

ОСНОВНЫЕ ПРИНЦИПЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ

Объектно-ориентированное программирование (ООП) — это парадигма программирования, которая существует уже несколько десятилетий и присутствует в популярных языках программирования, таких как Java, Python и C.

Этот метод структурирования программы использует объекты, обладающие свойствами и поведением. Java — это объектно-ориентированный язык программирования, основанный на классах, с принципом «напиши один раз, запусти где угодно».

Сегодня мы узнаем о концепциях ООП в Java. Мы рассмотрим основы синтаксиса и основные концепции. К концу вы сможете создавать классы и инициализировать объекты Java.

Обзор объектно-ориентированного программирования (ООП) в Java

Объектно-ориентированное программирование, также называемое ООП, представляет собой парадигму программирования, основанную на концепции классов и объектов. У объектов есть свои свойства и поведение. Класс подобен плану для создания объектов.

В ООП объект определяется со своими собственными свойствами. Например, скажем, наш объект — сотрудник. Этими свойствами могут быть их имя, возраст и роль. ООП позволяет легко моделировать объекты реального мира и отношения между ними.

Фактически, объекты в программе часто представляют объекты реального мира. Многие новички предпочитают использовать языки ООП, потому что они более интуитивно понятны. ООП также помогает решить проблему сложности за счёт разделения программы на разные объекты.

Четыре основных принципа ООП — это наследование, инкапсуляция, абстракция и полиморфизм. Мы рассмотрим их позже, используя Java.

Если вы знакомы с программированием на Java, вы, должно быть, заметили, что всякий раз, когда программа пишется, она записывается внутри класса, например `public class`. Вообще говоря, этот класс называется главным классом. Даже базовая программа на Java использует классы, поэтому Java является языком ООП.

Простой пример Java ООП

Мы рассмотрим это более подробно, но вот простой пример для начала. Следующий код создаёт Dogи сохраняет разные объекты.

```
public class Dog { };  
Dog GermanShepherd = new Dog();  
Dog Bulldog = new Dog();  
Dog Labrador = new Dog();
```

Объекты в Java

В реальном мире мы можем идентифицировать вещи с определёнными состояниями и поведением как объект. Например, у автомобиля есть название, цвет и марка. Мы можем классифицировать эти свойства как его атрибуты или состояние, а «вождение» — как его поведение.

В Java ООП объекты похожи. Мы создаём объект и определяем его состояния / поведение. Объекты — это экземпляры класса или сущности с определёнными данными. Думайте о классе как о схеме для создания объекта.

Например, мы могли бы иметь класс Car, экземпляр этого класса быть Truck1 с атрибутами ford, blue и used.

Примечание: объект и экземпляр — это одно и то же. Слово instance указывает отношение объекта к его классу.

У объекта есть три основных характеристики:

- Состояние: представляет данные (или значение) объекта.
- Поведение: представляет функциональность объектов, отображаемых с помощью методов.
- Идентичность: идентификация объекта реализуется с помощью уникального идентификатора. Значение используется внутри виртуальной машины Java (JVM) для идентификации каждого объекта.

Классы на Java

Класс — это группа объектов, имеющих общие свойства. Они используются для создания пользовательских типов данных. Подумайте об этом Class как о конструкторе объекта или «чертеже» для создания объектов. Мы можем использовать базовые типы данных для создания наших классов. Классы могут содержать несколько методов, переменных, конструкторов и функций.

Некоторые преимущества использования классов включают:

- Они упрощают создание сложных объектов и приложений на Java.
- Они обеспечивают возможность повторного использования кода
- Это позволяет легко обслуживать различные части приложения.

Абстрактные классы

В программировании абстракция — это процесс удаления или сокрытия атрибутов объекта или системы для поддержки более важных атрибутов. В Java абстракция может быть достигнута с помощью абстрактных классов или интерфейсов.

Абстрактный класс — это ограниченный класс, который нельзя использовать для создания объектов, и к нему можно получить доступ только через наследование. Эти типы классов объявляются с помощью abstract ключевого слова.

У абстрактного класса может быть абстрактный метод, в отличие от обычного класса. Абстрактный метод не имеет тела или определения и не должен быть частным, его объявление должно иметь:

- Идентификатор доступа.
- Ключевое слово `abstract`.
- Тип возврата.
- Название метода.
- Параметры, которые необходимо передать.
- «;» в конце объявления.

В абстрактном классе может быть всё остальное, как и в обычном классе Java, то есть конструктор, статические переменные и методы.

Конструкторы в Java

Конструкторы используются для инициализации новых состояний объекта. Они присваивают значения переменной класса при создании объекта. Конструктор объявляется с тем же именем, что и его класс, и не имеет возвращаемого типа. Рекомендуется объявить / определить его как метод первого члена.

Как и методы, конструкторы могут быть перегружены. Конструкторы отличаются от методов тем, что не имеют возвращаемого типа. Во-вторых, конструкторы вызываются только один раз при создании объекта.

Есть 2 основных типа конструкторов:

- **Конструктор по умолчанию:** он также известен как конструктор без аргументов. Это самый простой конструктор. Здесь определены значения по умолчанию для элементов данных класса, который создаёт объект с элементами данных, инициализированными значениями по умолчанию.
- **Параметризованный конструктор:** в параметризованном конструкторе мы передаём аргументы конструктору и устанавливаем их как значения наших членов данных. Они используются для установки полей класса с вашими собственными значениями.