности (Билл Гейтс, Сергей Брин, Стив Джобс и др.) так или иначе в начале своего пути был рядовым студентом технического вуза. И именно в этот период своей жизни, выбрав направление основной деятельности, они стали теми, кем их сегодня знает весь мир.

Вот некоторые из лучших технических вузов России, готовящих и программистов в том числе:

- [Московский государственный университет \(МГУ\)](#)
- [МГТУ им. Н. Э. Баумана](#)
- [Санкт-Петербургский государственный университет \(СПбГУ\)](#)
- [Санкт-Петербургский государственный университет информационных технологий, механики и оптики \(СПбГУ ИТМО\)](#)

Работа в компании

Одного диплома недостаточно для того, чтобы автоматически получить место программиста в серьезной компании. В первую очередь HR'ы (англ. Human Resource, люди в компании, которые занимаются кадровыми вопросами подбора персонала) смотрят на реальный опыт кандидата, который тот должен подтвердить на собеседовании и при реализации тестового задания.

Условно позиции программистов можно разделить на 3 уровня: начальный (Junior Software Developer), средний (Middle Software Developer) и высокий (Senior Software Developer). Причем каждая вакансия подразумевает специализацию по тому или иному языку программирования и технологии. Самые популярные и востребованные из них: Java, C/C++, .NET, PHP, Ruby, Python, Flash/Flex. Кроме того, существует градация по предметной области: программирование под Web, Enterprise-разработка коммерческих решений в сфере управления, создание игр, программирование под мобильные платформы, низкоуровневое и системное программирование.

Только в очень редких случаях вас могут взять на работу в качестве junior программиста или стажера без реального опыта работы. Поэтому я вам настоятельно рекомендую **начать свою карьеру программиста уже на последних курсах университета**. Это может быть стажировка или бесплатные курсы в какой-то компании или даже работа с частичной занятостью (part-time). Таким образом, по окончании вуза у вас будет не только диплом, но и ценный реальный опыт в вашем резюме.

Академическое знание в XXI веке старается не отставать от стремительно изменяющегося мира IT. Поэтому, и наши, и зарубежные вузы тесно сотрудничают с крупнейшими компаниями-разработчиками ПО. Так, например, преподаватели МГУ задействованы в школе обработки информации, занятия которой уже несколько лет организует компания Яндекс. С одной стороны, компании дают самые передовые практические навыки разработки, с другой – получают квалифицированные кадры в качестве выпускников факультетов прикладной математики и информатики.

Александр Алексейчук, программист

Источник: edunews.ru