

и ту же задачу, но на разных языках: язык спецификации — часто декларативный язык, язык программирования — императивный язык. Язык спецификации ориентирован на человека, язык программирования, напротив, в большей степени ориентирован на выполнение на ЭВМ и поэтому менее естествен для человека. В ряде случаев имеют смысл и неполные спецификации программ. Они могут использоваться для верификации динамических свойств программ (таких, как выход за границы массива и др.).

С позиций технологии программирования верификацию следует рассматривать как дополнение тестирования, позволяющее получить более совершенную стратегию отладки программ. При этом, с одной стороны, тестирование распространяется и на спецификации, а с другой стороны, формальная спецификация программы и ее фрагментов может эффективно использоваться в процессах тестирования, создавая избыточную информацию о данных и вычислениях, которая облегчает оценку тестовых прогонов программы. Именно интеграция тестирования и верификации обеспечивает качественно новый уровень в технологии отладки. Верификация даже небольших важных модулей программного продукта, которую уже сейчас можно достичь при вполне приемлемой трудоемкости, способна существенно повышать надежность конечного продукта.

Методы верификации могут быть применены на различных этапах разработки и при различных стратегиях конструирования программ. При этом различают несколько подходов.

1. *Аналитический подход*, при котором проводится верификация готовой программы или ее фрагмента. Программирование здесь «опережает» доказательство свойств полученной программы.

2. *Синтетический (конструктивный) подход*, при котором пошагово конструируется программа совместно с доказательством ее корректности. Спецификация отдельных фрагментов разрабатываемой программы здесь «идет впереди» соответствующих программных кодов.

3. *Верификация динамических свойств вычислений*, при которой обнаруживаются аномалии, такие, как незавершение циклов, некорректность операций (деление на нуль, выталкивание из пустого стека и др.), выход за границы массива, тупики в параллельных программах и др. Отличительной особенностью верификации этих свойств является то, что она может производиться при частичном или даже полном отсутствии спецификаций, что, несомненно, придает ей особую ценность.

4. *Верификация формальных спецификаций проблемных областей*, представляющих аксиоматизацию основных понятий этих областей.

5. *Верификация правил преобразований* для трансформационного синтеза программ.