







ING































Mc























































































































































































































































































































































TYPE-SAFE BUILDERS USING FUNCTIONAL INTERFACES WITH RECIPIENT

```
data class KotlinMeetup(val title: String, val talks: List<Talk>)  
data class Talk(val title: String, var author: String, var startTime: LocalTime, var endTime: LocalTime)
```





































































































































































































































































```
fun talk(title: String, init: Talk.() → Unit): Talk {
```



```
class KotlinMeetupBuilder(private var title: String) {  
    private val talks = mutableListOf<Talk>()  
    fun talks(vararg talks: Talk) = this.talks.addAll(talks)  
    fun build() = KotlinMeetup(title, talks)  
}
```

```
fun kotlinMeetup(title: String, init: KotlinMeetupBuilder.() → Unit): KotlinMeetup =  
    KotlinMeetupBuilder(title).apply(init).build()
```

```
fun kotlinMeetup(title: String, init: KotlinMeetupBuilder.() → Unit): KotlinMeetup =  
fun talk(title: String, init: Talk.() → Unit): Talk
```

## TYPE-SAFE BUILDERS USING FUNCTIONAL LITERALS WITH RECEIVER

```
fun kotlinMeetup(title: String, init: KotlinMeetupBuilder.() → Unit): KotlinMeetup  
fun talk(title: String, init: Talk.() → Unit): Talk
```

## TYPE-SAFE BUILDERS USING FUNCTIONAL LITERALS WITH RECEIVER