

Алгоритмизация и программирование

10.5. Kotlin DSL

Глухих Михаил Игоревич
mailto: glukhikh@mail.ru

Type-safe builders: что это?

- ▶ См. <https://kotlinlang.org/docs/reference/type-safe-builders.html>

Type-safe builders...

- ▶ Используются для создания т.н. DSL
- ▶ DSL = Domain Specific Language = предметно-ориентированный язык = какой-то ещё язык на базе «родного» (в данном случае родной язык - Kotlin)

Примеры Kotlin DSL

- ▶ Web programming
 - ktor
 - **kotlinx.html**
- ▶ UI
 - tornadofx (DSL for JavaFX UI)
 - anko (DSL for Android UI)
 - ...

Построение HTML «напрямую»

- ▶ См. Lesson 10.2
 - `List<List<String>>.convertToHtmlTable`
- ▶ `<html>` = документ
 - `<body>` = тело документа
 - `<table>` = таблица
 - `<tr>` = ряд
 - `<td>` = данные (ячейка)

Использование kotlinx.html

- ▶ См. Lesson 10.2
 - `convertToHtmlTableUsingKotlinxHtml`

Создание DSL

- ▶ Основывается на функциях-строителях вида

```
val s = buildString {  
    append(other)  
    append(another)  
}
```

Создание DSL

- ▶ Вместо того, чтобы писать

```
val sb = StringBuilder()  
sb.append(other)  
sb.append(another)  
val s = sb.toString()
```


Создание DSL

- ▶ Определение строителя

```
inline fun buildString (  
    action: StringBuilder.() -> Unit  
) : String {  
    val sb = StringBuilder()  
    sb.action()  
    return sb.toString()  
}
```

Создание DSL

- ▶ Или так

```
inline fun buildString (  
    action: StringBuilder.() -> Unit  
) : String {  
    return StringBuilder().apply(  
        action()  
    ).toString()  
}
```

Создание DSL

- ▶ Основывается на функциях-строителях вида
- ▶ Отличительный признак – наличие параметра-лямбды с получателем

Аналог для HTML

```
class Html : Tag {  
    private val children =  
        mutableListOf<Tag>()  
    fun body(init: Body.() -> Unit): Body {  
        val body = Body()  
        body.init()  
        children.add(body)  
        return body  
    }  
}
```