

HATÉKONY JAVA PROGRAMOZÁS

6. forduló

MSCI

A kategória támogatója: MSCI

RENDELKEZÉSRE ÁLLÓ IDŐ:

60:00

Ismertető a feladathoz

Felhasznált idő: 02:07/60:00

Elért pontszám: 0/18

1. feladat 0/15 pont

A következő rejtjel úgy lett kódolva, hogy az angol ABC kisbetűiből álló szöveg szavaira egyenként kiszámoltuk a Java hashCodeot. Mi volt a második szó?

114801 -290305467 3370 97 3075958 -1268786147 96891675 -795499759 3370 3117 93085691 -1206091904 1310997475
-1337619771 114801 110624917 3321751 97 98331279 -1249497167 -219784568 93111608 -25407024 3558823 93832333
114801 3433509 96727 -864995257 3707 110624778 1355153608 109627663 3125530 -1889045630 3370 3089282 3649734
3046161

Válaszok

A helyes válasz:

universe

Magyarázat

7-8 betűnél rövidebb szavakra egyszerű algoritmussal vissza lehet fejteni a bemeneti szöveget, mivel szűk intervallumból való karaktereket használunk (a-z). Ha a rövidebb szavakat visszafejtettük, akkor egy google kereséssel megtalálható az idézet, melynek a második szava a megoldás:

"the universe is a dark forest every civilization is an armed hunter stalking through the trees like a ghost gently pushing aside branches that block the path and trying to tread without sound even breathing is done with care"

Másik lehetséges megoldás:

Letöltöteni az összes angol szót tartalmazó txt fájlt innen: https://github.com/dwyl/english-words/blob/master/words_alpha.txt

Majd ezen ezt az egyszerű kódot futtatni, sorban szavanként (hashkódonként):

```
Path p = Path.of("c:\\verseny\\words_alpha.txt");
```

```
int PAR=-1268786147;
```

```
String x =Files.lines(p, StandardCharsets.UTF_8).filter( value -> value.hashCode() == PAR).findFirst().get();
```

```
System.out.println(x) ;
```

A teljes szöveg google-lel is meglett itt: <https://www.goodreads.com/quotes/7378369-the-universe-is-a-dark-forest-every-civilization-is-an>

2. feladat 0/3 pont

Java 16 JDK-ban az alábbi implementációt találjuk a String hashCode függvényére. Melyik fordulhat elő szabályos (reflection nélküli) használat során?

```
private final byte[] value;

private int hash; // Default to 0

private boolean hashIsZero; // Default to false;

public int hashCode() {
    int h = hash;
    if (h == 0 && !hashIsZero) {
        h = isLatin1() ? StringLatin1.hashCode(value)
            : StringUTF16.hashCode(value);

        if (h == 0) {
            hashIsZero = true;
        } else {
            hash = h;
        }
    }
    return h;
}
```

Válasz

- ☒ Egy String példányra a tényleges hashCode számítás (StringLatin1::hashCode vagy StringUTF16::hashCode) többször is lefut
- ☐ hashIsZero && hash != 0

- ☐ A value változása miatt hibás hash lesz eltárolva a hashCode függvény következő hívásáig, mivel a setValue metódus invalidálja a hash értékét
- ☐ Többszálú környezetben téves eredményt ad a hashCode függvény

Magyarázat

Legrosszabb esetben (konkurens elérésnél) többször számítjuk ki a hashcode-ot.

A String osztálynak nincs setValue metódusa.

[Legfontosabb tudnivalók](#)

[Kapcsolat](#)

[Versenyszabályzat](#)

[Adatvédelem](#)

© 2022 Human Priority Kft.

KÉSZÍTETTE

Megjelenés

☀ Világos ⇅