Часть вторая, Kotlin

GoncharovA Follow
Jan 1 · 2 min read

Продолжение списка ответов на вопросы, собранные Веселиной. Вторая часть списка — вопросы о языке Kotlin.

. . .

16. MutableList vs List:

List представляет последовательный список элементов. List представляет неизменяемую (immutable) коллекцию, которая в основном только обеспечивает получение элементов по позиции. Изменяемые списки представлены интерфейсом MutableList. Он расширяет интерфейс List и позволяют добавлять и удалять элементы.

17. Kotlin features (что такого классного в Kotlin?):

Null-безопасность

Классы данных (Data Classes)

Sealed Classes

Функции-расширения

Умные приведения типов

В Kotlin необязательно явно указывать тип переменной

Функциональное программирование

18. Data class — в чём преимущество:

Data классы с дефолтными значениями аргументов. Больше не нужны громадные классы-модели. Просто описываете конструктор и всё — toString, getter/setter, equals/hashCode будут сгенерированы автоматически и невидимо.

19. Модификатор open в Kotlin — где используется и что даёт:

По умолчанию класс всегда наследуется от Any (аналог Object в Java) и является закрытым (final) (в Java по умолчанию открыты). В этом случае нельзя наследоваться от него. Но мы можем наследоваться от другого конкретного класса, который явно объявлен как open или abstract.Также добавьте модификатор open ко всем свойствам и методам, которые можно переопределять.

20. Модификатор object в Kotlin — где используется и что даёт:

С его помощью можно реализовать шаблон “Одиночка». Ключевое слово object одновременно объявляет класс и создаёт его экземпляр. Также можно реализовать объект-компаньон, содержащий лишь фабричные методы, а также методы, связанные с классом, но не требующие обращения к его экземпляру. К членам такого объекта можно обращаться просто по имени класса. Ещё можно использовать для записи объекта-выражения в качестве замены анонимного внутреннего класса.

21. Что такое sealed class:

Изолированные классы используются для отражения ограниченных иерархий классов, когда значение может иметь тип только из ограниченного набора, и никакой другой. Они являются, по сути, расширением enum-классов: набор значений enum типа также ограничен, но каждая enum-константа существует только в единственном экземпляре, в то время как наследник изолированного класса может иметь множество экземпляров, которые могут нести в себе какое-то состояние.

. . .

Все ответы собраны из следующих источников:

https://metanit.com/kotlin/tutorial/

https://kotlinlang.org/docs/tutorials/

https://tproger.ru/

Android Development Kotlin

Medium About Help Legal