

【Kotlin】getter, setter

在 Kotlin 中，`getter` 和 `setter` 是屬性的一部分，可以自動生成或由開發者自定義。與 Java 不同，Kotlin 提供了一種更簡潔和直觀的方式來定義和使用屬性。

Java 中的 Getter 和 Setter

在 Java 中，我們通常通過編寫方法來訪問和修改私有屬性：

```
public class Person {
    private String name;
    private int age;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
}
```

Kotlin 中的屬性訪問

在 Kotlin 中，屬性有內置的 `getter` 和 `setter`，並且這些方法是自動生成的。你可以像訪問變量一樣訪問屬性。

```
class Person {
    var name: String = ""
    var age: Int = 0
}

fun main() {
    val person = Person()
    person.name = "Alice" // 使用 setter 設置值
    println(person.name) // 使用 getter 獲取值
    person.age = 30
    println(person.age)
}
```

自定義 Getter 和 Setter

如果需要自定義屬性的 `getter` 和 `setter`，可以使用以下語法：

```
class Person {
    var name: String = ""
    get() {
        println("Getter called")
        return field
    }
    set(value) {
        println("Setter called")
        field = value
    }
}
```

```
var age: Int = 0
    get() = field
    set(value) {
        field = if (value >= 0) value else throw IllegalArgumentException("Age cannot be negative")
    }
}

fun main() {
    val person = Person()
    person.name = "Alice" // Setter called
    println(person.name) // Getter called
    person.age = 30
    println(person.age)
}
```

在這個例子中：

- `field` 是一個後備字段，表示實際存儲屬性值的變量。
- 自定義的 `getter` 和 `setter` 可以在設置和獲取屬性值時執行額外的邏輯。

說明

- **自動生成的 Getter 和 Setter：**
 - 對於 `var` 屬性，Kotlin 會自動生成默認的 `getter` 和 `setter`。
 - 對於 `val` 屬性，Kotlin 只會生成 `getter`，因為它是只讀的。
- **自定義 Getter 和 Setter：**
 - 可以在屬性定義中使用 `get()` 和 `set(value)` 來自定義 `getter` 和 `setter` 的行為。
 - 可以添加額外的邏輯，例如日誌記錄、數據驗證等。
- **後備字段 `field`：**
 - 在 `getter` 和 `setter` 中使用 `field` 來引用屬性的實際存儲變量。
 - `field` 只能在屬性的 `getter` 和 `setter` 中使用。

ps.後備字段（backing field）是指屬性的實際存儲值的內部字段。當你聲明一個屬性時，Kotlin 會自動幫你生成相應的 `getter` 和 `setter` 方法，同時也會生成一個用來存儲實際數據的字段，這個字段就是後備字段。

🕒修訂版本 #2

★由 treeman 建立於 9 🕒C🕒🕒 2024 14:42:28

🔪由 treeman 更新於 31 🕒C🕒🕒 2024 10:36:13