



Java nyelven!

Írjon függvényt vagy metódust, amely egy valós szám köbgyökét adja vissza! Használja az $x_{k+1} = 1/3 \cdot (2 \cdot x_k + a/x_k^2)$ sorozatot, amely a köbgyökéhez konvergál, ha $x_1 = 1$.

```
public static double cubeRoot(double a) {
    double x = 1;
    for (int i = 0; i < 100; i++) {
        x = (2 * x + a / (x * x)) / 3;
    }
    return x;
}
```

Ez a függvény egy valós szám köbgyökét adja vissza az $x_{k+1} = 1/3 \cdot (2 \cdot x_k + a/x_k^2)$ sorozatot használva. A ciklus 100 iterációt végez, de a számítások számát a szükséges pontosságtól függően lehet módosítani.



Magyarázd el!

A fenti kód egy Java függvényt definiál, amelynek neve "cubeRoot" és egy valós számot vesz fel paraméterként. A függvény visszatérési értéke egy valós szám, ami a paraméterként kapott szám köbgyöke.

A függvény belsejében egy "for" ciklust használunk, amely 100 iterációt végez. Az iterációk során a ciklusváltozó (i) értékét nem használjuk, csupán a ciklus számát szabályozza. A ciklus belsejében a köbgyök számítását végző módszert használjuk, az $x_{k+1} = 1/3 \cdot (2 \cdot x_k + a/x_k^2)$ sorozatot. Az x_{k+1} következő x értékét jelenti, x_k pedig az aktuális x értékét jelenti. Az a pedig a paraméterként kapott számot jelenti. A ciklus végén az x változót adja vissza a függvény, ami a köbgyök szám.



Fontos megemlíteni, hogy a sorozat konvergálni fog a köbgyökhöz, ha $x_1 = 1$.