

Korszerű Programozási Technikák

KisZH IV

Adott a **Coefficients** osztály, amely egy másodfokú egyenlet három egész értékű együtthatóját (a , b , c) tárolja. Az osztálynak van egy **solvable** metódusa, ami **true**-val tér vissza ha az egyenlet a valós számok halmazán megoldható, **false**-el ha nem, továbbá van egy **print** függvény, ami kiírja a konzolra az együtthatókat. A **main.cpp**-ben a **generateData** metódus véletlenszerűen generál együtthatókat és feltölt velük egy listát, a **checkSolutions** pedig megszámlálja, hogy hány egyenletnek nincs megoldása és visszatér az értékkel. A feladat szál kezeléshez és a lambda kifejezésekhez kapcsolódik: sok-sok egyenlet megoldhatóságát kell ellenőrizni és az együtthatókat kiírni párhuzamosan, több szálon. Elég csak a **main.cpp**-be dolgozni.

1. Írj egy **printCoefficients** nevű metódust, ami referenciaként megkapja a **coefficients** listát, range-based loop használatával végig megy a listát és kiírja az együtthatókat. Minden kiíratás előtt aludjon a szál 1 milliszekundumot. A **main** metódusban indítsd el a **printCoefficients** egy új szálon a megfelelő paraméterekkel. (1 pont)
2. Írj egy **solution_checker** nevű lambda függvényt, ami a **capture-list**-ben megkapja a **no_solution_count**-ot, a függvény paramétere az együttható lista referencia, visszatérési értéke pedig **void**. A függvény törzsben a **checkSolutions** segítségével állítsd be a **no_solution_count** értékét a nem megoldható egyenletek számára. A lambda függvényt indítsd el egy külön szálon a **main** függvényben a megfelelő paraméterekkel. (2 pont)
3. A **printCoefficients** és a **checkSolutions** függvény is végez kiíratást a konzolra. Egészítsd ki a két függvényt (a lambda kifejezést, szálak indítását is módosítani kell) kölcsönös kizárás segítségével, hogy a kiíratások ne keveredjenek, mindegyik külön sorban jelenjen meg. A **checkSolutions** függvényben minden iterációban aludjon a szál 1 milliszekundumot. (2 pont)