Reflection

Αθανάσιος Ανδρούτσος

Class File (1)

- Ο μεταγλωττιστής (compiler) της Java μεταγλωττίζει σε μία ενδιάμεση μορφή που ονομάζεται Java bytecode που μπορεί να εκτελεστεί από το JVM
- Για κάθε .java file που τυπικά περιέχει μία κλάση, δημιουργείται ένα .class file. Αν ένα .java αρχείο περιλαμβάνει περισσότερες από μία κλάσεις, κάθε κλάση μεταγλωττίζεται σε ξεχωριστό αρχείο (σε αυτή την περίπτωση μία μόνο μπορεί να είναι public)

Class File (2)

- Τα .class files αντιστοιχούν και περιέχουν πληροφορίες (metadata) για ένα Java Type, δηλαδή class ή interface
- Είναι ένα stream από bytes (bytecode) και χρησιμοποιεί τύπους u1, u2, u4 για να περιγράψει unsigned bytes quantities (π.χ. 1, 2, 4, unsigned bytes)
- Για κάθε μεταβλητή ποσότητα (π.χ. πλήθος πεδίων, πλήθος μεθόδων, πλήθος παραμέτρων), υπάρχει το size, οπότε δεν χρειάζεται κάποιο padding ή κάποιος marker για να υποδείξει το τέλος των μεταβλητών δεδομένων

Class File (3)


```
ClassFile {
   u4             magic;
   u2             minor_version;
   u2             major_version;
   u2             constant_pool_count;
   cp_info        constant_pool[constant_pool_count-1];
   u2             access_flags;
   u2             this_class;
   u2             super_class;
   u2             interfaces_count;
   u2             interfaces[interfaces_count];
   u2             fields_count;
   field_info     fields[fields_count];
   u2             methods_count;
   method_info    methods[methods_count];
   u2             attributes_count;
   attribute_info attributes[attributes_count];
}
```

- Το Java class definition περιέχει όλα όσα χρειάζεται το JVM για να φορτώσει μία κλάση

Class File (4)


```
1   package testbed.ch10;
2
3   /**
4    * This is a POJO (Plain Old Java Object)
5    * class
6    */
7   public class HelloWorld {
8
9       public void sayHello() {
10          System.out.println();
11      }
12  }
```

- Αριστερά κάτω, φαίνονται τα πρώτα 8 bytes του HelloWorld.class file, με μεταγλωττιστή JDK8
- Όλα τα έγκυρα .class files ξεκινάνε με CAFEBABE στη συνέχεια το minor version είναι 00 00 και το major version JDK 1.8 (0x 00 34)
- Για Java SE 11 το major version είναι 00 37, για Java SE 21 (release Sept. 23) είναι 00 41

Offset(h)	00	01	02	03	04	05	06	07
00000000	CA	FE	BA	BE	00	00	00	34

SampleClass.java


```
1   package testbed.ch10;
2
3   public class SampleClass {
4
5       public int expr() {
6           int x = 9;
7           int y = 1;
8           int z;
9
10          z = x + y;
11          return z;
12      }
13  }
```

- Αν είχαμε ένα SampleClass όπως το αρχείο πηγαίου κώδικα αριστερά, και μεταγλωττίζαμε με JDK 8, τότε το .class file θα ήταν όπως στην επόμενη διαφάνεια

SampleClass.class

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	CA	FE	BA	BE	00	00	00	34	00	10	0A	00	04	00	0D	03
00000010	00	01	86	9F	07	00	0E	07	00	0F	01	00	06	3C	69	6E
00000020	69	74	3E	01	00	03	28	29	56	01	00	04	43	6F	64	65
00000030	01	00	0F	4C	69	6E	65	4E	75	6D	62	65	72	54	61	62
00000040	6C	65	01	00	04	74	65	73	74	01	00	03	28	29	49	01
00000050	00	0A	53	6F	75	72	63	65	46	69	6C	65	01	00	10	53
00000060	61	6D	70	6C	65	43	6C	61	73	73	2E	6A	61	76	61	0C
00000070	00	05	00	06	01	00	0B	53	61	6D	70	6C	65	43	6C	61
00000080	73	73	01	00	10	6A	61	76	61	2F	6C	61	6E	67	2F	4F
00000090	62	6A	65	63	74	00	21	00	03	00	04	00	00	00	00	00
000000A0	02	00	01	00	05	00	06	00	01	00	07	00	00	00	1D	00
000000B0	01	00	01	00	00	00	05	2A	B7	00	01	B1	00	00	00	01
000000C0	00	08	00	00	00	06	00	01	00	00	00	01	00	01	00	09
000000D0	00	0A	00	01	00	07	00	00	00	2F	00	02	00	04	00	00
000000E0	00	0B	12	02	3C	04	3D	1B	1C	60	3E	1D	AC	00	00	00
000000F0	01	00	08	00	00	00	12	00	04	00	00	00	04	00	03	00
00000100	05	00	05	00	06	00	09	00	07	00	01	00	0B	00	00	00
00000110	02	00	0C													

Offset(h)	ASCII
00000000	Ê¾º¾...4.....
00000010	..†Ÿ.....<in
00000020	it>...()V...Code
00000030	...LineNumberTab
00000040	le...test...()I.
00000050	..SourceFile...S
00000060	ampleClass.java.
00000070	.....SampleCla
00000080	ss...java/lang/O
00000090	bject.!.....
000000A0	.....
000000B0	.....*·..±....
000000C0	.....
000000D0	...../.....
000000E0	....<.=..`>.¬...
000000F0	.....
00000100	.....
00000110	...

- Μετά το version, ακολουθούν constant pool, access modifier, class name, superclass name, interfaces, fields, constructors, methods, attributes (class attributes, field attributes, κλπ).

Reflection (1)

- Το Reflection API είναι ένα σύνολο κλάσεων της Java μέσα στο java.lang.reflect package, που επιτρέπει @runtime να έχουμε πρόσβαση στο .class αρχείο
- Για παράδειγμα μπορούμε να έχουμε πρόσβαση στους Constructors και να δημιουργούμε αντικείμενα καθώς και να έχουμε πρόσβαση στα μέλη της κλάσης, πεδία και μεθόδους

Reflection (2)


```java
package gr.aueb.cf.ch14.reflect;

public class Teacher {
    private Long id;
    private String firstname;
    private String lastname;

    public Teacher() {}
    public Teacher(Long id) { this.id = id; }

    public Long getId() { return id; }
    public void setId(Long id) { this.id = id; }
    public String getFirstname() { return firstname; }
    public void setFirstname(String firstname) { this.firstname = firstname; }
    public String getLastname() { return lastname; }
    public void setLastname(String lastname) { this.lastname = lastname; }

    public void sayHello() {
        System.out.println("Hello");
    }

    public void saySomething(String message) {
        System.out.println(message);
    }
}
```

- Έστω η κλάση Teacher

Reflection (3)


```java
package gr.aueb.cf.ch14.reflect;

import ...

public class Main {

    public static void main(String[] args) {
        try {
            Class<?> clazz = Class.forName("gr.aueb.cf.ch14.reflect.Teacher");

            Constructor<?> defaultCtr = clazz.getDeclaredConstructor();
            defaultCtr.setAccessible(true);
            Teacher teacher1 = (Teacher) defaultCtr.newInstance();

            Constructor<?> longCtr = clazz.getConstructor(Long.class);
            Teacher teacher2 = (Teacher) longCtr.newInstance(10L);
```

- Με **Class.forName()** και το πλήρες όνομα της κλάσης **φορτώνουμε την κλάση**
- Με **.getConstructor()** παίρνουμε τους public constructors και με **.newInstance()** δημιουργούμε ένα νέο instance**.** Με **getDeclaredConstructor(.class)** παίρνουμε όλους (public και private) constructors. Δημιουργούμε new instance με **.newInstance(value)**

Reflection (4)


```
22          Method sayHello = clazz.getMethod("sayHello");
23          sayHello.invoke(teacher1);
24
25          Method saySomething = clazz.getMethod("saySomething", String.class);
26          saySomething.invoke(teacher2, "Coding Factory");
27
28          System.out.println();
29
30          Constructor<?>[] ctrList = clazz.getDeclaredConstructors();
31          System.out.println(Arrays.toString(ctrList));
32
33          System.out.println();
```

- Με .getMethod() παίρνουμε την μέθοδο με το συγκεκριμένο όνομα και παραμέτρους (μπορούμε να πάρουμε ονόματα μεθόδων με getDeclaredMethods() και παραμέτρους με getParameterTypes() καθώς και return type με .getReturnType()). Κάνουμε invoke με .invoke(instance, λίστα παραμέτρων)
- Με .getDeclaredConstructors() παίρνουμε μία λίστα όλων των constructors (pubic, protected, private, package private). Με getConstructors() θα παίρναμε μόνο τους public constructors

Reflection (5)


```
35              Method[] mList = clazz.getDeclaredMethods();
36              for (Method m : mList) {
37                  System.out.println("Method name: " + m.getName());
38                  int mod = m.getModifiers();
39                  System.out.println(Modifier.toString(mod));
40              }
41
42              System.out.println();
43
44              Field[] fList = clazz.getDeclaredFields();
45              for (Field f : fList) {
46                  System.out.println("Field name: " + f.getName());
47                  int mod = f.getModifiers();
48                  System.out.println(Modifier.toString(mod));
49              }
50
51          } catch (Throwable e) {
52              e.printStackTrace();
53          }
54      }
55  }
```

- Παίρνουμε ονόματα methods και fields
- Για κάθε method μπορούμε να πάρουμε το όνομα της μεθόδου με .getName() τους modifiers ενώ μπορούμε να πάρουμε και getParameterTypes(), getReturnType(), getExceptionTypes()
- Επίσης, για πεδία, getType(), getModifiers, getName, getDeclaringClass

Άσκηση

- Ορίστε μία κλάση Student (id, firstname, lastname) ως JavaBean και με Reflection δημιουργήστε ένα instance του Student με τον default constructor