

## **A tartós adathordozón tárolt adatok végleges hozzáférhetetlenné tételéhez szükséges adattörölő kód CDV-ellenőrzése**

A Nemzeti Adó- és Vámhivatal (NAV) a SORSZIG elnevezésű nyomtatványt vezette be a végleges adattörölő kód (a továbbiakban: kód) igénylésére<sup>1</sup> a tartós adattárolásra alkalmas adathordozót értékesítő kereskedők számára. A NAV a benyújtott kérelem alapján kódot generál, amit kiad a kereskedőnek. A kereskedő a fogyasztónak átadott kódokról adatot szolgáltat a NAV-nak. Ezt az AATKOD elnevezésű adatlapon<sup>2</sup>, illetve pénztárgéppel vagy számlázóprogrammal<sup>3</sup> kell teljesítenie.

Ez a tájékoztató programozási támogatást nyújt az adatszolgáltatást pénztárgéppel, számlázóprogrammal teljesítő kereskedőknek és a fejlesztőknek az adatszolgáltatás hibátlan teljesítéséhez.

A NAV által generált kód egy 16 jegyű, kriptográfiai módszerekkel titkosított karaktorsor.

### **1. A kód felépítése**

A kód az angol ábécé nagybetűiből és számokból véletlenszerűen generált 16+3 karakter hosszúságú alfanumerikus karaktersorozat, mely:

- 15 karakter hosszúságú magból,
- 1 karakter hosszúságú CDV-ellenőrző számjegyből, és
- 3 karakternyi kódcsoportokat elválasztó kötőjelből („-„) áll.

A kódképzés további kritériuma, hogy nem tartalmazhat vizuálisan könnyen összetéveszthető („B” „8”) és („O” „0”) karaktereket.

A könnyebb olvashatóság érdekében a kódot 4 csoportban, 3 kötőjellel elválasztva képi a NAV, így a kód összesen 19 karakter hosszúságú.

Példa:

**QG64-R9VE-41PX-JAE9**

A könnyebb olvashatóságot biztosító kötőjelek ('-') a kódnak nem részei, így a felhasználását tekintve a fenti példa egyenértékű a QG64R9VE41PXJAE9 kóddal.

---

<sup>1</sup>Az adatok végleges hozzáférhetetlenné tételét lehetővé tevő alkalmazás biztosításával kapcsolatos eljárási szabályok meghatározásáról szóló 726/2020. (XII. 31.) Korm. rendelet 5. §-a alapján.

<sup>2</sup>[AATKOD, adatszolgáltatás az adattörölő kódokról - Nemzeti Adó- és Vámhivatal \(gov.hu\).](#)

<sup>3</sup>[Adatszolgáltatás az adattörölő kódról pénztárgéppel, számlázóprogrammal - Nemzeti Adó- és Vámhivatal \(gov.hu\).](#)

## 2. A kód CDV-ellenőrzése

A kód ellenőrzésekor az alábbi példa szerint kell eljárni:

A példában szereplő kód legyen: UH59-7VGH-G3UE-ITA6

1. lépés: a kötőjelek eltávolítása: UH597VGHG3UEITA6
2. lépés: ASCII-konverzió a kód első 15 karakterén:  
UH597VGHG3UEITA → 857253575586717271518569738465
3. lépés: kiegészítjük a CDV-ellenőrző kóddal:  
8572535755867172715185697384656
4. lépés: elvégezhető a Hans Peter Luhn<sup>4</sup> féle CDV-ellenőrző szám előállítása.

A mellékletben mintafüggvény található a CDV-ellenőrzéshez.

**Nemzeti Adó- és Vámhivatal**

---

<sup>4</sup> <https://hu.wikipedia.org/wiki/Luhn-formula>.

## Melléklet: mintafüggvény CDV-ellenőrzésre (java)

```
package hu.gov.nav.cdv;

public class Cdvcheck {
    /**
     * @param args
     */
    public static void main(String[] args) {
        try {
            // 1. lépés: a kötőjelek eltávolítása
            String input = args[0].replace("-", "");
            String asciiString = "";

            // 2. lépés: ASCII konverzió a kód első 15 karakterén:
            for (int i = 0; i < input.length() - 1; i++) {
                asciiString = asciiString + (int) input.charAt(i);
            }
            // 3. lépés: kiegészítjük a CDV ellenőrző kóddal:
            asciiString = asciiString + input.charAt(input.length() - 1);

            // 4. lépés: elvégezhető a Luhn ellenőrzés
            System.out.println(luhnEll(asciiString));

            // Hibakezelés
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private static int luhnEll(String p_asciiString) {
        int asciiStringLength = p_asciiString.length();
        int sumOfNumbers = 0;
        boolean isSecond = false;

        for (int i = asciiStringLength - 1; i >= 0; i--) {
            int d = p_asciiString.charAt(i) - '0';

            if (isSecond == true)
                d = d * 2;

            sumOfNumbers += d / 10;
            sumOfNumbers += d % 10;

            isSecond = !isSecond;
        }

        if (sumOfNumbers % 10 == 0)
            return 1;
        else
            return 0;
    }
}
```