

Generarea și validarea codului IBAN

~ Andronie Iustinian ~

În acest document voi explica procesul de generare și validare a codului IBAN și implementarea sa în JAVA. Codul IBAN (International Bank Account Number) este reprezentat de un șir de caractere alfanumerice de lungime 24. În acest șir se vor regăsi codul țării, caractere de verificare, codul băncii și codul contului bancar.

Structura generală unui cod IBAN din România este:

- RO – Codul țării (Primele 2 caractere);
- XX – Caracterele de verificare (2 caractere);
- YYYY – Codul băncii (Primele 4 caractere ale codului BIC);
- ZZZZZZZZZZZZZZZZZZ – Identificatorul unității teritoriale și al contului bancar (16 caractere).

Generarea Codului IBAN

Pentru generarea codului IBAN, formăm șirul de 24 de caractere, respectând structura de mai sus. În acest proces, caracterele de verificare sunt setate inițial la "00", urmând ca acestea să fie calculate ulterior utilizând algoritmul MOD 97 – 10.

Codul aferent:

```
4 public class Main {
    + Codeium: Refactor Explain Docstring
    1 usage
5   @
6   public static String genereare_IBAN_artificial()
7   {
8       //IBAN = 24 caractere
9       // Cod tara. Ex pentru Romania: RO
10      String cod_tara = "RO";
11
12      // Codul de verificare (2 pozitii) - setat la 00
13      String caracter_verificare = "00";
14
15      // Codul bancii (4 cifre)
16      // Cod Swift Raiffeisen
17      String cod_swift = "RZBRR0BUXXX";
18      String cod_banca = cod_swift.substring(0, 4);
19
20      if (cod_banca.length() != 4)
21          throw new IllegalArgumentException("Codul bancii nu are 4 caractere");
```

```

22 // Numarul contului bancar
23 String alfanumerice = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
24 Random random = new Random();
25 StringBuilder cont_bancar = new StringBuilder();
26
27 for (int i=0; i<16; i++)
28 {
29     int poz_random = random.nextInt(alfanumerice.length());
30     cont_bancar.append(alfanumerice.charAt(poz_random));
31 }
32
33 if (cont_bancar.length() != 16)
34     throw new IllegalArgumentException("Contul bancar nu are 16 caractere");
35
36 String iban = cod_tara + caracter_verificare + cod_banca + cont_bancar;
37
38 if (iban.length() != 24)
39     throw new IllegalArgumentException("Codul IBAN nu are 24 de caractere");
40
41 return iban;
42 }

```

Verificarea Codului IBAN

Validarea unui cod IBAN implică verificarea corectitudinii caracterelor de verificare utilizând algoritmul MOD 97 – 10.

Pasul 1: Se mută primele patru caractere (codul de țară și caracterele de verificare) la sfârșitul codului IBAN.

Pasul 2: Se convertesc literele în numere folosind următoarele valori: A = 10, B = 11, C = 12 ... X = 33, Y = 34, Z = 35.

Pasul 3: Se împarte numărul obținut după conversie la 97 și restul se scade din 98. Prin scăderea din 98 se obțin caracterele de verificare.

Codul aferent:

```

44 @ public static String verificare_iban(String iban)
45 {
46     if (iban.length() != 24)
47         throw new IllegalArgumentException("Codul IBAN nu are 24 de caractere");
48
49     // Pasul 1: mutam primele 4 caractere la sfarsit
50     String iban_prefix = iban.substring(0, 4);
51     String iban_verificare = iban.substring(beginIndex: 4) + iban_prefix;
52
53     System.out.println("IBAN pasul 1 verificare: " + iban_verificare);
54
55     // Pasul 2: calculare a caracterelor de verificare
56     // A = 10, B = 11, ... Z = 35
57     StringBuilder iban_conversie = new StringBuilder();
58
59     for (char c : iban_verificare.toCharArray())
60     {
61         if (Character.isDigit(c))
62             iban_conversie.append(c);
63         else if (Character.isLetter(c))
64         {
65             int code = (int) c - 55;
66             String cod = Integer.toString(code);
67             iban_conversie.append(cod);
68         }
69     }

```

```

69
70     System.out.println("IBAN pasul 2 conversie: " + iban_conversie);
71
72     // Pasul 3: aplicare algoritmul MOD 97-10
73     BigInteger nr_conversie = new BigInteger(iban_conversie.toString());
74
75     // Impartim la 97 si extragem restul
76     int rest_mod97 = nr_conversie.mod(BigInteger.valueOf(97)).intValue();
77
78     System.out.println("Restul impartii la 97: " + rest_mod97);
79
80     // Scadem restul din 98
81     int rest_din_98 = 98 - rest_mod97;
82
83     System.out.println("Restul scaderii din 98: " + rest_din_98);
84
85     // Inlocuim in IBAN codul de verificare
86     String iban_final = iban.substring(0,2) + rest_din_98 + iban.substring(beginIndex: 4);
87
88     if (iban_final.length() != 24)
89     |     throw new IllegalArgumentException("Codul IBAN final nu are 24 de caractere");
90
91     return iban_final;
92 }

```

Consola după rulare:

```

↓ Generare IBAN artificial: R000RZBRVMSUR1Y0RWGHAYDE
| IBAN pasul 1 verificare: RZBRVMSUR1Y0RWGHAYDER000
| IBAN pasul 2 conversie: 27351127312228302713402732161710341314272400
| Restul impartii la 97: 53
| Restul scaderii din 98: 45
| IBAN final: R045RZBRVMSUR1Y0RWGHAYDE

Process finished with exit code 0

```

```

↓ Generare IBAN artificial: R000AAAA1B31007593840000
| IBAN pasul 1 verificare: AAAA1B31007593840000R000
| IBAN pasul 2 conversie: 1010101011131007593840000272400
| Restul impartii la 97: 49
| Restul scaderii din 98: 49
| IBAN final: R049AAAA1B31007593840000

Process finished with exit code 0

```