* 1. **Введение**

В окончательном виде любая программа представляет собой набор инструкций процессора. Все, что написано на любом языке программирования, — более удоб­ная, упрощенная запись этого набора инструкций, облегчающая написание, от­ладку и последующую модификацию программы. Чем выше уровень языка, тем в более простой форме записываются одни и те же действия. Например, для реали­зации цикла на языке ассемблера требуется записать последовательность инст­рукций, позаботившись о размещении переменных в регистрах, а в С++ или Паска­ле для этого достаточно одного оператора.

С ростом объема программы становится невозможным удерживать в памяти все детали, и становится необходимым структурировать информацию, выделять главное и отбрасывать несущественное. Этот процесс называется повышением степени абстракции программы.

Первым шагом к повышению абстракции является *использование функций,* по­зволяющее после написания и отладки функции отвлечься от деталей ее реализа­ции, поскольку для вызова функции требуется знать только ее интерфейс. Если глобальные переменные не используются, интерфейс полностью определяется заголовком функции.

Следующий шаг — *описание собственных типов данных,* позволяющих структу­рировать и группировать информацию, представляя ее в более естественном виде. Например, можно представить с помощью одной структуры все разнород­ные сведения, относящиеся к одному виду товара на складе.

Для работы с собственными типами данных требуются специальные функции. Естественно сгруппировать их с описанием этих типов данных в одном месте программы, а также по возможности отделить от ее остальных частей. При этом для использования этих типов и функций не требуется полного знания того, как именно они написаны — необходимы только описания интерфейсов. *Объедине­ние в модули* описаний типов данных и функций, предназначенных для работы с ними, со скрытием от пользователя модуля несущественных деталей, является дальнейшим развитием структуризации программы.

Все три описанных выше метода повышения абстракции преследуют цель упро­стить структуру программы, то есть *представить ее в виде меньшего количества более крупных блоков и минимизировать связи между ними.* Это позволяет управ­лять большим объемом информации и, следовательно, успешно отлаживать бо­лее сложные программы.