Kotlin(コトリン)について



佐野 尚之

Θ



この作品は、クリエイティブ・コモンズのAttribution 3.0 Unportedライセンスの下でライ センスされています。

この使用許諾条件を見るには、http://creativecommons.org/licenses/by/3.0/をチェック するか、クリエイティブ・コモンズに郵便にてお問い合わせください。

住所は: 171 Second Street, Suite 300, San Francisco, California 94105, USA です。



原作者のクレジット(氏名、作品タイトルとURL)を表示することを守れば、改変はもちろん、 営利目的での二次利用も許可される最も自由度の高いCCライセンス。

http://creativecommons.jp/



■第1版・・・2012/7/16

本ドキュメントは、オープンソースの「LibreOffice 3.5.5」を使用して作成。





・OS、開発環境などの対象バ-	-ジョン・・・・	• • • • • • •	••••6
・各ソフトウェアのインストー	ル先 および 作業フ	オルダについて	•••••7
・Kotlinについて・・・・・	• • • • • • • •		•••• 8
\cdot Sample $\cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot$	• • • • • • • •	• • • • • • • •	••••9
・参考情報・・・・・・・・・	• • • • • • • • •		• • • • • 23





■動作確認環境

Acer Aspire 1410

Windows 7 Home Premium(64bit版)

Intel Celeron processor SU2300(1.2GHz, 800MHz FSB)。8GBメモリに変更。HDD250GB。

■開発環境

IntelliJ IDEA 11 Community Edition

Java 7 Update3



本ドキュメントの指定通りの場所ではなくても問題はありま せん。別のドライブやフォルダにインストールした場合は、 ドライブ名やフォルダ名を読み替えてインストール後の設定 を行ってください。



今回はJDK7とIntelliJ IDEA 11 Community Editionが インストール済みであるという前提です。プラグイン の画面で「Kotlin」の追加を忘れずに!





- ・Java仮想マシン上で動作する。
- ・IntelliJ IDEA 11のCommunity Editionでも開発ができる。
- ・工業利用を想定して開発された新しいJava仮想マシン向けのプログラミング言語。
- ・開発したJetBrains社は、Javaの統合開発環境であるIntellJ IDEAを開発している企業。
- ・既存のJavaの資産をそのまま活用できる。
- ・文法はJavaと似ている。
- ・Java SE 8で導入されることになる機能や、Javaには導入されていない機能がすでに導入 されている。
- ・Apacheライセンス バージョン2.0に基づいてオープンソース化されている。
- ・文の末尾にセミコロンが不要、また、functionの意味のキーワードが短縮形のfunで 書ける。
- ・将来はEclipseでのサポートも予定している。





(1). InteliJ IDEAを起動し、メニューバーの「File」 – 「New Project」を選択します。









(2). 「Next」ボタンをクリックします。







(3). Project nameに「kotlin001」、「Java Module」を選択して「Next」ボタンを クリックします。

Intelliji DEA Module name: kotinu01 Ontent ror: C:#Users¥sano¥IdeaProjects¥kotlin001	W Project	Project n <u>a</u> me: Project files <u>location</u> : Project file format:	kotlin001 Ţ C:¥Users¥sano¥IdeaProjects¥kotlin001 .idea (directory based) ▼	
Module Ste location: C:¥Users¥sano¥IdeaProjects¥kotlin001	IntellijIDEA	Content root:	kotlin001 C:¥Users¥sano¥IdeaProjects¥kotlin001	
Project Description Select type Description Project Gosu Module Sou Module Sou Module Sou Module Maven Module Android Module Android Module Previous Next Enrish Cancel	NEW	Module file location:	C:¥Users¥sano¥IdeaProjects¥kotlin001	
Image: Second	project	Select type	Description	
Previous Next Einish Cancel Help		Gosu Module Gosu Module Griffon Applicat Maven Module Android Module		
			Previous Next Einish Cancel He	elp



(4). 「Next」ボタンをクリックします。

New Project	
Please specify a directory where source files for your project can be found. This path should correspond to default (root, unnamed, top level) package. Note: the program will recognize only those source files, that are located under the	is directory.
 <u>C</u>reate source directory Enter relative path to module content root (example: java¥src): 	
STC	
The following directory will be marked as a source directory:	
C:¥Users¥sano¥IdeaProjects¥Hello¥src	
Previous	N <u>e</u> xt Einish Cancel Help







(5). 「Finish」 ボタンをクリックします。

New Project	of the set of the	a per	×
New Project	Please select the desired to The will download all need Groovy	technologies. ded libraries and create Facets in project configuration. Use library [No library selected]	
ADD module			
		Previous Nect Einish Car	icel Help





(6).「src」を選択し、右クリックメニューの「New」 – 「Kotlin File」を選択します。







(7).「sample001」を入力し、「OK」ボタンを入力します。



(8). 「Set Up module 'Kotlin001' as JVM Kotlin module」をクリックします。



(9).「OK」ボタンをクリックします。









(10). 以下のコードを入力します。





(11).「Sample001.kt」を選択し、右クリックメニューの「Run 'namespace'」を 選択します。



(12). 以下のように表示されれば成功です。









(13). if (コードと実行結果)



"C:¥Program Files¥Java¥jdk1.7.0_03¥bin¥ a と りま同じ値です Process finished with exit code 0





(14). when (コードと実行結果)

```
fun main(args : Array<String>) {
    val a : String = "てすと"
    when (a) {
        "java" -> print("Java")
        "てすと" -> print("ですと")
        "PHP" -> print("PHP")
        else -> print("その他")
    }
```

```
"C:¥Program Files¥Java¥jdk1.7.0_0:
てすと
Process finished with exit code 0
```

18





(15). for (コードと実行結果)

```
🕞fun main(args : Array<String>) {
for (i in 1..9) {
         for (j in 1..9) {
             if(i * j < 10) {
                 print(" " + (i * j))
             } else {
                 print(" " + (i * j))
         println()
Ę
```

01111081cm 111001001010100100100									
	1	2	3	4	5	6	7	8	9
	2	4	6	8	10	12	14	16	18
	3	6	9	12	15	18	21	24	27
	4	8	12	16	20	24	28	32	36
	5	10	15	20	25	30	35	40	45
	6	12	18	24	30	36	42	48	54
	7	14	21	28	35	42	49	56	63
	8	16	24	32	40	48	56	64	72
	9	18	27	36	45	54	63	72	81
Pr	Process finished with exit code 0								





(16). while (実行結果は、for文のサンプルと同じ)

```
∋fun main(args : Array<String>) {
    var <u>i</u>:Int = 1
    var j:Int = 1
    // クロージャの例(clearJ(), nextI(), nextJ())
    // ローカル関数などが、その外側の関数内で宣言されている変数の値を読み書きすることができる
    fun clearJ() { j = 1 }
    fun nextI() { <u>i++</u> }
    fun nextJ() { <u>i</u>++ }
    while (i <= 9) {
        while (j <= 9) {
            if(⊥ * ⊥ < 10) {
                print(" " + (i * j))
            } else {
                print(" " + (<u>i</u> * j))
            }
           nextJ()
        clearJ()
        println()
        nextI()
```

20



(17). do-while (実行結果は、for文のサンプルと同じ)

```
]fun main(args : Array<String>) {
    var <u>i</u>:Int = 1
    var j:Int = 1
    // クロージャの例(clearJ(), nextI(), nextJ())
    // ローカル関数などが、その外側の関数内で宣言されている変数の値を読み書きすることができる
    fun clearJ() { j = 1 }
    fun nextI() { <u>i++</u> }
    fun nextJ() { j++ }
    do {
        do {
            if(i * j < 10) {
                print(" " + (\underline{i} * \underline{j}))
            } else {
                print(" " + (j * j))
           nextJ()
        } while (j <= 9)
        clearJ()
        println()
        nextI()
    } while (\underline{i} \le 9)
```





(18). クラス(コードと実行結果)

```
  fun main(args : Array<String>) {
    val p:Product = Product(1, "パソコン",1000)
    println(p.code)
    println(p.name)
    println(p.price)

  // 商品クラス
  open class Product(wkcode : Int, wkname:String, wkprice : Int) {
    open val code:Int = wkcode // 商品コード
    open val name:String = wkname // 商品名
    open val price :Int = wkprice // 商品定価
    }
```



22



Project Kotlin

http://blog.jetbrains.com/kotlin/

Kotlin言語公式サイト

http://confluence.jetbrains.net/display/Kotlin/Welcome

Kotlin - Wikipedia

http://ja.wikipedia.org/wiki/Kotlin

プログラミングKotlin(仮)(a)

https://sites.google.com/site/tarokotlin/

ことりん - Programming Language Kotlin -

http://kotlinja.wiki.fc2.com/

算譜王におれはなる!!!!

http://d.hatena.ne.jp/ngsw_taro/

