конструкция, объединяющая и надёжно скрывающая детали реализации определённой подзадачи

Принцип модульности нашел широкое отражение во многих областях человеческой деятельности: от космонавтики и строительства до программирования. Общепризнанно, что соблюдение принципа модульности дает ряд несомненных преимуществ как технического, так и экономического характера. Согласно этому принципу основным структурным элементом системы (изделия, продукта) является **модуль.**

В программировании модуль - это «замкнутая единица, составленная из данных и процедур, доступных вне модуля только по явному запросу на конкретное обслуживание». Данная концепция реализована в таких языках как Ада, Модула-2 и в С и С++.

«Аксиомы модульности» Коуэна, показывают какими свойствами должен обладать «хороший» модуль [16]:

блочность организации, позволяющая вызывать программные единицы из других программных единиц и/или объявлять их в других программных единицах при произвольной глубине;

синтаксическая обособленность, обеспечивающая вызов программной единицы и/или объявление ее в тексте программы без необходимости каких бы то ни было синтаксических изменений текста программы, в котором эта единица помещается;

семантическая, контекстная независимость, обеспечивающая фиксированное значение программной единицы независимо от места, в котором она объявлена и/или вызывается внутри алгоритма;

общность данных, обеспечивающая возможность взаимодействия программных единиц одного и того же языкового уровня посредством использования произвольных структур данных;

полнота определений, требующая чтобы на данном языковом уровне программная единица была самообъяснимой.

Есть более лаконичное определение «хорошего модуля» [17]:

хороший модуль снаружи проще, чем внутри;

хороший модуль проще использовать, чем построить.

Каждый модуль в системе огражден своего рода «забором», «границей», пересечь которую можно, используя четко регламентированный межмодульный интерфейс, основанный на механизме «экспорта-импорта». Предметами «экспорта-импорта» являются:

процедуры (функции);

типы данных или классы;

отдельные экземпляры данных или объекты.

В языках высокого уровня таких как Модула-2 и С/С++ реализация данного механизма выполнена путем разделения модулей на интерфейсную и реализующую части. Это обеспечивает ряд дополнительных преимуществ:

ограничивает взаимное влияние модулей друг на друга, позволяет изменять реализацию «внутренностей» модуля, не затрагивая при этом взаимодействующие с ним модули (при условии, что интерфейсная часть не изменилась);

локализует влияние потенциальных ошибок рамками модуля, препятствуя их распространению по системе, подобно тому, как переборки в трюме корабля обеспечивают его непотопляемость при отдельных пробоинах.

Существенно, что модуль - это независимая единица компиляции и хранения. Данное свойство модуля позволяет создавать библиотеки модулей, обеспечивая тем самым их многократное повторное использование. Кроме того, обеспечивается разделение работы по созданию программной системы между отдельными исполнителями.