

1) внешние, относящиеся к задаче независимо от способов ее реализации, т. е. обращенные к конечному пользователю или заказчику программы;

2) внутренние, относящиеся к программе, реализующей задачу, и описывающие свойства вычислений в этой программе (возможно, и для некоторых ее внутренних объектов или подзадач); эти спецификации обращены к разработчику программы и создаются разработчиком в процессе проектирования на основе внешних спецификаций.

Языки внешних спецификаций — языки спецификации задач, языки внутренних спецификаций — языки спецификации свойств программ. Языки спецификации задач должны обеспечить переход от неформальной постановки задачи (на естественном языке) к полностью формальной постановке (на некотором формальном языке спецификации). Между этими двумя крайностями лежат полуформальные (или частично формальные) постановки задач на некоторой смеси естественного языка с формальными языками, которые и обеспечивают необходимую плавность перехода. В настоящее время принято выделять в языках спецификации задач два класса: 1) языки описания требований; 2) языки функциональных спецификаций.

Языки описания требований — обычно полуформальные языки, подобные языкам блок-схем программ. Формальные понятия используются в этих языках для описания данных, их потоков, иногда и возможностей параллельного выполнения. К языкам описания требований относятся PSL, RSL, PRL, SDL и другие, отличающиеся степенью своей формализованности. Возможности этого класса языков ограничены средствами задания общих и специальных требований (в том числе на используемые ресурсы), они также приближают задачу к функциональной форме «вход-выход».

Языки функциональных спецификаций — формальные языки полного описания функции, которую должна реализовать программа или ее части. В отличие от языков программирования (которые также описывают функцию, реализующую искомую задачу) языки функциональных спецификаций — языки сверхвысокого уровня, ориентированные на описание того, что известно и что требуется получить в качестве результата. Такой характер спецификаций часто называют непроцедурным в противоположность процедурному характеру представления алгоритма решения задачи в программе. Непроцедурное задание — высокого уровня, оно свободно от деталей реализации и не навязывает ту или иную реализацию.

Например, при функциональной спецификации задачи сортировки массива важно указывать требуемый порядок между элементами отсортированного массива (на выходе) и то, что он достигается только перестановками элементов входного массива. Ме-