Методы изучения новых языков программирования

Программирование как новый вид человеческой деятельности появился сравнительно недавно. Для ЭВМ первых поколений это было достаточно сложным и трудоёмким занятием, искусством которого овладевали за многие годы. Прогресс вычислительной техники и развитие программирования привели к тому, что им начало заниматься всё большее число людей, а профессия программиста стала престижной. Многие школьники мечтают стать программистами, поэтому материал данной темы имеет большое значение для профориентации.

Программирование - это раздел информатики, изучающий вопросы разработки программного обеспечения ЭВМ. В узком смысле под программированием понимают процесс создания программы на одном из языков программирования. Разработку средств системного программного обеспечения и систем программирования называют *системным программированием.* Создание прикладных компьютерных программ принято называть *прикладным программированием.* По такому же принципу проводят деление программистов на системных и прикладных.

Существует несколько парадигм (образчиков) программирования:

- процедурное;
- логическое;
- функциональное;
- объектно-ориентированное.

Методика изучения языков программирования достаточно хорошо разработана. Языки программирования делятся на две большие группы: машинно-ориентированные (Автокод, Ассемблер) и языки высокого уровня. Языками первой группы пользуются весьма малое число программистов профессионального уровня для специфических целей. Большинство программистов используют в настояще время языки высокого уровня, причем имеет место некоторая мода на языки. В то же время язык Фортран существует уже 50 лет и всё ещё достаточно популярен среди физиков-теоретиков и части математиков.

Существует более 500 языков программирования. Следовательно, вполне нормально начать изучать новый язык программирования прямо сегодня. Возможно, вы знакомы с C ++ и Java, но ваша работа требует знания Python, или вы хорошо разбираетесь в Python, а вам нужно программировать на работе на Java. Видя всё это множество и разнообразие языков со своими различиями в синтаксисе и основах конструкции языка, возникает вопрос «Как запомнить все эти конструкции, а самое главное как впоследствии не забыть?» . Ведь работая над одним проектом на одном языке, ты непременно забываешь некоторые нюансы другого.

1. Метод конспектирования.

Чем хорош метод конспектирования? Это был очевидный факт. Выполнив один, пять, десять раз какое-то действие в программировании, но потом долгое время не возвращаясь к нему из-за изучения других языков или углубляясь в текущий, новичок скорее всего не сможет его раз и навсегда запомнить. Тогда что выгоднее? Тратить время на конспектирование по ходу изучения или на поиски информации каждый раз, когда ты что-либо при необходимости?

Я пришел к следующему выводу, который, как мне кажется, в большей степени касается новичков: лучше писать конспекты, если изучаются какие-то базовые принципы (например, из самоучителя), которые доступно разжевываются именно в этом источнике. Ведь если потом они забудутся, нужно будет вспомнить источник, найти страницу (в книге), время (в видеоуроке) и заново ознакомиться с материалом.

Также хочу сразу оговориться, что данное правило не распространяется, например, на встроенные функции, коих огромное количество и которые гораздо разумнее искать в документации к языку, где, как правило, есть и описание и примеры. Можно записать разве что основные, которыми будешь регулярно пользоваться во время обучения. Конспектировать — не значит просто копировать. Конспект в данном случае подразумевает не принцип «Ctrl+C -> Ctrl+V», а соблюдение следующего порядка действий:
1. Прочитать материал (абзац, главу, описание чего-либо);
2. Если нужно, очень кратко выписать суть, которую вы сами потом сможете понять (именно это не оставит вам выбора, кроме как разобраться в вопросе);
3. Записать примеры кода (если нужно).

Плюсы и минусы.

Используя этот метод некоторое время, однозначно могу сказать, что огромным минусом является необходимость структурировать информацию по разделам, а также оформлять её. Но в то же время это является и плюсом, ведь в итоге очень хорошо запоминается, какую информацию и куда ты вносил. А после прочтения одного предложения, записанного своими словами, сразу же открывается вся картина. Нет необходимости искать, в каком месте книги или, не дай Бог, видеоурока, затрагивался нужный вопрос.

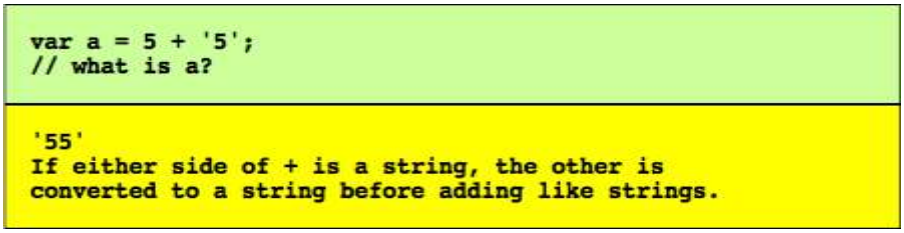
2. Интервальное повторение

Когда вам попадается новый факт, вы рискуете в скором времени его забыть, если он не будет периодически мелькать у вас перед глазами.
Программное обеспечение на основе метода интервального повторения все сделает за вас, вам лишь необходимо занести в него ту информацию, которую вы хотели бы запомнить, затем каждый день проходить тестирование – программа определяет интервалы повторения, основываясь на вашей оценке собственных знаний.
Если, ответив на вопрос, вы оцениваете его как легкий, то в следующий раз программа его предложит не скоро, если же вы допустили ошибку или не вспомнили ответ, программа задаст вопрос снова несколько раз, пока вы не справитесь.
На этом сайте ankisrs.net можно скачать программу Anki – бесплатную и самую популярную программу такого рода. Существуют версии для Mac, Windows, Linux, iPhone, Android и т.д. Написана на Python.

Вам предстоит создать огромное количество карточек. Вопрос — с одной стороны. Ответ – с другой. **Карточки нужны для того, чтобы запомнить то, что выучил.** Прежде чем создавать карточку, посвященную чему-либо, нужно в первую очередь это что-либо **понять и запомнить**. Каким бы не был ваш источник – будь то книги, лекции, видеозаписи, код, онлайн-уроки, или любой другой – создавать карточку можно лишь после того, как новая информация была выучена и усвоена.

Если учите JavaScript, и сталкиваетесь со следующим правилом:
«Оператор сложения (+)… если хотя бы один операнд – строковый, второй также преобразуется в строковый, таким образом, результат сложения – объединение двух строковых операндов.»

Вы проверяете эту информацию на практике, изучаете вопрос вдоль и поперек, в итоге выясняете, что 1 + '1' таки равно '11'. После чего создаете карточку, которая поможет запомнить этот факт:



Иногда я пишу пояснение. Иногда в нем нет нужды, поскольку в некоторых случаях достаточно простого ответа. Для наиболее эффективных результатов, запускайте программу каждый день. Если надолго забросить ее, у вас собьется график, в итоге вам придется заново учить вещи, которые вы по идее должны помнить. Вы можете запомнить тысячи фактов, тратя на них по 20 минут в день.

3. Вносить вклад в проект с открытым исходным кодом на этом языке.

Естественно для того чтобы вносить изменения в проект вы должны уже обладать базовыми навыками программирования на этом языке, и конечно быть достаточно опытным программистом вообще.

Как помогают проекты с открытым исходным кодом?

Итак, теперь вы можете быть удивлены тем, как работа с open-source проектами может помочь вам в изучении новых языков программирования. Существуют различные аспекты. Давайте обсудим их один за другим.

Качество кода

У большинства хороших проектов с открытым исходным кодом достаточно строгий code conduct (соглашения, как писать код), которого вы должны придерживаться, чтобы совместно работать над проектом. Это поможет вам адаптироваться и написать код хорошего качества, даже если вы еще на начальном этапе изучения языка.

Кроме того, у вас есть шанс взглянуть на существующий код и посмотреть, насколько хорошо он написан и документирован.

Большая часть в коммитов open-source проекты сопровождается тщательным анализом написанного кода. Вы получаете отзывы от экспертов, связанных с этим проектом, и, следовательно, это дает вам возможность улучшить понимание языка.

Это похоже на получение бесплатного руководства о том, как писать хороший код.

Даже если вы успешно обучаетесь самостоятельно, рано или поздно наступит момент, когда станет необходимо показать свои навыки окружающим. Какая польза от того, что вы пишете, если этого никто не видит? Обучение в команде позволяет не только быстро получать ответы на вопросы, но и даёт возможность заявить о себе, когда вы почувствуете внутреннюю уверенность