



## Урок 1

# Введение в ООП

Введение в объектно-ориентированное программирование. Классы, объекты, конструкторы, инкапсуляция, модификаторы доступа.

[Происхождение концепции ООП](#)

[Основные понятия](#)

# Происхождение концепции ООП

Размеры программ увеличивались, и требовалось привлекать больше специалистов, изыскивать дополнительные ресурсы, чтобы организовать их согласованную работу. При разработке приложений заказчик зачастую изменял функциональные требования, усложняя создание ПО.

Произошли и качественные изменения, связанные со смещением акцента в использовании компьютеров. В эпоху «больших машин» основными потребителями ПО были крупные заказчики: производственные предприятия, финансовые компании, государственные учреждения. Небольшие организации не могли себе позволить дорогостоящие вычислительные устройства.

Позже появились персональные компьютеры, которые были значительно компактнее и дешевле. Это позволило широко использовать их в малом и среднем бизнесе. Основные задачи в этой области — обработка данных и манипулирование ими, поэтому вычислительные и расчетно-алгоритмические возможности с появлением ПК отошли на второй план.

Как показала практика, традиционные методы процедурного программирования не способны справиться ни с нарастающей сложностью программ и разработки, ни с необходимостью повышения их надежности. Во второй половине 80-х появилась новая методология, способная решить этот комплекс проблем: объектно-ориентированное программирование (ООП).

Преимущества объектно-ориентированного подхода:

- Сложность программного обеспечения уменьшается;
- Надежность ПО возрастает;
- Возможна модификация отдельных компонентов ПО без изменения остальных;
- Можно повторно использовать отдельные компоненты ПО.

## Основные понятия

**Класс** определяет форму и сущность объекта и является логической конструкцией, на основе которой построен весь язык Java. Наиболее важная особенность класса в том, что он определяет новый тип данных, которым можно воспользоваться для создания объектов. Класс — это шаблон (чертеж), по которому создаются объекты, а объекты — это экземпляры класса. Для определения формы и сущности класса указываются данные, которые он должен содержать, а также код, воздействующий на них.

Переменные, определенные в классе, называются полями экземпляра — поскольку каждый объект класса содержит собственные копии этих переменных. Данные одного объекта отделены от данных другого и отличаются от них. Код содержится в теле методов. Поля экземпляра и методы, определенные в классе, — это члены класса. В большинстве классов действия над полями осуществляются через методы, определенные в этих классах.

Вторая основная структурная единица в ООП — **объект**. Это мыслимая или реальная сущность, обладающая определенным поведением и отличительными характеристиками, важная в предметной области. Каждый объект имеет состояние и обладает уникальной идентичностью. Об объекте говорят, что он является экземпляром класса, то есть построен по «чертежу», созданному при описании класса.

**Состояние (state)** — совокупный результат поведения объекта: одно из стабильных условий, в которых объект может существовать, охарактеризованное количественно. Состояние объекта всегда включает перечень его свойств (обычно статический) и их текущие значения (обычно динамические).

**Поведение (behavior)** — действия и реакции объекта, выраженные в терминах передачи сообщений и изменения состояния. Это видимая извне и воспроизводимая активность объекта.

**Identity (уникальность)** объекта состоит в том, что всегда можно определить, указывают две ссылки на один и тот же объект или на разные. При этом два объекта могут во всем быть похожими, их образ в памяти может представляться одинаковыми последовательностями байтов, но их Identity может различаться.