Добрый день тому, кто прочитает мою эссе! Сегодня я буду излагать свои мысли по поводу выступления от Брета Виктора про будущую программирования. В выступлении идёт речь про 2 основных сюжетных линии — про то, что будет из себя представить программирование после где-то в районе 40 лет, и про то, как мы программисты должны быть непредубеждённым перед новыми идеями. Эти сюжетные линии тесно связаны между собой, так как новые идеи могут не воплощаться просто из-за догмы некоторых лиц, и без восприимчивости к новым идеям человек просто не будет придумать новые идеи. Тогда вопрос: зачем придумать чего-то нового, если всё хорошо?

Ответ довольно простой: если считать, что мы, как программисты, осознаем свои действия, знаем, что мы делаем, то таким образом мы огораживаем свой кругозор от новых типов мышления, где всё гораздо проще и эффективнее. Возьмём тот пример из видео: до того, как Стэн Полей изобрёл SOAP, люди писали программы на чистом бинарном. Те самые люди активно отвергли идею ассемблера. Потом почему-то «оказалось», что программирование на языке ассемблера довольно неплохая идея. Такая же история наблюдалась и с фортраном. От этого можно сделать, по моему мнению, очевидный вывод. Программист имеет доступ к новым способам мышления только лишь избавившись от своей предвзятости.

В выступлении упоминаются 4 основных идеи: прямая манипуляция данных, целевое программирование, пространственное представление информации и параллельное программирование. В качестве примера для первого пункта он привёл такую доску как Sketchpad, где она смогла выравнивать криво изрисованную фигуру в что-то внятное, провести гомотетичные преобразования фигур и свободно изменять вес гири, подвешенного от середины моста. Это возможно с помощью прямой манипуляции данных и непосредственного задавания ограничения. Для второго примера он привёл Planner — язык программирования, где вместо подсказывания алгоритма программист должен подсказать цель, к решению которого должен прийти сам компьютер. От этого вытек Prolog — подмножество Planner-а, программирование на котором является примером логического программирования. На этом заложено развитие искусственного интеллекта. Ещё он привёл такой пример, где 2 ВПл-ы находил общий язык путём самостоятельного достижения к этому. Это залог ARPANET-а, или интернета. Третий пример — пространственное представление информации — он привёл такие понятия, как NLS, RAND(GRAIL), Smalltalk и Pluto, где они все имеют уникальные возможности представить информацию в более интерактивном виде. И последний пример — параллельное программирование, где он привёл в качестве примера модель актора, где основной единицей является актор, который «разговаривает» с другими акторами, принимает решения на ответ от других акторов, создаёт новые акторы. Такой модель используется для проектирования параллельных вычислительных систем.

Этот выступление устроено так, что он из 1973 года. На самом деле эти пункты уже реализованы. Но с другой стороны можно увидеть, что мы не продвинулись от старых привычек, как например программирование на текстовом файле. Этим он хотел показать, что мы как программисты должны открыть глаза к новым идеям. И выступление имел предназначенный эффект для меня: программирование это больше чем строки кода. Программирование по моему мнению — искусство общения с вычислительными устройствами. Мы такие же искусственники, как художники, скульпторы и авторы. Избавившись от предвзятости к старым методам мышления, мы можем открыть неизвестного — мы не должны знать то, что мы делаем. Именно так произошли великие открытия.

Что тогда нас ждёт в будущем? К сожалению, я могу привести лишь один конкретный пример — слияние человеческого языка в сферу программирования. Как правило, Java, Python и прочее — высшие языки. Но по моим прогнозам, это лишь начало. Мы будем достигать такого понимания программирования и лингвистики, что создается новая сфера на стыке этих дисциплин. Люди смогут общаться с компьютерами на своём языке! Насколько ли это круто!