

1 Compilation/exécution en ligne de commande de fichiers Kotlin

La commande pour compiler est `kotlinc <options> <source files>` avec les options suivantes :

| | |
|---------------------------------------|--|
| <code>-cp <path></code> | List of directories and JAR/ZIP archives to search for class files |
| <code>-d <directory jar></code> | Destination for generated class files |
| <code>-include-runtime</code> | Include Kotlin runtime into the resulting JAR |
| <code>-nowarn</code> | Generate no warnings |
| <code>-verbose</code> | Enable verbose logging output |
| <code>-version</code> | Display compiler version |

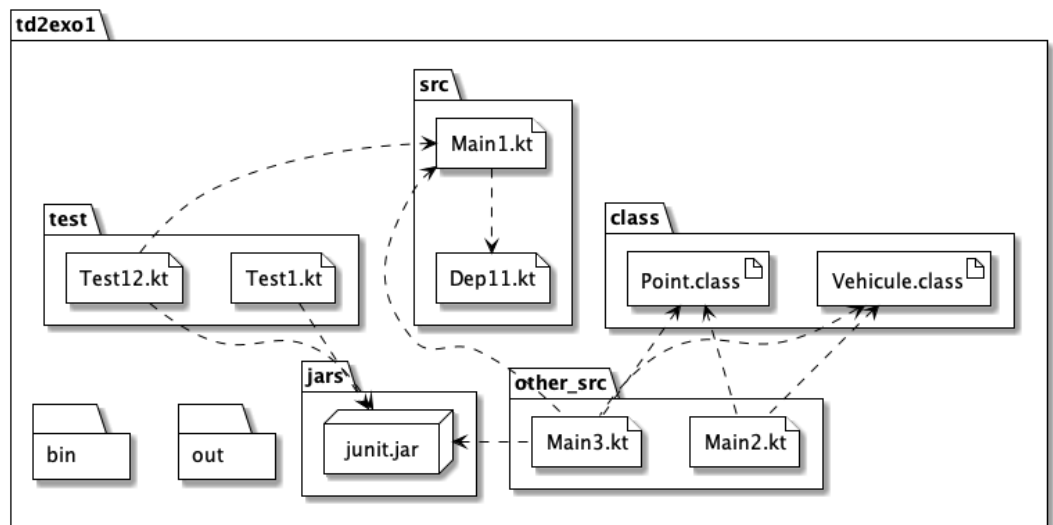
La commande pour exécuter est `kotlin <options> <command>` avec les options suivantes :

| | |
|-------------------------------|--------------------------------------|
| <code>-cp <path></code> | Paths where to find user class files |
| <code>-version</code> | Display Kotlin version |

and command is interpreted according to the following rules:

| | |
|-----------|---|
| classfile | Runs the 'main' function from the class with the given qualified name |
| app.jar | Runs the given JAR file as 'java -jar' would do |

Le dossier `td2exo1` contient les dossiers/fichiers suivants. Les flèches représentent des *dépendances* : utilisation, importation, etc. Toutes les commandes demandées seront à exécuter depuis la racine du projet (= le dossier `td2exo1`).



1. Comment compiler **uniquement** le fichier `Main1.kt` pour que le résultat soit dans `bin` ? Que devrait-il se passer ? pourquoi ?
2. Comment compiler d'abord `Dep11.kt` puis `Main1.kt` ?
3. Comment compiler les deux fichiers en une seule fois ?
4. Comment exécuter `Main1Kt` ?
5. Comment créer un JAR nommé `main1.jar` contenant `Dep11.kt` et `Main1.kt` (placer le JAR dans `out`) ?
6. Comment exécuter le JAR `main1.jar` ?
7. Comment exécuter la classe `Main1Kt` présente dans le JAR `main1.jar` ?
8. On suppose `bin` vide ; comment compiler `Main2.kt` dans `bin` ?
9. Comment exécuter `Main2Kt` ?

10. On suppose `bin` vide ; comment compiler `Test1.kt` dans `bin` ?
11. Comment compiler `Test12.kt` dans `bin` ?
12. On suppose `bin` vide ; comment compiler `Main3.kt` et produire un JAR nommé `main2.jar` dans `bin` ?
13. Comment exécuter la classe `Main3Kt` présente dans le JAR `main3.jar` ?

2 Les variables sont des références : représentation mémoire

Considérez le code Kotlin suivant :

```

1 fun main() {
2     var xx : Double = 4.2
3     var yy : Double = 42.0
4     xx = yy
5     println("${xx} == yy")
6     yy = 0.42
7     var ab = "abc"
8     var cd = "cd"
9     cd = ab
10    var ef = "ABCDEF"
11    cd = ef.substring(2,4)
12    /***** > garbage collector *****/
13    println(cd)
14    ef = ef.substring(0,3).lowercase()
15    println("${ab} == ef" et "${ab} == ef")
16    /***** > garbage collector *****/
17    var tab = arrayOf(20, 15, 2, 3)
18    tab[3] = 4
19    var sum = 1
20    for (ii in tab) {
21        sum += ii
22    }
23    println(sum)
24    var tax : Array<Int>
25    /***** > garbage collector *****/
26    tax = tab
27    tab[3] = 424242
28    tax[0] = (ab+cd+ef).length
29    tab[1] = yy.toInt()
30    yy = tab[2].toDouble()
31    /***** > garbage collector *****/
32    tax[1] = mult(tab[0], tax[2])
33    println("xx=$xx, yy=$yy, cd=$cd"
34            + "ef= $ef, sum=$sum")
35    print("tab = ")
36    for (ii in tab) {
37        print("$ii ")
38    }
39    println("");
40 }
41 fun mult(xx : Int, yy :Int) : Int {
42     var res = xx * yy
43     return res
44 }
```

14. Pouvez vous déterminer les valeurs des variables `xx`, `yy`, `cd`, `ef`, `sum` et `tab` à la fin du programme ?
 15. Dessinez schématiquement la mémoire¹ après l'exécution de la ligne 6
 16. Dessinez la mémoire après l'exécution de la ligne 10.
 17. Dessinez la mémoire après l'exécution de la ligne 18².
 18. Dessinez la mémoire après l'exécution de la ligne 21 –3ième tour de boucle.
 19. Dessinez la mémoire après l'exécution de la ligne 24.
 20. Dessinez la mémoire après l'exécution de la ligne 31.
 21. Dessinez la mémoire durant l'appel de la fonction ligne 32.
 22. Dessinez la mémoire après l'exécution de la ligne 32.
- Vérifiez votre réponse à la question 14.

1. Pile et tas mémoire représentés par un rectangle et une "patate"

2. L'indication `> garbage collector` correspond au passage du garbage collector pour l'exemple