**Лекция 1. Знакомство с профессиями программиста и тестолога**

Программист — человек, который создаёт компьютерные программы, сайты, веб-сервисы и мобильные приложения с помощью языков программирования — Python, C, C++, Go, Java, JS, Swift и других. Программист работает в команде с другими программистами и дизайнерами, чтобы спроектировать каждую часть приложения или программного обеспечения, а затем определить, как они будут работать вместе.

**История профессии программиста:**

История профессии программиста берет свое начало в 40-50-х годах XX века с изобретением первых электронных вычислительных машин. Первые программы для них создавались на машинных кодах и в ассемблере. Это была кропотливая работа, требовавшая высокой квалификации от специалистов.

Настоящий прорыв произошел в 50-60-х годах с появлением языков программирования высокого уровня, таких как Фортран, Алгол, Кобол, Бейсик. Это позволило существенно упростить процесс создания программ и сделало программирование более доступным. Профессия программиста получила широкое распространение.

В СССР подготовка программистов велась с 1950-х годов на базе математических факультетов университетов. В 1960 году был основан первый вуз по подготовке программистов - Московский институт электронного машиностроения.

С 1990-х годов, с бурным развитием IT-технологий, спрос на программистов резко вырос. Появилось множество новых языков программирования, фреймворков, инструментальных средств разработки. Сегодня программисты востребованы как никогда.

**Востребованность**

Профессия программиста в настоящее время является одной из самых востребованных в России и мире в целом. По данным различных исследований, специалисты в области IT занимают лидирующие позиции по уровню зарплаты и востребованности на рынке труда.

Роль программистов в современном обществе трудно переоценить. Практически все сферы жизни сегодня тесно связаны с информационными технологиями. Программисты создают и поддерживают работоспособность различных приложений, веб-сайтов, систем автоматизации процессов, а также участвуют в разработке новых технологий и продуктов.

Среди родственных профессий программиста можно выделить следующие:

- Тестировщик ПО. Этот специалист отвечает за обнаружение и документирование ошибок в программном обеспечении.

- Системный аналитик. Он анализирует и определяет потребности пользователей, а также преобразует их в функциональные требования для разработчиков.

- Дизайнер интерфейсов (UI/UX). Этот специалист занимается созданием удобного и понятного интерфейса для пользователей.

- Администратор баз данных. Он отвечает за управление и поддержание работоспособности баз данных.

- Системный администратор. Специалист, обеспечивающий надежную и эффективную работу компьютерных систем и сетей.

- Data Scientist. Это специалист, занимающийся анализом больших объемов данных с целью выявления закономерностей и предсказания развития событий.

Атлас новых профессий — это документ, который представляет перспективные профессии и компетенции будущего, востребованные в условиях активного развития технологий. Профессия программиста тесно связана с многими из перспективных профессий, представленных в Атласе. Рассмотрим некоторые из них (https://atlas100.ru/catalog/it-sektor/):

1. Инженер по искусственному интеллекту. Этот специалист занимается созданием и разработкой алгоритмов искусственного интеллекта. Эта работа требует отличного знания программирования и аналитических навыков.

2. Специалист по большим данным (Big Data). Эта профессия включает обработку и анализ больших объемов информации, что невозможно без программирования и знания специализированных инструментов.

3. Blockchain-разработчик. Программисты в этой области занимаются созданием и поддержкой блокчейн-систем и приложений на их основе.

4. Разработчик VR/AR приложений. Профессия, связанная с созданием приложений виртуальной и дополненной реальности, что требует от программистов знаний в области графики и 3D моделирования.

5. Cybersecurity Specialist. Это специалисты, которые занимаются защитой информационных систем от различных угроз, и они тоже нуждаются в глубоких знаниях в области программирования.

6. Bioinformatics Specialist. Эти специалисты используют программирование для анализа и интерпретации биологических данных.

Профессия программиста требует широкого кругозора и навыков, связанных с информатикой и программированием. Однако для успешной карьеры в этой области также полезно иметь хорошие знания в базовых общеобразовательных предметах, таких как:

1. Математика: Основы алгебры, геометрии, тригонометрии и математического анализа могут быть полезны при решении сложных задач программирования и оптимизации алгоритмов.
2. Физика: Понимание физических принципов может помочь в разработке программ для моделирования и симуляции.
3. Английский язык: Множество документации, учебных материалов и сообщений на форумах на английском языке. Хорошее знание английского языка поможет в поиске информации и общении с международными коллегами.
4. Литература и грамотность: Навыки чтения, анализа и написания текстов могут пригодиться при документировании кода и взаимодействии с клиентами или коллегами.
5. История и обществознание: Понимание исторического и социального контекста может быть полезным при разработке программных решений для конкретных отраслей и общественных нужд.
6. Этика и правоведение: Знание основ этики и правовых норм может помочь в разработке программ, соответствующих нормативным требованиям и этическим стандартам.
7. Экономика: Понимание экономических принципов может быть полезным при разработке программ для финансовой сферы и бизнес-анализа.

Информатика - это ключевой предмет для будущего программиста. Информатика представляет собой научное изучение вычислительных процессов, алгоритмов, структур данных и других основных концепций, связанных с компьютерами и программированием. Она включает в себя такие темы, как:

1. Основы программирования: изучение языков программирования, синтаксиса, создание алгоритмов, структур данных и работа с переменными.
2. Архитектура компьютера: понимание внутренних структур компьютера, как работают процессоры, память, ввод/вывод и другие компоненты.
3. Операционные системы: изучение принципов работы операционных систем, процессов, потоков выполнения и управления ресурсами компьютера.
4. Сети и безопасность: знание принципов работы компьютерных сетей, протоколов и мер безопасности, связанных с сетевым взаимодействием.
5. Базы данных: понимание структур данных, SQL и основ проектирования баз данных.
6. Алгоритмы и структуры данных: изучение эффективных алгоритмов, таких как сортировка и поиск, и структур данных, таких как списки, деревья и графы.
7. Теория вычислений: изучение формальных моделей вычислений, автоматов и теории языков.

Информатика предоставляет фундаментальные знания, которые необходимы для разработки программных приложений, и позволяет программисту лучше понимать компьютерные системы, с которыми он работает.

**Возможности получения профессионального образования по профессии в ТГУ и других организациях профобразования Самарской области**

<https://priem.tltsu.ru/bachelor-and-specialist/number-of-seats-for-admission/7/>

**И другие университеты**

1. МГУ им. Ломоносова
2. МФТИ
3. НИЯУ МИФИ
4. СПбГУ
5. НИУ ВШЭ 5
6. МГИМО
7. Новосибирский национальный исследовательский государственный университет
8. НИ Томский политехнический университет
9. МГТУ имени Н. Э. Баумана
10. Санкт-Петербургский политехнический университет Петра Великого

**Специфика подготовки IT-специалистов в вузах**

Подготовка IT-специалистов в вузах имеет свои особенности, которые могут различаться в зависимости от страны, университета и конкретной программы обучения. Однако существуют некоторые общие черты и компоненты подготовки IT-специалистов, которые можно выделить:

1. Основы информатики и программирования: В рамках бакалаврских программ IT-специалисты обычно изучают основы информатики, включая алгоритмы, структуры данных, языки программирования и базы данных. Это формирует фундаментальные навыки, необходимые для разработки программного обеспечения.
2. Специализация: Студенты обычно имеют возможность выбрать специализацию в области информационных технологий, такую как разработка веб-приложений, искусственный интеллект, кибербезопасность, базы данных, сети и другие. Это позволяет студентам сосредотачиваться на конкретной области IT, которая их интересует.
3. Проектная работа: Вузы акцентируют внимание на практической обучающей деятельности. Студенты участвуют в проектах, разрабатывают программное обеспечение, решают задачи и применяют свои знания на практике. Это способствует формированию практических навыков.
4. Лабораторные работы: Часто студенты проводят время в лабораториях, где могут экспериментировать с аппаратным и программным обеспечением, разрабатывать и тестировать программы.
5. Преподаватели и научные исследования: Вузы часто имеют преподавателей и исследователей, специализирующихся в области информационных технологий. Студенты могут участвовать в научных исследованиях и изучать передовые технологии.
6. Сертификации: Некоторые вузы предоставляют студентам возможность получения профессиональных сертификаций, таких как Cisco CCNA для сетевых специалистов или CompTIA Security+ для специалистов по кибербезопасности.
7. Практика и стажировки: Вузы часто сотрудничают с компаниями и предоставляют студентам возможность прохождения стажировок или практик в IT-компаниях, что помогает студентам получить опыт работы в индустрии.
8. Собственные проекты: Студенты могут быть вдохновлены разработкой собственных проектов, стартапов или участием в хакатонах.

**Направления работы:**

1. **Веб-разработка**
   1. **Фронтенд ―** разработка пользовательского интерфейса. Специалист отвечает за визуальную составляющую программы. Например, список магазинов и ресторанов поблизости, которые пользователь видит в приложении для заказа еды, возможность полистать карточки товаров и выбрать желаемое блюдо — всё это результат работы фронтенд-разработчика. Его задача — перевести готовый дизайн-макет в код и отследить, чтобы сайт корректно отображался на различных платформах и мобильных устройствах. (JavaScript, TypeScript)
   2. **Бэкенд ―** разработка бизнес-логики приложения. В бэкенде программист занимается серверной частью: разрабатывает логику работы сервисов, взаимодействия их частей друг с другом и следит за работой сайта или приложения в целом. Если пользователь не может найти сэндвич с тунцом через поиск, но он точно есть в меню — значит, бэкенд-разработчик допустил ошибку. Его задача структурировать информацию в базе данных и обеспечить их обмен или передачу между запрашиваемыми сервисами. (Java, Python, Go, PHP, Ruby)
2. **Разработка мобильных приложений.** Программисты разрабатывают приложения для смартфонов, планшетов, умных часов и других мобильных гаджетов. Самые популярные: мобильные версии социальных сетей и банков, мессенджеры, электронные книги, переводчики.



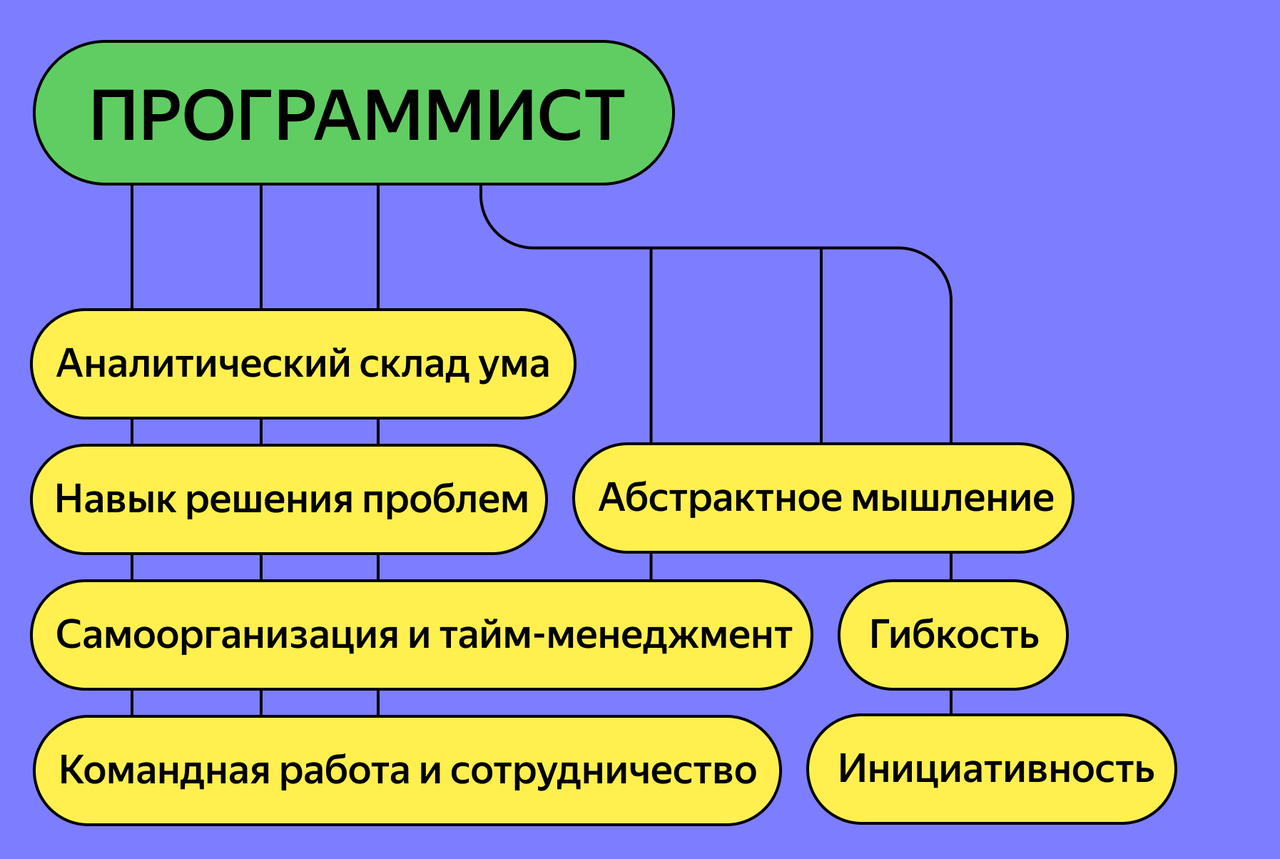
1. Десктопная разработка. Суть работы программиста этого направления — создавать программное обеспечение для различных операционных систем. Такие приложения знакомы пользователям стационарного компьютера или ноутбука. Например, это фотошоп, Microsoft Word и Excel, блокнот — их можно скачать в магазине приложений. Популярные языки: С++, C#, Java, JS

Должностные обязанности

В основные задачи профессионала входят:

* описание блоков программы;
* разработка ПО, операционных систем;
* проверка на совместимость с приложением;
* умение пользоваться условными обозначениями в коде;
* умение создавать подробные инструкции по эксплуатации;
* знание работы алгоритмов для поиска, сортировки и обработки информации;
* работа с библиотеками, фреймворками для эффективного выполнения поставленных задач;
* чтение кода;
* работа с системами управления.

**Необходимые качества и навыки разработчика**



* Аналитический склад ума. Программист должен уметь критически оценивать информацию. Например, судить о полезности фреймворка или технологии не по звёздочкам на GitHub, а проверив производительность и скорость работы через синтетические тесты и бенчмарки. Без базового знания логики трудно разобраться, как выполняется программное обеспечение и как писать простой и легко поддерживаемый код. К тому же специалист, который умеет выстраивать логические связи, может легко выполнять большие задачи, разбивая их на мелкие части.
* Навык решения проблем. В коде часто что-то ломается. Умение справляться с трудностями поможет спокойно подойти к отладке кода. Обладая этим навыком, программист может легко преобразовать абстрактные запросы коллег или клиентов в реальные задачи и выполнимые решения.
* Самоорганизация и тайм-менеджмент. Многим программистам в самых разных отраслях приходится работать в сжатые сроки, что требует от них чёткого планирования времени, усидчивости и сосредоточенной работы. Но в этой сфере сотрудники часто сталкиваются с перфекционизмом, прокрастинацией и синдромом самозванца.
* Командная работа и сотрудничество. В процессе разработки приложения или веб-сайта разные специалисты внутри компании неизбежно будут использовать код, который принадлежит другим командам, или полагаться на него. Чтобы это не затрудняло процесс, программисту нужно уметь сотрудничать с коллегами из разных отделов.
* Абстрактное мышление. Иногда уже готовые и реализованные алгоритмы не подходят для проекта и нужно найти новый способ решения проблемы. Научиться мыслить абстрактно сложнее, чем следовать шаблонным решениям, поэтому ценится разработчик, который освоил оба подхода.
* Гибкость. Изначально заданные требования могут измениться, например, клиент попросит внести правки и программисту придётся переписывать часть кода. Открытость к изменениям и быстрое включение в новые запросы — важный элемент сотрудничества и командной работы.
* Инициативность. Энтузиазм и желание помогать бизнесу зарабатывать с помощью новых решений повышает ценность разработчика и способствует его карьерному росту.

**Задачи и обязанности программиста**

Программист должен обладать определённым набором технических навыков: разбираться в используемом языке и его экосистеме, понимать, как устроен и функционирует интернет.

1. Разбираться в алгоритмах и структурах данных. Чтобы развить этот навык, можно начать с решения алгоритмических задач на LeetCode. Сервис позволяет выбрать уровень сложности задачи, её популярность среди других пользователей и тему. Структуры данных относятся к способу расположения данных в памяти компьютера;
2. Знать языки программирования. Чтобы компьютер понял, что от него хотят, нужно обращаться к нему на языке программирования. Выбор языка зависит от отрасли программирования. Вот некоторые распространённые языки:

● Python — высокоуровневый язык программирования общего назначения. Его часто используют в разработке веб-приложений и прикладного программного обеспечения, а также в машинном обучении и обработке больших данных;

● Java — это язык общего назначения, который часто используется для разработки десктопных и мобильных приложений на Android, программ для работы с большими данными, веб-серверов;

● С++ — расширение языка программирования C. C++ часто используют для разработки видеоигр, веб-сервисов, нейросетей и даже графики для метавселенных.

3) Иметь опыт работы с фреймворками. Фреймворки объединяют набор технических средств, которые упрощают выполнение задачи и ускоряют работу программиста. Их принято разделять на два направления:

● фронтенд-фреймворки — React, Vue, Angular — отвечают за внешний вид проекта и работу в браузере. С их помощью можно создавать удобные пользовательские интерфейсы, адаптировать сайты под разные устройства, добавлять анимацию;

● бэкенд-фреймворки — Django, Flask, FastAPI — нужны для хранения, обработки и анализа больших массивов данных, они отвечают за работоспособность сайта или приложения, которые на нём реализованы.

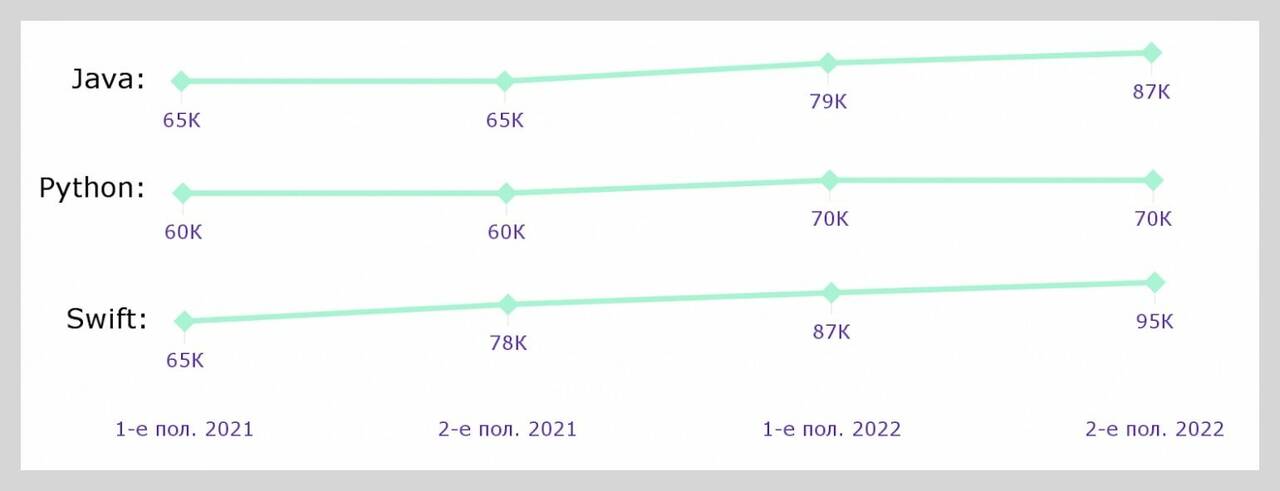
4) Иметь представление о больших данных. Сбором, обработкой, анализом и расшифровкой больших данных занимается аналитик данных. Но утилиты и алгоритмы для их анализа пишут программисты, поэтому будет полезно понять, как они устроены.

5) Уметь отладить код. Это помогает программисту улучшить свои навыки владения языком — изучить его тонкости, исправляя ошибки. Искусственно создать ошибку непросто, поэтому, чтобы набраться опыта в этой области, нужно не бояться создавать сложные проекты и по ходу решать возникающие проблемы. Другой вариант — исправлять ошибки в проектах с открытым исходным кодом.

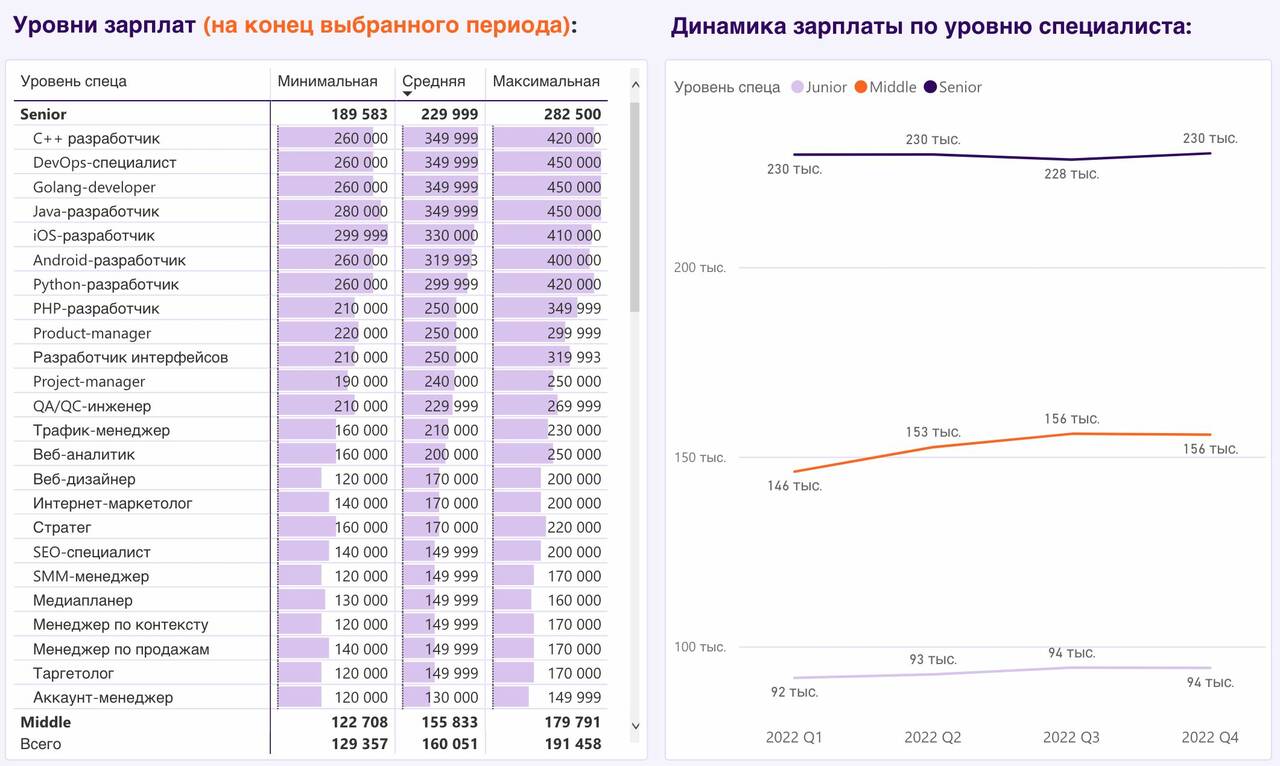
6) Понимать, как работают сетевые протоколы и безопасность. Вопросы безопасности особенно важны для серверных разработчиков, поскольку они отвечают за взаимодействие и хранение пользовательских данных и приватной информации: номеров карт, паролей. Программисты из других отраслей также могут применять эти знания. Без понимания принципов безопасности вопрос времени, когда данные будут скомпрометированы или украдены. Знание сетевых протоколов ― правил связи между устройствами в сети ― важно для программистов, работающих над сетевым сервисом, например облачным файловым хранилищем, или для тех, кто занимается корпоративными сетями компании.

**Востребованность и актуальность профессии**

По данным Хабра, зарплата начинающего программиста Java в 2022 году составляла 87 000 ₽, у специалистов, которые выбрали Python, — 70 000 ₽, а у iOS-разработчиков — 95 000 ₽.



Карьерный рост программиста подразумевает соотношение навыков, опыта и зарплаты. Чем шире круг обязанностей, тем выше зарплата и дополнительные способы поощрения, например: стоки, акции за вклад в проект, лояльность компании.



**Формат работы программистов**

Существуют три основных формата работы — штат, фриланс и аутсорс. Какой вид выбрать — зависит от предпочтений программиста и возможностей компании. На стартовых позициях не всегда удаётся найти удалёнку, но это возможно.

Работа в штате подразумевает официальное трудоустройство и работу из офиса, но по согласованию с работодателем можно трудиться и дистанционно. Даже на удалёнке отношения с компанией регламентируются трудовым договором, поэтому за специалистом остаётся право на оплачиваемые больничные и отпуска.

На фрилансе. Программист не связан трудовым договором с работодателем, работает на себя и платит налоги как самозанятый или ИП. Загруженность регулируется самостоятельно — можно работать над проектами одного заказчика или сотрудничать с несколькими. Доходы определяют не отработанные часы, а выполненные заказы.

Аутсорс. Это официальная работа в агентстве или студии, но при этом программисты решают задачи других компаний — в зависимости от особенностей проекта, на удалёнке или в офисе заказчика.

**Плюсы и минусы профессии**

IT-компании предлагают опытным программистам высокие зарплаты, карьерный рост и возможность выбора локации, оборудование и девайсы для рабочего процесса. Привлекательным фактором для многих сотрудников выступает удалённый или гибридный формат работы. Взамен программисту нужно уметь адаптироваться к быстро меняющейся среде, запросам рынка и потребностям заказчика. Разберём другие плюсы и минусы.

|  |  |
| --- | --- |
| **Преимущества** | **Недостатки** |
| Удалённая работа. Программисты могут выполнять большинство своих задач из любого места с доступом в интернет. | Быстро меняющаяся среда. Работа в сфере технологий предполагает готовность к переменам и умение быстро адаптироваться. Если это вызывает тревогу, быть программистом будет трудно. |
| Перспективы роста. Это сфера, где можно быстро построить карьеру, ― часто достаточно двух лет для перехода на следующую ступень. Как следствие — повышение зарплаты, более интересные задачи, возможность выбирать позиции и проекты. | Рутинные задачи. Часто программисты ищут новаторские способы решения различных проблем, но иногда им подолгу приходится выполнять скучную работу. Например, разработка программного обеспечения целиком состоит из ряда повторяющихся задач. Это может значительно снизить мотивацию. |
| Высокие зарплаты. В IT опытные разработчики много зарабатывают, поэтому создание полезной программы или приложения может принести не только профессиональное удовлетворение, но и заметно улучшить качество жизни. | Слишком высокая конкуренция. На рынке много разработчиков начального уровня, которым трудно найти работу после окончания курсов и стать востребованными. За продвижение по службе придётся соревноваться. |
| Непрерывное развитие. Обычно программисты берутся за большие задачи, разделяют их на множество мелких частей и ищут способы их решения. Этот навык полезен не только в работе, но и в решении повседневных проблем. Также разработчик должен следить за развитием технологий и при необходимости освоить новую версию языка или нового инструмента. | Риск выгорания. Всего за пять лет в IT-сфере можно вырасти до опытного специалиста, или сеньора. Это большой объём ответственности и задач, который может привести к выгоранию. |
| Баланс между работой и личной жизнью. В некоторых компаниях работодатель устанавливает тайм-трекер, чтобы контролировать рабочее время и фиксировать личные показатели разработчика. В остальных случаях код пишут 4–5 часов в день. Оставшееся время отводится на планирование, синхронизацию с командой, проверку и обдумывание идей или альтернативных подходов к решению проблемы. Такой график освобождает время для личной жизни. |  |

**Как стать программистом**

1. Освоить программирование самостоятельно с нуля.

Учиться можно по бесплатным урокам на YouTube или подкастам. Рекомендуем ориентироваться на тех спикеров, чьи имена на слуху. Иначе есть риск потратить время на изучение устаревшей или недостоверной информации.

Для самостоятельного изучения подойдёт официальная страница Практикума или подкасты Moscow Python, где группа экспертов обсуждает IT-новости и инсайты с конференций. За новостями и новинками — в англоязычные источники. Текст страницы чаще всего можно перевести прямо в браузере.

2. Учиться у ментора.

Обычно это знакомый с опытом в IT-сфере или, предположим, эксперт, который давно ведёт Telegram-канал и делится своими знаниями. Обратная связь от специалиста, который уже прошёл похожий путь, может стать для новичка решающим фактором — оставаться в профессии или нет. Хороший ментор может дать мотивацию начинающему программисту двигаться дальше, несмотря на трудности.

3. Пройти онлайн-курс.

Чтобы стать программистом — не всегда нужно сдавать экзамены и получать высшее образование. Программы вузов часто отстают от запросов рынка на 3–5 лет. А курсы — простой способ сэкономить время, получить актуальные навыки, найти комьюнити в IT-сфере.

4. Пойти учиться в ТГУ по одному из направлений (https://priem.tltsu.ru/bachelor-and-specialist/number-of-seats-for-admission/7/).

**Профессия тестировщика (QA-инженер)**

**История профессии**

Профессия тестировщика зародилась в 1950-1960-х годах, когда в процессе создания программного обеспечения стали привлекаться специалисты для проверки работоспособности программ и поиска ошибок. Изначально роль тестировщика была вспомогательной и не выделялась в отдельную профессию.

В 1970-1980-х годах, с разработкой крупных и сложных программных комплексов, тестирование ПО выделилось в самостоятельный этап. Появились отдельные должности тестировщиков в штате.

Настоящий расцвет профессии тестировщика произошел в 1990-2000-х годах, когда разработка ПО превратилась в крупную индустрию. Были разработаны стандарты и методологии тестирования, появились специализированные инструменты. Возник высокий спрос на профессиональных QA-специалистов.

**Распространенность и востребованность**

В настоящее время в России насчитывается порядка 300 000 специалистов в области тестирования ПО. Это одна из наиболее востребованных и быстрорастущих профессий в IT-сфере.

Высокая потребность в тестировщиках обусловлена активным развитием рынка разработки программного обеспечения и продуктовых компаний, выпускающих цифровые продукты и сервисы.

**Родственные профессии:**

* Разработчик программного обеспечения;
* Инженер по технической поддержке ПО;
* Инженер по тестированию в области информационной безопасности
* Специалист по автоматизированному тестированию

**Связь с трендами по Атласу профессий:**

Тестирование ПО тесно связано с такими технологическими трендами, как искусственный интеллект, Интернет вещей, Big Data, роботизация. Развитие этих областей предъявляет новые требования к тестированию.

Базовые дисциплины:

* Математика
* Информатика
* Русский язык

Эти предметы закладывают основу логического мышления и грамотной работы с информацией.

Возможности обучения в Самарской области:

* Курсы тестировщиков ПО
* IT-колледжи
* Вузы, готовящие специалистов в области информационных технологий

Характеристика труда:

Основные задачи тестировщика:

* Функциональное тестирование ПО на соответствие требованиям
* Поиск ошибок и дефектов в программном коде
* Разработка тест-кейсов и проверочных сценариев
* Составление bug-репортов
* Анализ результатов тестирования
* Регрессионное тестирование

Психофизиологические особенности:

* Высокая концентрация и устойчивость внимания;
* Способность к монотонной работе;
* Аккуратность и внимательность к деталям;
* Стрессоустойчивость и терпение

Важные качества:

* Скрупулёзность и дотошность;
* Настойчивость в достижении результата;
* Аналитические способности;
* Коммуникабельность.

Медицинские противопоказания:

* Расстройства зрения и нарушения моторики;
* Повышенная нервная возбудимость;
* Быстрая утомляемость и истощаемость.