

**Прокин Александр Александрович**, преподаватель,

*ФГБОУ ВО «Национальный исследовательский Мордовский государственный университет им. Н.П. Огарёва», г. Саранск*

**Баландин Илья Андреевич**, студент ФДО и СПО ФГБОУ ВО «Национальный исследовательский Мордовский государственный университет им. Н. П.

*Огарёва», г. Саранск*

*E-mail: [Sergushinaes@yandex.ru](mailto:Sergushinaes@yandex.ru)*

## КАЖДЫЙ МОЖЕТ СТАТЬ ПРОГРАММИСТОМ

**Аннотация:** В данной статье были рассмотрены вопросы становления программистом, требования к программистам, а также рассмотрены основные ошибки начинающих программистов.

**Ключевые слова:** программист, новичок, требования, ошибки.

**Abstract:** In this article questions of programmer formation, programmer requirements were considered. Except of this, basic mistakes of beginning programmers are considered too.

**Key words:** programmer, beginner, requirements, mistakes.

Всех айтишников спрашивают, кем нужно быть, чтобы стать программистом? Стать программистом может каждый, нужно лишь развивать в себе некоторые качества. Их минимум два: умение анализировать и думать. Образование неважно, но обладание техническим складом ума упростит процесс становления программистом [1].

Знание математики и значимость этих знаний зависит от области работы в будущем. Например, в Data Science обязательно надо очень хорошо знать математику, так как в этой сфере необходимо создавать сложные модели.

Английский язык – хоть и не обязателен, но уже является негласным стандартом данной сферы. Не все по языкам программирования и данной сфере можно найти на русском языке. Огромное количество документации, обучающих уроков, новостей, лекций находятся в интернете на английском языке [2].

Постоянное обучение.

Развитие IT-отрасли непрерывно идет полным ходом. Постоянно разрабатываются различные новые языки программирования. Различные библиотеки, дополнения, инструменты появляются еще чаще. Поэтому, решив стать программистом, нужно быть готовым к постоянному обучению. ВУЗы и другие учебные заведения дают только базовые шаблоны мышления. Очень часто эти знания не применяются на практике. Реально важную и нужную информацию приходится находить самостоятельно [3].

В обучении языкам нельзя делать перерывы. Если хоть один большой перерыв, то все практические навыки «уходят под ноль».

Очень тесно с этой профессией находится тайм-менеджмент. Хоть это и не относится к программированию, но правильно распоряжаться своим временем необходимо, от этого зависит учеба и работа. В своем времени помимо работы, необходимо уделять время общению с коллегами, именно от них можно узнать о многих новинках и профессиональных хитростях. Где найти нужную актуальную информацию?

Интернет – лучший источник информации.

И для начинающих программистов, и для профессионалов своего дела основной источник знаний – интернет. Там множество как специализированных ресурсов для программистов, так и сайты общего характера. Например, через Twitter максимально удобно следить за новостями и лайфхаками от профессионалов [4; 5].

На YouTube публикуются различные tutorиалы, записи семинаров, различные лекции, презентации, которые демонстрируют возможности современных технологий. Очень часто на таких платформах как Twitch и

YouTube проводят лайвкодинги. В данных трансляциях можно понаблюдать за профессионалами и их работой над кодом и задать любой интересующий вас вопрос напрямую в чате трансляции.

При работе с кодом необходимо смотреть на свой работающий код и анализировать его. Для данных целей подходит такая платформа, как GitHub и другие подобные сервисы.

Программист – универсальная специальность.

Программисты в IT-сфере – универсальны. Им необязательно останавливаться на определенной сфере деятельности. Любой программист может из одной сферы технологий перейти в другую. Таким образом любой программист, например, может перейти из геймдева в разработку сайтов, и т.д.

Обучаясь такой профессии возникает вопрос, касаясь технологий, в которые стоит «удариться». Сразу исследовать максимальное число технологий или выбрать что-то одно. Ответ приходит с опытом.

Стандартные ошибки новичков.

Усложнение кода без необходимости.

Любой листинг кода должен читаться без проблем, чтобы сразу было ясно, что выполняет данный кусок программы. Если вы спустя некоторое время не можете разобраться в своем листинге или как вы использовали те или иные библиотеки, то это очень плохо.

Изобретение велосипеда.

Почти для каждой задачи можно найти уже готовое решение на просторах интернета. Если же вы не можете найти, то скорее всего был некорректно сформулирован запрос. Когда начинающий программист находит готовое решение, ему может показаться, что самописный код лучше найденного решения. Однако это не всегда так. Зачастую начинающий программист закапывается в своем коде и все равно начинает использовать найденную наработку [6].

Отказ от книг

Начинающий программист очень быстро забивает на книги по программированию и начинает набираться опыта на статьях из интернета. Однако данные статьи не помогают новичку сформировать фундаментальную теоретическую базу. Из-за чего он медленнее находит оптимальные решения, допускает больше ошибок.

#### Нюансы

Если до дедлайна еще далеко, то всю структуру, все нюансы следует продумать заранее, перед тем как начать писать код. Важнее улучшать навыки правильного мышления, нежели оттачивать быстрый набор кода. Иначе начинающий программист может потерять много времени на поиск ошибок и стабилизацию алгоритмов [4; 5].

#### Чередование быстрой и качественной работы

Появляются задачи, которые необходимо максимально оперативно решить, даже если код выглядит кривым и некрасивым. Однако есть и задачи, при решении которых следует упираться в качество. Если чередовать данные подходы, то в итоге можно научиться выполнять задачи быстро и качественно.

### **Библиографический список:**

1. Лапчик, М. П. Методика преподавания информатики: учеб. пособие для студ. пед. Вузов / М. П. Лапчик, И. Г. Семакин, Е. К. Хеннер; под общей ред. М. П. Лапчика. – М.: Издательский центр «Академия», 2001. – 624 с.
2. Лапчик, М. П. Теория и методика обучения информатике / М. П. Лапчик, М. И. Рагулина, Н. Н. Самылкина, И. Г. Семакин, Е. К. Хеннер. – М.: Издательство Academia, 2008. – 592 с.
3. Могилев, А. В. Информатика: учеб. пособие для студ. пед. вузов / А. В. Могилев, Н. И. Пак, Е. К. Хеннер; под ред. Е. К. Хеннера. – М.: Издательский центр «Академия», 2004. – 848 с.
4. Прокин А. А. Современное состояние и основные проблемы интернет-торговли в российской федерации / А. А. Прокин, В. А. Богатырская, Е. С. Сергушина, И. С. Листратов // E-Scio. – Саранск, 2018 – № 3 (18). – С. 36-41.

5. Прокин А. А., Богатырская В. А., Сергушина Е. С., Лукин М. А. Создание и продвижение интерактивного веб-сайта для коммерческой организации [Электронный ресурс] // E-Scio: Электронное периодическое издание «E-Scio.ru». – Режим доступа: <http://escio.ru/wp-content/uploads/2018/05/Прокин-А.-А.-Богатырская-В.-А.-Сергушина-Е.-С.-Лукин-М.-А.-.pdf>.

6. Панфилов С. А. Алгоритм энергосбережения для автономных систем теплоснабжения / С. А. Панфилов, О. В. Кабанов // Вестник Южно-Уральского гос. ун-та. Серия: строительство и архитектура. / Южно-Уральский гос. ун-т. – Челябинск, 2017. – Т. 17. – №1. – С. 67–74.