

Основные принципы ООП

- Инкапсуляция
- Наследование
- Полиморфизм
- Абстракция

Инкапсуляция:

Одним из самых важных факторов при проектировании компонентов приложения является сокрытие внутренних данных компонента и деталей его реализации от других компонентов приложения и предоставление набора методов для взаимодействия с ним (API). Этот принцип является одним из четырёх фундаментальных принципов ООП и называется **инкапсуляцией**.

То есть - это когда классы не имеют прямого доступа к полям друг друга, а взаимодействие между ними осуществляется через публичные методы.

Инкапсуляция реализована с помощью **системы классов**, которые позволяют собрать информацию об объекте в одном месте; **пакетов**, которые группируют классы по какому-либо критерию, и **модификаторов доступа**, которыми можно пометить весь класс или его поле или метод.

Всего модификаторов доступа четыре:

- **public** – полный доступ к сущности (полю или методу класса) из любого пакета;
- **protected** – доступ к сущности только для классов своего пакета и наследников класса;
- **неявный модификатор по умолчанию** (при отсутствии трёх явных) – доступ к сущности только для классов своего пакета;
- **private** – доступ только внутри класса, в котором объявлена сущность.

Наследование:

Наследование является одним из важнейших принципов объектно-ориентированного программирования, поскольку оно позволяет создавать иерархические структуры объектов. Используя наследование, можно создать общий класс, который будет определять характеристики и поведение, свойственные какому-то набору связанных объектов. В дальнейшем этот класс может быть унаследован другими, более частными классами, каждый из которых будет добавлять уникальные, свойственные только ему характеристики и дополнять или изменять поведение базового класса.

По сути - это когда дочерний класс наследует свойства и методы родительского, тем самым реализуя повторное использование.

Полиморфизм:

При рассмотрении полиморфизма необходимо помнить, что этот принцип неразрывно связан с другим принципом ООП – наследованием, которое помогает реализовать полиморфизм.

Т. е. дочерние классы могут расширять и переопределять методы при решении одной и той же задачи.

Абстракция:

Теоретический прием исследования, позволяющий отвлечься от некоторых несущественных в определенном отношении свойств изучаемых явлений и выделить свойства существенные и определяющие.

Объектно-ориентированные языки, такие как Java, выводят разработку на ещё более высокий уровень абстракции: объекты в ООП по своей сути представляют собой модели понятий окружающего мира, таких как Работник, Сервер, Запись в дневнике, и выделяют только те свойства этого понятия, которые необходимы в конкретном случае для решения конкретной проблемы.

В этом и состоит абстракция: фокусировка разработчика на конкретных свойствах объекта зависит от тех задач, которые призван решать объект. Следствием такого подхода является то, что, если в императивных языках разработчику необходимо думать в терминах компьютерной логики, в объектно-ориентированных языках разработчик думает в терминах проблемной области, в которой он разрабатывает приложения.

То есть - это использование только тех характеристик объекта, которые с достаточной точностью представляют его в данной системе.