

Java Fundamentals

Абстрактные классы и интерфейсы



Абстрактные классы и интерфейсы

Абстракция

Понятие абстракции

Абстракция в ООП – это при дание объекту характеристик, которые отличают его от всех других объектов, четко определяя его концептуальные границы.

Абстрагирование в ООП—это способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые. Соответственно, абстракция —это набор всех таких характеристик.

Абстракция

Абстрактные классы

Абстрактный класс в объектно-ориентированном программировании — это базовый класс, который не предполагает создания экземпляров через вызов конструктора напрямую, но экземпляр абстрактного класса создается неявно при построении экземпляра производного конкретного класса.

```
abstract class MyClass {  
  
}
```

Абстракция

Ключевое слово `abstract`

Ключевое слово `abstract` может использоваться с ***классами и методами.***

Абстракция

Абстрактные классы

Возможности и ограничения абстрактных классов:

- Экземпляр абстрактного класса создать нельзя через вызов конструктора напрямую, но экземпляр абстрактного класса создается неявно при построении экземпляра производного конкретного класса.
- Абстрактные классы могут содержать как абстрактные, так и не абстрактные члены.
- Не абстрактный (конкретный) класс, являющийся производным от абстрактного, должен содержать фактические реализации всех наследуемых абстрактных членов.

Абстракция

Абстрактные методы

Возможности абстрактных методов:

- Абстрактный метод является неявным виртуальным методом.
- Создание абстрактных методов допускается только в абстрактных классах
- Тело абстрактного метода отсутствует; создание метода просто заканчивается двоеточием, а после сигнатуры ставить фигурные скобки ({ }) не нужно
- Реализация предоставляется методом переопределения `override`, который является членом неабстрактного класса.

Абстракция

Интерфейсы

Интерфейс — семантическая и синтаксическая конструкция в коде программы, используемая для специфицирования услуг, предоставляемых классом или компонентом.

Интерфейс-стереотип, являющийся аналогом чистого абстрактного класса, в котором запрещена любая реализация.

Абстракция

Создание интерфейсов

Для создания интерфейсов используется ключевое слово `interface`.

```
interface MyInterface {  
  
}
```

Абстракция

Интерфейсы

Правила использования интерфейсов:

- Невозможно создать экземпляр интерфейса.
- Интерфейсы и члены интерфейсов являются абстрактными. Интерфейсы не имеют реализации в Java.
- Интерфейс может содержать только абстрактные члены.
- Члены интерфейсов автоматически являются открытыми, абстрактными, и они не могут иметь модификаторов доступа.
- Интерфейсы не могут содержать константы, поля, операторы, конструкторы экземпляров, деструкторы или вложенные типы (интерфейсы в том числе).
- Классы, которые реализуют интерфейс, должны реализовать члены этого интерфейса, указанные при его создании.

Абстракция

Интерфейсы

- Интерфейс может наследоваться от одного или нескольких базовых интерфейсов.
- Базовый класс также может реализовать члены интерфейса с помощью виртуальных членов. В этом случае производный класс может изменить поведение интерфейса путем переопределения методов.
- Если класс реализует два интерфейса, содержащих член с одинаковой сигнатурой, то при реализации этого члена в классе оба интерфейса будут использовать этот член для своей реализации.

Абстракция

Интерфейсы

Преимущество использования интерфейсов:

- Класс или структура может реализовать несколько интерфейсов (допустимо множественное наследование от интерфейсов).
- Если класс или структура реализует интерфейс, она получает только имена и сигнатуры метода.
- Интерфейсы определяют поведение экземпляров производных классов.
- Базовый класс может обладать не нужным функционалом, полученным от других его базовых классов, чего можно избежать, применяя интерфейсы.

Java Fundamentals

Q&A