В предложенных статьях докладчики освещали две темы – будущее программирования и худший язык программирования.

Я считаю, что более практичным и полезным является выступление Брета Виктора, поэтому с него я и начну. Автор хотел описать то, как он видит будущее программирования, показывая историю развития языков программирования. Он выделил четыре концепции, на которых в будущем будет основано программирование.

1. Прямое манипулирование дынными вместо кодирования.
2. Использовать цели и ограничения вместо процедур
3. Пространственное представление информации вместо последовательного
4. Параллельное программирование вместо последовательного

Я думаю, что первые два пункта имеют много общего и по сути дают ответы на один вопрос с разных сторон. Действительно уже сейчас есть языки программирования, в основе которых лежит декларативный подход Prolog, язык запроса к БД SQL, Фреймворк для автоматизации сборки проектов Apache Maven. Сама идея достаточно здравая – мы облегчаем процесс разрабатывания алгоритма, реализовав основную логику самостоятельно, давая возможность разработчику писать условия цели и ограничения. Это ведет к уменьшению ошибок и увеличению скорости написания программ.

На счет третьего пункта, мне кажется, я не до конца понял автора. Сейчас почти все программы представляют собой последовательность команд, идущих друг за другом, а процессор исполняет их. Если речь не идет о разделении дынных программы на секции или их представление в виде диаграмм, то да, это действительно хорошая тактика, которая лежит в основе нынешнего программирования. Это позволяет нам проще мониторить программу, избегать ошибок при её проектировании и вообще представлять её работу.

С последним пунктом я полностью согласен, нам все труднее и труднее уменьшать размер транзисторов тем самым увеличивая линейную скорость процессора. Однако нам никто не запрещает увеличивать количество процессоров и ядер на них. Использования параллельного программирования может значительно повысить производительность программы, однако это очень сложно – писать параллельно выполняющуюся программу. Я думаю, что люди должны учиться разрабатывать параллельно работающие программы, ведь это просто очередной шаг развития.

В докладе Марка Рендле, по сути, говорилось о плохих практиках в создании языка программирования. По сути, просто смешение всего плохого из всего что есть т.е. из всех языков программирования. Мне довольно сложно прокомментировать все тезисы автора, но я прокомментирую т.е. что мне больше всего запомнились.

1. Смешивание нотаций в языке. Например, использование и camelCase и snake_case. Возможно, в начале, когда вы только начинаете изучать программирование вам будет все равно какую нотацию использовать, однако со временем вы поймете, что это довольно важный аспект, так как использование корректной нотации облегчает чтение и рефакторинг кода. А вот смешивание нескольких стилей приведет к полной неразберихе.
2. Путаница с пробелами и табуляцией. Оптимально чтобы они не зависели друг от друга и были взаимозаменяемы. Иначе вы долго можете искать ошибку, которая будет в неправильной табуляции
3. Использование префикса перед переменной – вышло из Php и это действительно немного бесит, и замедляет время написания кода.
4. Также был полный бред в виде “Нумерация массива начинается с -1 и далее в отрицательную сторону”, “От четность нечетности пробелов зависят что пред нами – комментарий или код”,

“Использование номеров строк вместо ярлыков, и номера только с 42 и кратные 42” итд. Думаю, эти пункты даже пояснять не нужно, использование данных практик просто бы убило всю логику и процесс)

Подводя итоги, я считаю, что в докладе Виктора больше ценности и практического применения чем в докладе Марка. Многие из его идей действительно проходят проверку временем и создаются. По большей части они мне показались правильными и объективными. Доклад Марка был больше про набор плохих практик. Из его доклада я узнал идеи, о которых не слышал ранее. Мне он показался научно-популярным. В целом оба автора проделали большую работу при подготовке докладов и достойны вашего внимания!