45) В результате анализа мы получаем три модели: объектную, динамическую и функциональную. При этом объектная модель составляет базу, вокруг которой осуществляется дальнейшая разработка. При построении объектной модели в ней не всегда указываются операции над объектами, так как с точки зрения объектной модели объекты это, прежде всего, структуры данных. Поэтому разработка системы начинается с сопоставления действиям и активностям динамической модели и процессам функциональной модели операций и внесения этих операций в объектную модель. С этого начинается процесс разработки программы, реализующей поведение, которое описывается моделями, построенными в результате анализа требований к системе.

Поведение объекта задается его диаграммой состояния; каждому переходу на этой диаграмме соответствует применение к объекту одной из его операций; можно каждому событию, полученному объектом, сопоставить операцию над этим объектом, а каждому событию, посланному объектом, сопоставить операцию над объектом, которому событие было послано. Активности, запускаемой переходом на диаграмме состояний, может соответствовать еще одна (вложенная) диаграмма состояний.

Результатом этого этапа проектирования является уточненная объектная модель, содержащая все классы проектируемой программной системы, в которых специфицированы все операции над их объектами.

46) Третья фаза жизненного цикла программной системы состоит в реализации разработанных программных единиц (классов, функций, библиотек), которые в совокупности составляют разрабатываемую программную систему. Реализация каждой программной единицы может осуществляться как на объектно-ориентированном, так и на не объектно-ориентированном языке программирования, с использованием ранее разработанных программ, библиотек и баз данных.

Каждый язык программирования имеет средства для выражения трех сторон спецификации разрабатываемой прикладной системы: структур данных, потоков управления и функциональных преобразований

47) Правильно разработанные программы должны не только удовлетворять своим функциональным требованиям, но и обладать такими свойствами, как:

* повторная используемость;
* расширяемость;
* устойчивость к неправильным данным;
* системность.

Правильный объектно-ориентированный стиль программирования обеспечивает наличие этих свойств. Поясним это на примере свойства системности.

Программа обладает свойством *системности*, если она применима в качестве обобщенного оператора при "крупноблочном программировании". *Крупноблочное программирование* - это систематическое использование ранее разработанных крупных программных единиц (таких, как классы, подсистемы, или модули) при разработке новых программных систем.

Следующие рекомендации могут помочь разрабатывать классы, обладающие свойством системности.

* Методы должны быть хорошо продуманы.
* Методы должны быть понятны всем, кто их прочитает.
* Методы должны быть легко читаемы.
* Имена должны быть такими же, как и в модели.
* Методы должны быть хорошо документированы.
* Спецификации методов должны быть доступны.