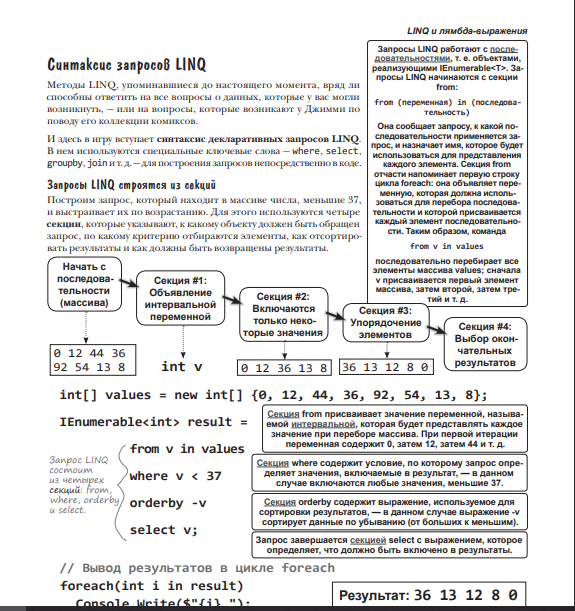
1. Инкапсуляция — объединение в единое целое данных объектов и алгоритмов обработки этих данных, позволяющее в максимальной степени изолировать объект от внешнего окружения. Означает создание объекта, который хранит внутреннюю информацию о своем состо янии в приватных полях. При - этом открытые свойства и методы используются для того, чтобы другие классы могли работать только с той частью внутренних дан ных, которую им разрешено - видеть.
2. Наследование — механизм, позволяющий описать новый класс на основе уже существующего (родительского), при этом свойства и функциональность родительского класса заимствуются новым классом. Означает, что класс или интерфейс наследует от другого класса или интерфейса.
3. Полиморфизм — возможность использовать одинаковые имена для методов, входящих в различные классы (т.е. возможность объектов с одинаковой спецификацией иметь различную реализацию).
4. Абстракция - Абстракция используется при создании модели классов, которая начинается с самых общих (или абстрактных) классов, а затем переходит к более конкретным классам, наследующим от них.
5. Композиция— более строгий тип связи. При использовании композиции объект не только является частью какого-то объекта, но и не может принадлежать другому объекту того же типа.
6. Конструктор — специальный метод, выполняемый сразу же после создания объекта. Конструктор инициализирует поля объекта — приводит объект в начальное состояние. Конструкторы могут быть как с параметрами, так и без. Конструктор без параметров называют конструктором по умолчанию, который может быть только один.
7. Свойства, Совокупность данных и методов их чтения и записи называют свойством. Свойства можно устанавливать в процессе проектирования, изменять во время выполнения программы.
8. События сигнализируют системе о том, что произошло определенное действие.
9. Перегрузки методов — это приём программирования, который позволяет разработчику в одном классе для методов с разными параметрами использовать одно и то же имя.
10. Индексаторы позволяют индексировать объекты и обращаться к данным по индексу.
11. Интерфейс представляет ссылочный тип, который может определять некоторый функционал - набор методов и свойств без реализации.
12. Структуры - пользовательский тип данных, который используется наряду с классами и может содержать какие-либо данные и методы. Структурами также являются такие типы данных как int, double и т.д.
13. Делегат— это тип, который представляет ссылки на методы с определенным списком параметров и типом возвращаемого значения.
14. Регулярного выражения, — это последовательность символов, которая определяет шаблон.
15. Коллекции предоставляют более гибкий способ работы с группами объектов. В отличие от массивов, коллекция, с которой вы работаете, может расти или уменьшаться динамически при необходимости.
16. Параметризованный класс представляет собой что-то вроде шаблона для построения других классов; шаблон, может быть, параметризован другими классами, объектами или операциями.
17. Операции со списком Класс List<T> представляет собой простейший универсальный список для хранения однородных объектов.
18. 
19. 