

مفاهیم **boxing** و **unboxing** در برنامه‌نویسی به فرآیندهای تبدیل بین نوع‌های اولیه (primitive types) و نوع‌های مرجع (reference types) اشاره دارند. این مفاهیم به ویژه در زبان‌های شی‌گرا مانند **کاتلین** و **جاوا** مهم هستند.

Boxing ۱.

Boxing به فرآیند تبدیل یک نوع اولیه به نوع مرجع مربوط می‌شود. به عبارت دیگر، وقتی یک مقدار از یک نوع اولیه (مثل `Int` یا `Double`) به یک شیء (object) تبدیل می‌شود تا بتواند به عنوان یک نوع مرجع در ساختارهای داده‌ای مانند `ArrayList` یا `Collections` استفاده شود.

مثال در کاتلین:

```
```kotlin
val num: Int = 42
val boxedNum: Int? = num // boxing
...

```

در این مثال، مقدار `num` که از نوع `Int` است به نوع مرجع `Int?` تبدیل می‌شود.

### Unboxing ۲. ###

**Unboxing** به فرآیند تبدیل یک نوع مرجع به نوع اولیه مربوط می‌شود. این شامل استخراج مقدار از یک شیء یا نوع مرجع و تبدیل آن به نوع اولیه است.

**مثال در کاتلین:**

```
```kotlin
val boxedNum: Int? = 42
val unboxedNum: Int = boxedNum ?: 0 // unboxing
...

```

در این مثال، مقدار `boxedNum` که از نوع `Int?` (نوع مرجع) است به نوع اولیه `Int` تبدیل می‌شود. اگر `boxedNum` برابر `null` باشد، از مقدار پیش‌فرض `0` استفاده می‌شود.

نکات مهم

- ****نوع‌های اولیه**** در کاتلین به عنوان نوع‌های غیر مرجع (non-nullable) شناخته می‌شوند، مانند ``Int``, ``Double``, ``Boolean`` و غیره.

- ****نوع‌های مرجع**** می‌توانند ``null`` را قبول کنند و با علامت سؤال (``?``) مشخص می‌شوند، مانند ``Int?``, ``Double?``, ``Boolean?``.

عملکرد

- ****Boxing**** و ****unboxing**** ممکن است بر روی عملکرد تأثیر بگذارند، زیرا زمانی که یک نوع اولیه به نوع مرجع تبدیل می‌شود، مقداری از حافظه برای شیء جدید تخصیص می‌یابد.

- در کاتلین، استفاده از نوع‌های مرجع ممکن است به عملکرد کمتری نسبت به نوع‌های اولیه منجر شود، بنابراین در مواردی که کارایی مهم است، بهتر است از نوع‌های اولیه استفاده کنید.

نتیجه‌گیری

Boxing و **unboxing** فرآیندهای مهمی در مدیریت نوع‌ها در زبان‌های برنامه‌نویسی مانند کاتلین هستند. در کاتلین، این فرآیندها به تعامل بین نوع‌های اولیه و نوع‌های مرجع کمک می‌کنند و درک آن‌ها برای نوشتن کد کارآمد و بهینه ضروری است.