

Алматы 2023



**INTERNATIONAL  
UNIVERSITY**

## **Эссе: “ООП принципы”**

**Выполнено: Камбар Ажар  
SIS-2134**

Объектно-ориентированное программирование (ООП) — это подход, при котором программа рассматривается как набор объектов, взаимодействующих друг с другом. У каждого есть свойства и поведение. Если постараться объяснить простыми словами, то ООП ускоряет написание кода и делает его более читаемым.

ООП строится на четырех основных принципах: инкапсуляция, наследование, полиморфизм и абстракция.

## Инкапсуляция

Инкапсуляция — это упаковка данных и функций в одну единицу, называемую объектом. Так, например, можно представить автомобиль как объект, который содержит данные (марка, модель, цвет) и функции (едет, останавливается).

- Пользователь может взаимодействовать с объектом только через этот интерфейс. Реализуется с помощью ключевого слова: `public`.
- Пользователь не может использовать закрытые данные и методы. Реализуется с помощью ключевых слов: `private`, `protected`, `internal`.

## Преимущества инкапсуляции

1. Инкапсуляция защищает объект от нежелательного доступа клиентов.
2. Инкапсуляция позволяет получить доступ к уровню, не раскрывая сложных деталей ниже этого уровня.
3. Снижает количество человеческих ошибок.
4. Упрощает обслуживание приложения
5. Делает приложение более понятным.

## Наследование

Наследование — это способность объекта наследовать свойства и методы от другого объекта. Это помогает сократить повторение кода. Если мы продолжим пример с автомобилем, то можно представить объект «спортивный автомобиль», который наследует свойства от базового объекта «автомобиль», но при этом имеет дополнительные свойства, такие как «максимальная скорость».

## Преимущества наследования

1. Возможность повторного использования кода
2. Один суперкласс может использоваться для количества подклассов в иерархии.
3. Никаких изменений во всех базовых классах делать нельзя, просто вносите изменения только в родительский класс.
4. Наследование позволяет избежать дублирования и избыточности данных.
5. Наследование используется, чтобы избежать пространственной и временной сложности.

## Полиморфизм

Полиморфизм позволяет объектам использовать методы, которые были определены в их родительских классах, но при этом они могут иметь свою собственную реализацию этих методов. Например, у всех автомобилей есть метод «едет», но для спортивного автомобиля этот метод может иметь более сложную реализацию, например, учитывающую максимальную скорость.

## Типы полиморфизма

Два типа полиморфизма - это **динамический полиморфизм** и **статический полиморфизм**

### Динамический полиморфизм

1. Полиморфизм времени выполнения
2. Динамическое связывание
3. Привязка во время выполнения
4. Поздняя привязка
5. Переопределение метода

## Статический полиморфизм

1. Полиморфизм времени компиляции
2. Статическая привязка
3. Привязка во время компиляции
4. Раннее связывание
5. Перегрузка метода

## Абстракция

Абстракция — это процесс сокрытия деталей реализации и показа только функциональности. В контексте ООП это означает, что детали реализации методов и свойств объектов скрыты от пользователя. Если мы продолжим пример с автомобилем, то пользователю не нужно знать, как именно работает двигатель, чтобы управлять автомобилем.

## Абстрактный класс

Абстрактный класс - это просто обычный класс, у которого есть несколько особых исключений:

1. У абстрактного класса *могут* быть методы, не имеющие тела
2. Абстрактный класс должен быть на некотором уровне подклассифицирован до не абстрактного класса, прежде чем вы сможете создать экземпляр объекта.
3. Не разрешается **напрямую** создавать экземпляр объекта абстрактного класса.

