Программирование в современном мире имеет важное значение. Ни одна современная отрасль не обходится без программного обеспечения. Ни одна современная область науки не развивается так быстро и динамично, как в программирование. Как только компьютеры стали общедоступными, возникла потребность в различных программных продуктах. Программирование стало не просто уделом военных или учёных, оно вошло в жизни людей по всему миру и стало что-то вроде субкультуры. В свою очередь, как я уже писал выше, IT-технологии развиваются очень быстро, появляются новые платформы, языки программирования, технологии. Современному программисту приходится постоянно «держать руку на пульсе». Именно поэтому программу обучения школьников программированию нужно тщательно продумывать со взглядом в будущее. Подход к программированию в частности и информационным технологиям в общем постоянно изменяется, но что-то остаются неизменным. Эту базу необходимо передавать учащимся, поэтому я делаю упор на то, чтобы обучающиеся в первую очередь учились самостоятельно искать нужный материал и изучать его. Необходимо объяснить, что, работая в этой сфере, им, вероятно, придётся буквально за вечер изучить новый язык или framework (библиотека готовых функций), и поэтому требуется научиться самостоятельно и быстро осваивать информацию. Но давайте остановимся более подробно на том, с какими проблемами сталкиваются современные школьники и студенты начальных курсов.

В программировании есть несколько парадигм, подходов к созданию программ: модульная парадигма, процедурные парадигма, объектно-ориентированная парадигма. В школах обычно преподают процедурный подход к программированию, в то время как ООП (объектно-ориентированное программирование) - очень редко или никогда. Проблема заключается в том, что все современные языки объектно-ориентированные. А так как ООП - это основы всех современных языков программирования, то его необходимо начинать изучать ещё в школе, чтобы учащиеся, придя в высшие учебные заведения, уже имели правильное представление о программировании. Уже много раз замечал, как студентам сложно разобраться в ООП, так как там немного друга парадигма, немного другой подход к созданию программ. В ООП появляются объекты, классы и методы, такие понятия как инкапсуляция, наследование и т.д.  
 Поэтому я считаю, что чтобы повысить уровень освоения предмета, необходимо готовить учащихся к реалиям современного мира уже со школьной парты.   
 ООП появилось в начале 70-х годов, появлению ООП способствовал кризис, наступивший в программировании в те годы.   
Программные продукты становились все объемнее и разрабатывались огромными коллективами в течение нескольких лет. Неудивительно, что в конечном продукте находили тысячи ошибок, исправление которых требовало огромных ресурсов. На помощь пришел модульных подход к созданию программ, который основывался на «разделении труда»: каждый программист отвечал за свой модуль и по завершению тестирования передавал его дальше; другие программисты работали с этим модулем только через его интерфейс. В дальнейшем этот метод перерос в ООП и широко используется повсеместно. Я стремлюсь, чтобы мои учащиеся умели разрабатывать программы, используя современные и релевантные подходы и технологии, а не пользовались наработками 80-х годов.

Я разбиваю курс программирования на несколько основных блоков, которые зависят от возрастной группы учащихся. В 7 и 8 классах изучается базовое программирование: основные операторы, типы данных, условия; учимся понимать, что такое блочная структура. На этом этапе необходимо заложить правильное понимание программирования, понимание переменных и типов данных, условий. Как результат учащиеся на этом этапе должны уверенно разрабатывать собственные программы.

Учащиеся 9-х классов учатся использовать циклы и одномерные массивы, более подробно работают со строчными данными. На этом этапе необходимо научить работать с циклами и массивами, а также понять важность правильности написания кода.  
Учащиеся знакомятся с понятием «валидация кода», осваивают международные стандарты оформления кода.

Учащиеся 10 классов, как правило, уже имеют уверенное представление о базовых знаниях, таких как переменные, условия, циклы, массивы. На том этапе мы изучаем процедурную парадигму программирования. Я подробно объясняю, что такое процедуры и актуальность их применения. Учащиеся создают собственные функции и используют их на практике, изучают матрицы, двухмерные массивы. Далее мы знакомимся с понятием «структуры», разрабатываем собственные типы и классы.