**Основные принципы ООП.**

    Объектно-ориентированное программирование базируется на трех важнейших принципах, придающих объектам новые свойства. Этими принципами являются ***инкапсуляция, наследование и полиморфизм***.

***1. Инкапсуляция - это объединение в единое целое данных и алгоритмов обработки этих данных***. В рамках ООП данные называются ***полями объекта (свойствами)***, а алгоритмы - ***объектными методами*** или просто ***методами***.

    Инкапсуляция позволяет в максимальной степени изолировать объект от внешнего окружения. Она существенно повышает надежность разрабатываемых программ, т.к. локализованные в объекте алгоритмы обмениваются с программой сравнительно небольшими объемами данных, причем количество и тип этих данных обычно тщательно контролируются. В результате замена или модификация алгоритмов и данных, инкапсулированных в объект, как правило, не влечет за собой плохо прослеживаемых последствий для программы в целом (в целях повышения защищенности программ в ООП почти не используются глобальные переменные).

    Другим немаловажным следствием инкапсуляции является легкость обмена объектами, переноса их из одной программы в другую.

***2. Наследование - есть свойство объектов порождать своих потомков***. Объект-потомок автоматически наследует от родителя все поля и методы, может дополнять объекты новыми полями и заменять (перекрывать) методы родителя или дополнять их.

    Принцип наследования решает проблему модификации свойств объекта и придает ООП в целом исключительную гибкость. При работе с объектами программист обычно подбирает объект, наиболее близкий по своим свойствам для решения конкретной задачи, и создает одного или нескольких потомков от него, которые "умеют" делать то, что не реализовано в родителе.

***3. Полиморфизм - это свойство родственных объектов (т.е. объектов, имеющих одного общего родителя) решать схожие по смыслу проблемы разными способами***. В рамках ООП поведенческие свойства объекта определяются набором входящих в него методов. Изменяя алгоритм того или иного метода в потомках объекта, программист может придавать этим потомкам отсутствующие у родителя специфические свойства. Для изменения метода необходимо ***перекрыть его в потомке***, то есть ***объявить в потомке одноименный метод и реализовать в нем нужные действия***. В результате в объекте-родителе и объекте-потомке будут действовать два одноименных метода, имеющие разную алгоритмическую основу и, следовательно, придающие объектам разные свойства. Это и называется полиморфизмом объектов.

    В языке программирования Turbo Pascal полиморфизм достигается не только описанным выше механизмом наследования и перекрытия методов родителя, но и их ***виртуализацией***, позволяющей родительским методам обращаться к методам потомков.

    На следующем шаге мы сформулируем задачу, которую будем решать рассмотренными средствами ООП.