Kotlin(コトリン)について

佐野　尚之

本ドキュメントのライセンスについて

変更履歴

■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。

目次

OS、開発環境、対象バージョン

■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

各ソフトウェアのインストール先 および 作業フォルダについて

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。

注意

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！

Kotlinについて

・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。

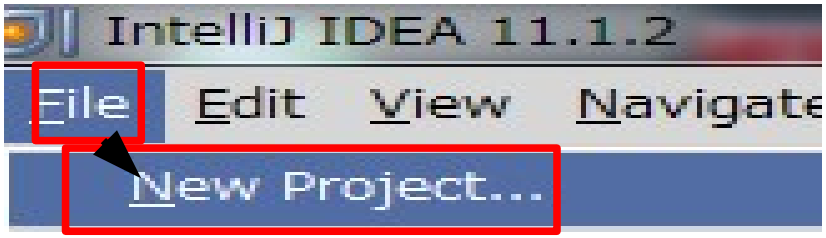
・Apacheライセンス バージョン2.0に基づいてオープンソース化されている。

・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。

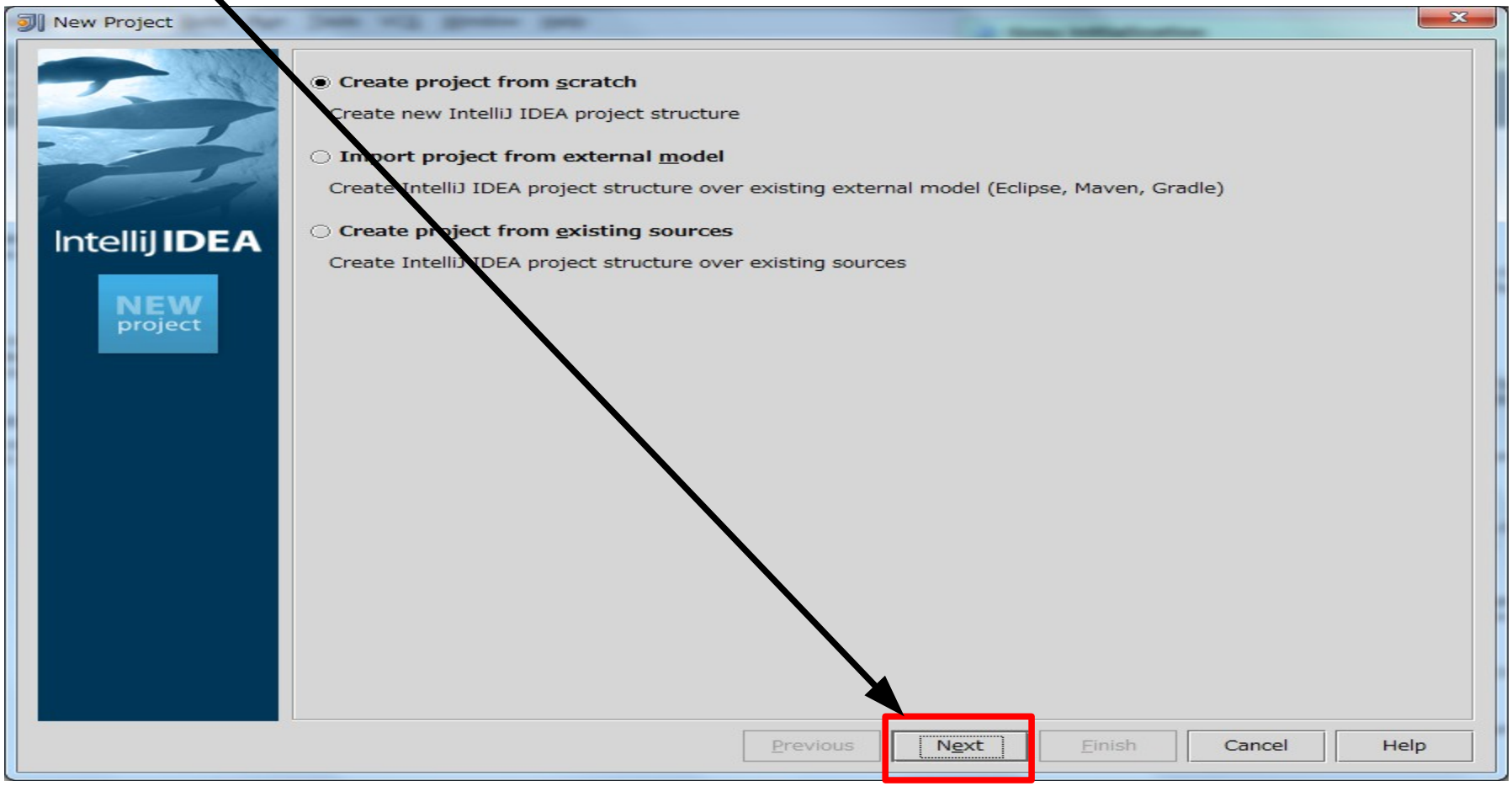
Sample (1/15)

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



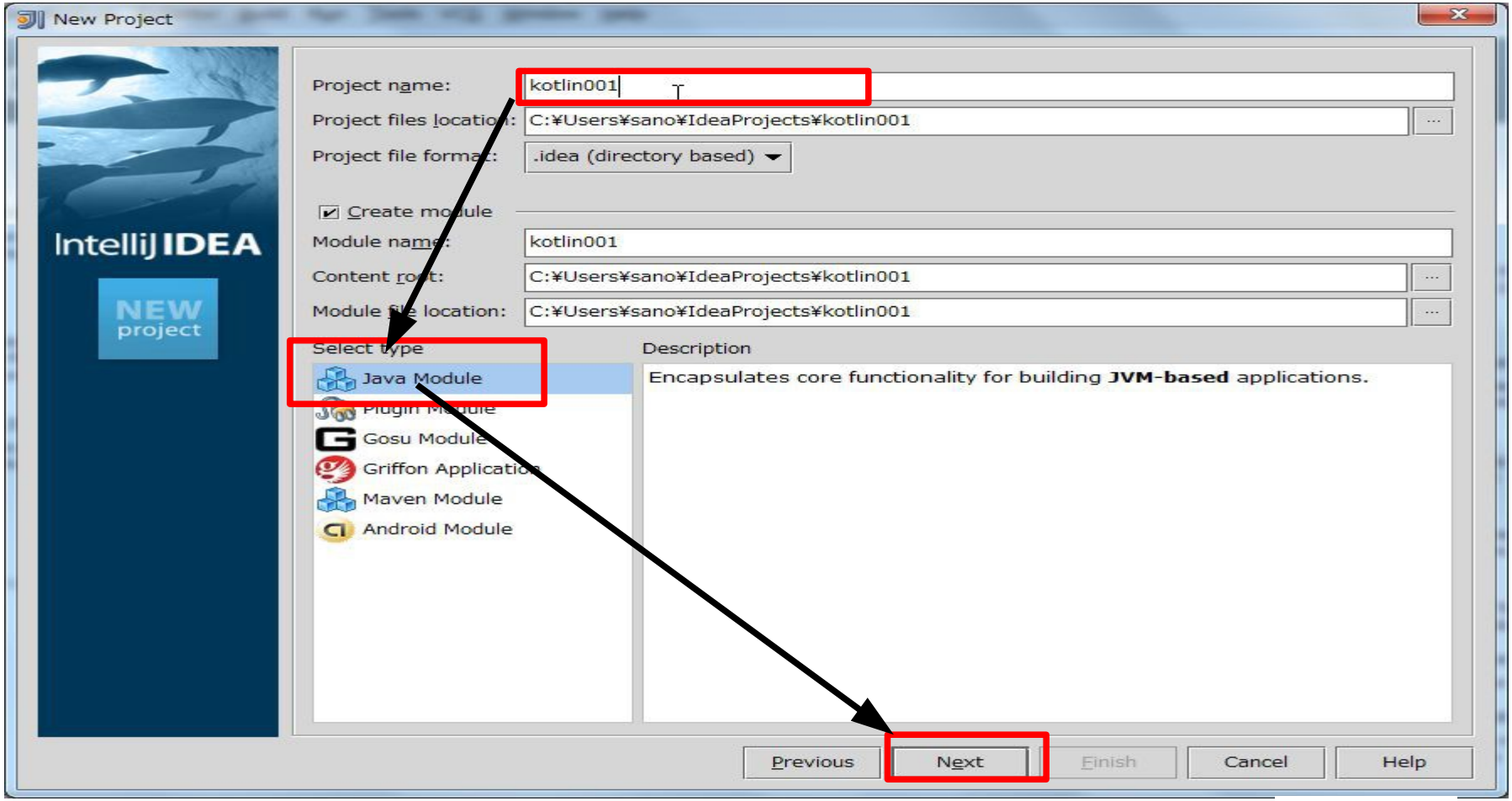
Sample (2/15)

(2).「Next」ボタンをクリックします。



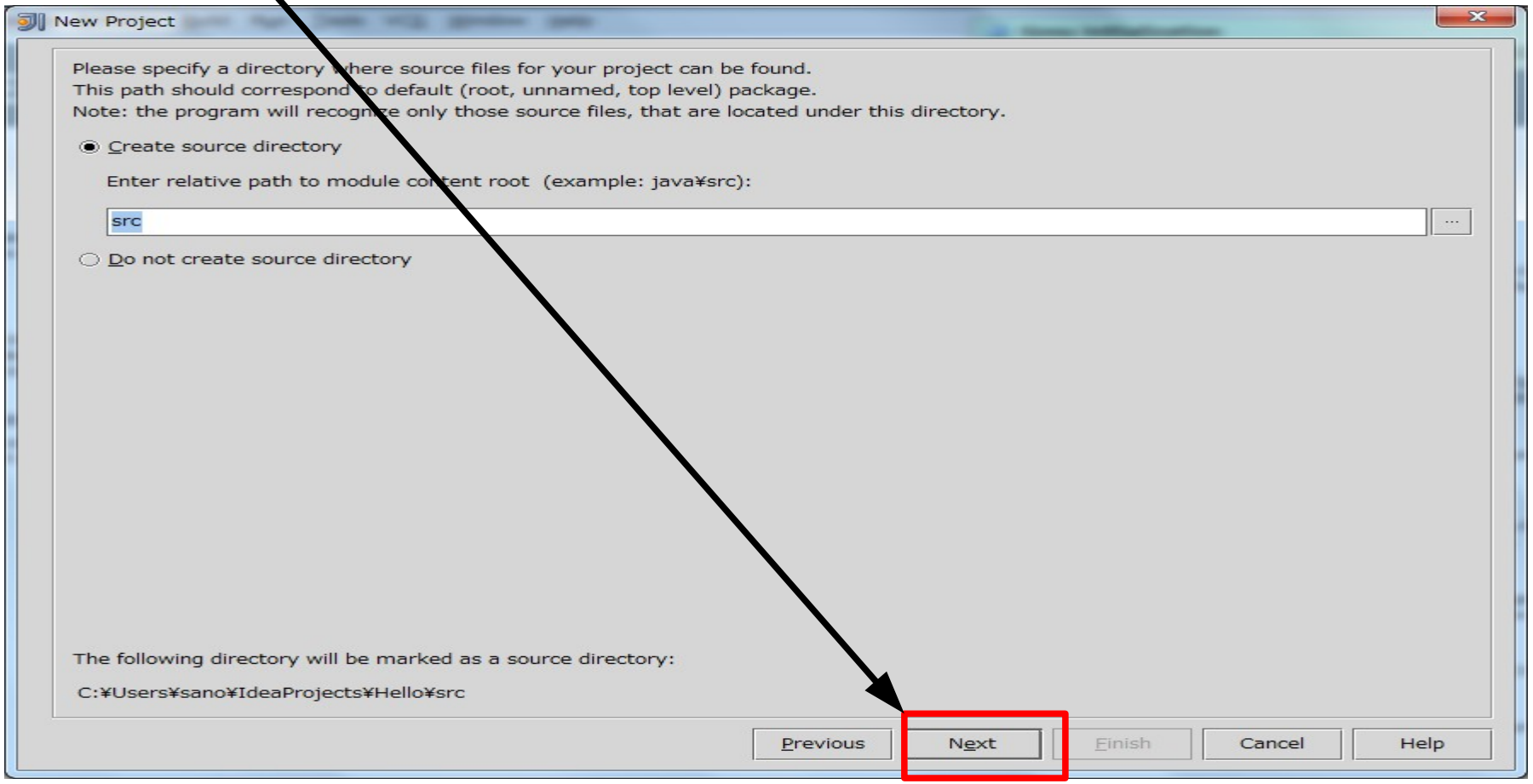
Sample (3/15)

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



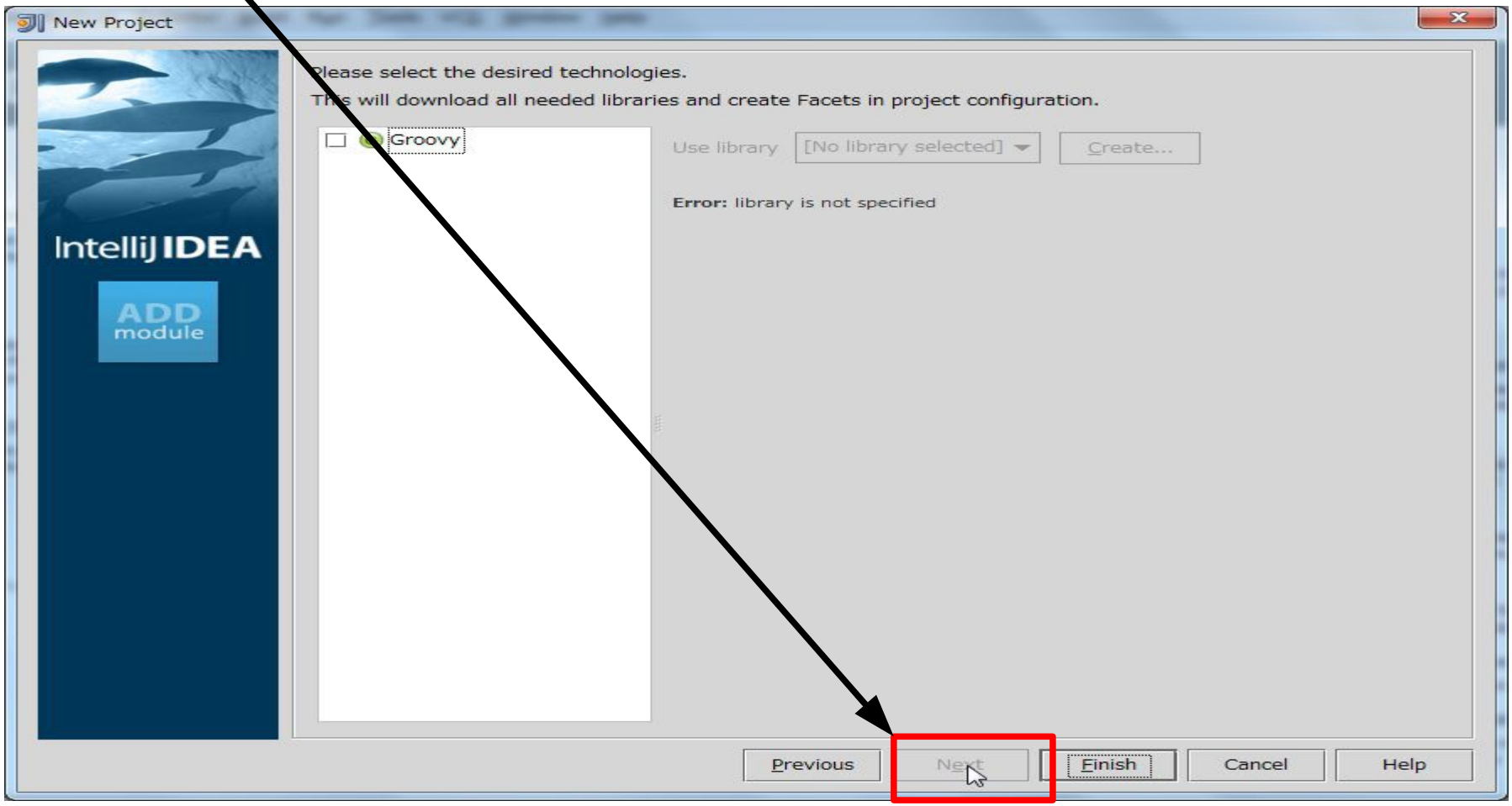
Sample (4/15)

(4).「Next」ボタンをクリックします。



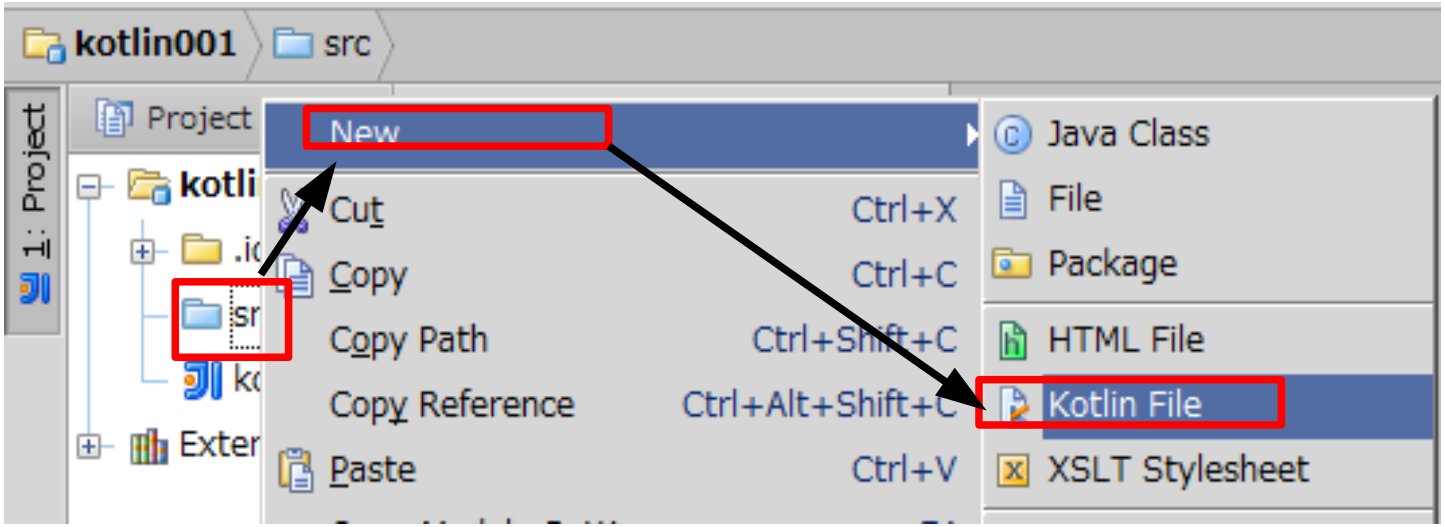
Sample (5/15)

(5).「Finish」ボタンをクリックします。



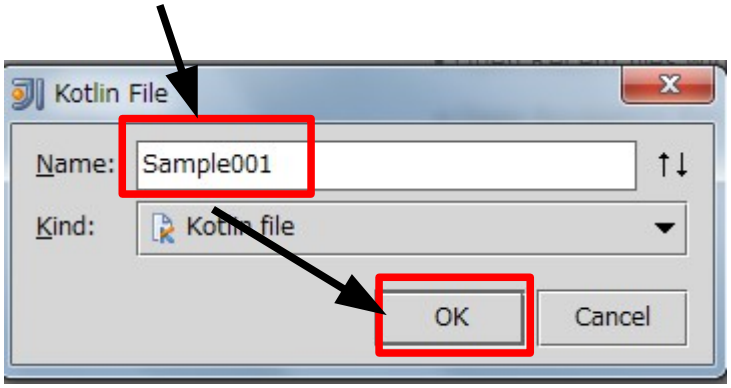
Sample (6/15)

(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。

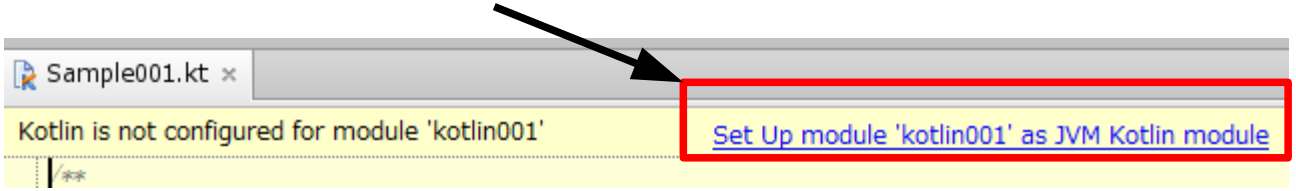


Sample (7/15)

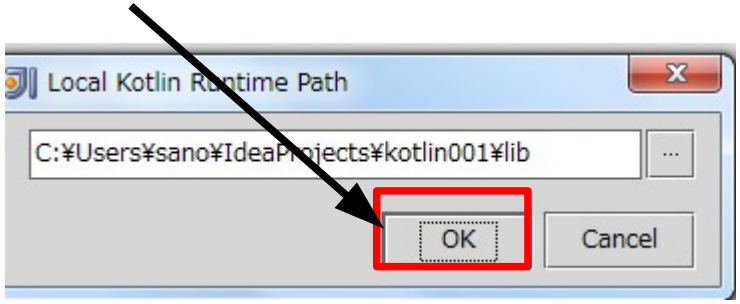
(7).「sample001」を入力し、「OK」ボタンを入力します。



(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。



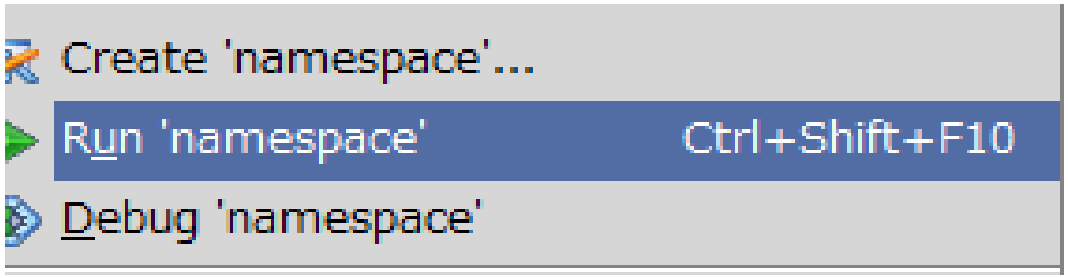
Sample (8/15)

(10). 以下のコードを入力します。

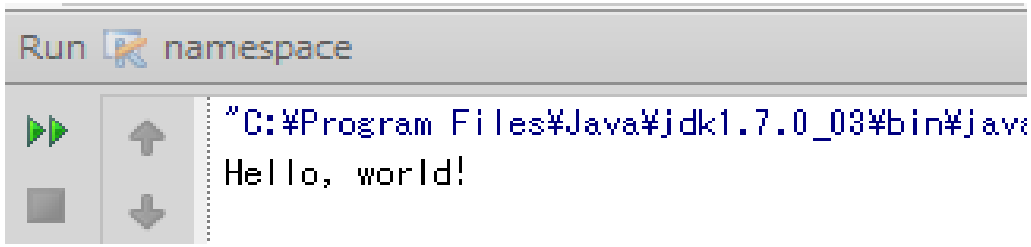
```
/**
 * Created with IntelliJ IDEA.
 * User: sano
 * Date: 12/07/16
 * Time: 15:17
 * To change this template use File | Settings | File Templates.
 */
fun main(args : Array<String>) {
    println("Hello, world!")
}
```

Sample (9/15)

(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。



Sample (10/15)

(13). if (コードと実行結果)

```
fun main(args : Array<String>) {
    val a : Int = 5
    val b : Int = 5
    if (a == b) {
        print("a と bは同じ値です")
    }
}
```

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

Sample (11/15)

(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("てすと")
        "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
```

Sample (12/15)

(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()
    }
}
```

```
  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

Process finished with exit code 0
```

Sample (13/15)

(16). while (実行結果は、for文のサンプルと同じ)

```
fun main(args : Array<String>) {
    var i: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() { i++ }
    fun nextJ() { j++ }
    while (i <= 9) {
        while (j <= 9) {
            if(i * j < 10) {
                print("  " + (i * j))
            } else {
                print(" " + (i * j))
            }
            nextJ()
        }
        clearJ()
        println()
        nextI()
    }
}
```

Sample (14/15)

(17). do-while (実行結果は、for文のサンプルと同じ)

```
fun main(args : Array<String>) {
    var i: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() { i++ }
    fun nextJ() { j++ }
    do {
        do {
            if(i * j < 10) {
                print("  " + (i * j))
            } else {
                print(" " + (i * j))
            }
            nextJ()
        } while (j <= 9)
        clearJ()
        println()
        nextI()
    } while (i <= 9)
}
```

Sample (15/15)

(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
}
```

```
"C:¥Program Files¥Java
1
パソコン
1000
```

参考情報

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）（α）

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/