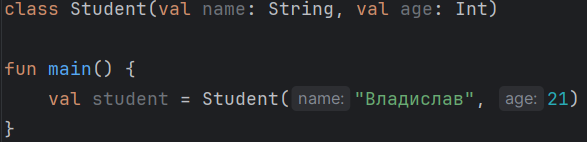
**Что такое класс и объект. Как создать объект класса?**

**Класс** – это шаблон или описание структуры данных и поведения. Он определяет, какие свойства (поля) и методы (функции) будет иметь объект, созданный на основе этого класса.

**Объект** – это экземпляр класса. Каждый объект может иметь свои уникальные значения свойств, но методы и структура данных у объектов одного класса будут одинаковыми.

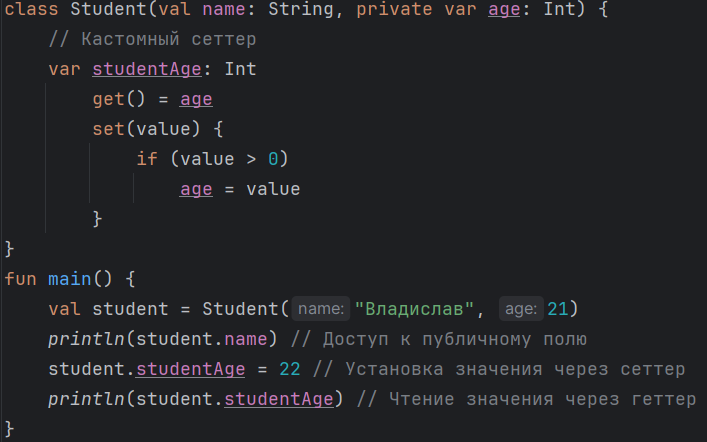
В Kotlin для создания объекта используется ключевое слово val (неизменяемая) или var (изменяемая).



**В чем заключается принцип инкапсуляции? Как получить доступ к полям объекта из внешнего класса?**

**Инкапсуляция** – это принцип ООП, согласно которому свойства и методы объекта скрыты внутри класса и могут быть недоступны из внешнего кода. Инкапсуляция помогает защитить данные от некорректного использования и напрямую управлять доступом к ним через геттеры и сеттеры.

Для получения доступа к полям объекта из внешнего класса применяются **геттеры** и **сеттеры**. В Kotlin геттеры и сеттеры генерируются автоматически для свойств класса, но можно их переопределить:



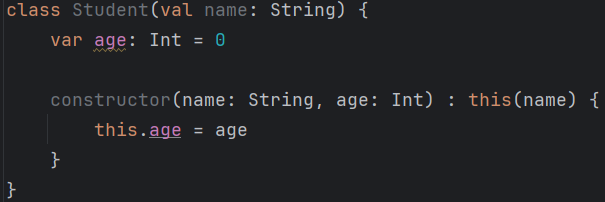
**Что такое конструктор, зачем он нужен, как описывается конструктор в произвольном классе? Разница первичного и вторичного конструктора в Kotlin?**

**Конструктор** – это специальный метод класса, который вызывается при создании объекта и инициализирует его поля. В Kotlin классы могут иметь **первичный конструктор** и несколько **вторичных конструкторов**.

**Первичный конструктор** описывается в объявлении класса:



**Вторичный конструктор** используется для добавления альтернативных способов инициализации объекта:



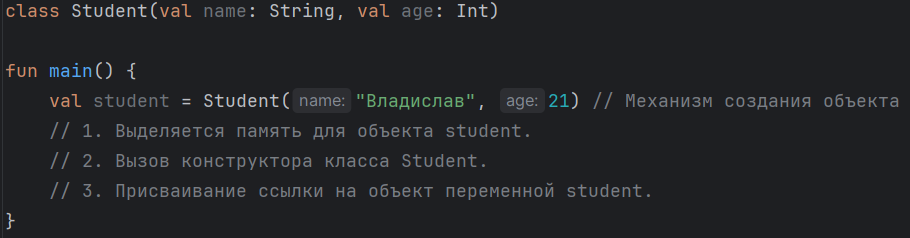
**Первичный конструктор** объявляется в заголовке класса и инициализирует все свойства.

**Вторичный конструктор** может вызывать первичный через ключевое слово this, позволяя использовать разные способы создания объекта.

**Опишите механизм создания объекта.**

Механизм создания объекта включает несколько шагов:

1. **Выделение памяти** для объекта.
2. **Вызов конструктора** класса.
3. **Возвращение ссылки** на созданный объект.



**Что такое метод класса, в чем его отличие от метода объекта. Когда введение метода класса необходимо, согласно концепциям ООП.**

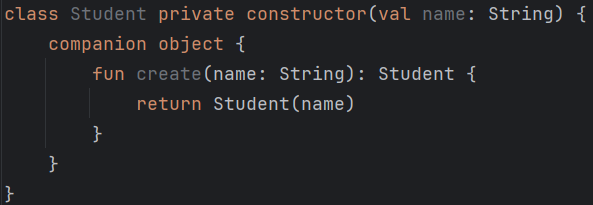
**Метод класса** принадлежит классу и не требует создания объекта для вызова. В Kotlin метод класса объявляется внутри **companion object**.

**companion-объект** определяется внутри некоторого класса и позволяет определить свойства и методы, которые будут общими для всех объектов этого класса.

**Метод объекта** – это обычный метод, который вызывается на конкретном объекте и может работать с его полями.

* **Метод класса** может вызываться без создания экземпляра.
* **Метод объекта** работает с конкретным экземпляром класса и имеет доступ к его полям и методам.

**Пример 1: Фабричный метод** (создание объекта через метод класса):



**Пример 2: Валидация данных**:

