

А. Расширяемостью, т. е. возможностью включения в язык новых понятий, определяемых пользователем (программистом). Успех формализации часто зависит от того, насколько удачен выбор понятий, характеризующих искомую проблемную область. Потребности спецификации нельзя покрыть лишь стандартными, универсальными понятиями; поэтому развитый язык спецификации должен быть открыт для последующих расширений пользователями.

Б. Модульностью — использованием средств, расчленяющих сложные спецификации на независимые части (модули), которые позволяют регулировать сложность, накапливать и применять стандартные (встроенные) модули, доступные в пределах одной или нескольких проблемных областей.

В. Иерархичностью структур сложных понятий, допускающей вложенность одних понятий в другие и рассмотрение образуемых иерархий понятий по принципу «сверху вниз». Совместно с модульностью иерархия понятий позволяет достичь ясного и удобного в работе представления спецификаций.

Г. Приближением нотаций языка спецификации к естественному языку и используемому языку программирования, что необходимо также для обеспечения ясности спецификаций и удобства работы с ними в процессе разработки программ.

Для удовлетворения всем этим требованиям изобразительные средства языка спецификации должны быть более развитые, чем обычные средства языка логики предикатов, создающие для него лишь теоретическую основу. Будем считать также, что каждый логический язык спецификации есть дополнение к некоторому базовому языку программирования, составляющему с ним общую языковую среду (лексикон).

Основным объектом логического языка спецификации является формула, выражающая утверждение (предикат) о свойствах или отношениях объектов рассматриваемой области. Формула представляется синтаксической структурой, допускаемой в языке, и, вообще говоря, более мощной, чем синтаксическая структура формул логики предикатов. Чтобы наделить формулы семантикой, в логическом языке спецификации предусматриваются формальные средства описания проблемных областей. Возможны два способа описания:

1) процедурный, вводящий модули процедур для функций и предикатов проблемной области, которые могут использоваться для контроля и диагностики программ (выполнимых утверждений о свойствах программ);

2) аксиоматический, определяющий функции и предикаты с помощью аксиом, характеризующих требуемые свойства этих функций и предикатов и используемый для формальных доказательств свойств программ (верификации программ).