HF05 - Hello World Library

Kisvári Benedek

2024/11/05

Leadási Határidő: 2024/11/12, 23:59

Jelenleg több mint 8000 programozási nyelv létezik.[1] Szeretnénk készíteni egy könyvtárat, ami képes példaprogramok tárolására, és adott szempontok szerinti rendszerezésére. A feladat egy `HelloWorldLibrary` osztály implementációja, mely lehetővé teszi tetszőleges programok tárolását, valamint különböző szempontok szerinti gyors bejárását. A könyvtár osztályhoz implementálni kell egy `Program` típust, amiben egy program tárolható el. Program mezői:

- `name` - Az adott program neve.
- `lang` - Az adott program forráskódjának nyelve.
- `code` - A program forráskódja.
- `difficulty` - A program bonyolultsága, megértés nehézsége.
- `abstraction` - A program absztrakciós szintje.

A könyvtárban tetszőleges számú indexet szeretnénk kezelni. Ehhez egy `AbstractIndex` osztály készítése szükséges, ami a következő metódusokkal rendelkezik:

- `void addProgram(Program *p)` - Hozzáadja az indexhez a paraméterként kapott program mutatót.
- `void removeProgram(Program *p)` - Eltávolítja az indexből a programot.
- `vector<Program*> getRange(Program *begin, Program *end)` - Visszaad egy rendezett vektort azokkal az x elemekkel, amikre igaz, hogy $begin \leq x \leq end$, ahol a rendezést az adott index típusa határozza meg.

A teszteléshez szükséges egy `DifficultyIndex` és egy `AbstractionIndex` osztály készítése, amik az `AbstractIndex` osztály leszármazottai lesznek. A programnak tetszőleges számú `AbstractIndex` osztályból leszármazott indexet kell tudnia kezelni, a teszteléshez jelenleg kettőt használunk. Ezeknek az indexeknek elsődlegesen nehézség/absztrakció, majd nyelv, aztán név szerint kell növekvő sorrendben, rendezve tárolnia a könyvtárban lévő programokat. Az index célja, hogy könnyen lehessen több tulajdonság alapján keresni/szűrni az adatok között. [2]

Az indexek mutatók formájában, rendezetten tárolják a könyvtárban lévő programokat. A könyvtár osztályhoz a következő műveleteket kell implementálni:

- `void addProgram(Program p)` - Már létező programnév esetén `duplicateName` kivételt dob.
- `void removeProgram(Program p)` - Könyvtárban nem eltárolt programmal való meghívás esetén `noElementInLibrary` kivételt dob.
- `void addIndex(string indexName, AbstractIndex* idx)` - Ha van már ilyen nevű index, akkor `duplicateIndex` kivételt dob.
- `void removeIndex()` - Nemlétező index esetén `noSuchIndex` kivételt dob.
- `vector<Program*> getRangeFromIndex(string indexName, Program begin, Program end)` - Ha nemlétező programra vagy indexre hivatkozunk, akkor `noElementInLibrary` vagy `NoSuchIndex` kivételt dob. Ha a `begin` az adott index szerinti rendezés szempontjából nagyobb, akkor `invalidRange` kivételt dob.

- `int libSize()` Visszaadja az eltárolt programok számát.
- `Program operator[](string name)` Visszaadja az adott névhez tartozó programot. Nemlétező programnév esetén `noElementInLibrary` kivételt dob.

Egy program létrehozásakor meg kell adni a *nevét, nyelvét, kódját, nehézségi szintjét, absztrakciós szintjét.* A nehézségi és absztrakciós szintet nemnegatív egész számok jelölik. A nem megfelelő nehézségi - vagy absztrakciós szintű programok könyvtárba való beillesztésekor `invalidDifficulty` vagy `invalidAbstraction` kivétel dobódik. A könyvtárban két azonos nevű program nem tárolható, azonban ugyanannak a programnak két különböző nyelvű implementációja igen.

1. Példa

Adottak a következő programok:

- PythonHelloWorld
  - Name = Python Hello World
  - Language = Python
  - Code =

```
print("Hello World!")
```

  - Difficulty = 1
  - Abstraction = 10
- BrainfuckHelloWorld
  - Name = Brainfuck Hello World
  - Language = Brainfuck
  - Code =

```
>++++++++[<+++++++++>-]<.
>++++[<+++++++>-]<+.
+++++++..
+++.
>>++++++[<+++++++>-]<++.
------------.
>++++++[<+++++++++>-]<+.
<.
+++.
------.
--------.
>>>++++[<++++++++>-]<+.
```

  - Difficulty = 10
  - Abstraction = 2
- CPPHelloWorld
  - Name = CPP Hello World
  - Language = C++
  - Code =

```
#include <iostream>

using namespace std;

int main(){
```

```
    cout << "Hello World!" << endl;
    return 0;
}
```

- Difficulty = 2
- Abstraction = 6

Ha ezeket beillesztjük egy könytárba, ami tartalmaz egy `DifficultyIndex`-et, akkor az indexben a programok a következő sorrendben lesznek:
`PythonHelloWorld, CPPHelloWorld, BrainfuckHelloWorld`
Az index neve legyen `difficulty`

Ha a könyvtárhoz hozzáadunk egy `AbstractioIndex`-et, akkor abban a programok sorrendje:
`BrainfuckHelloWorld, CPPHelloWorld, PythonHelloWorld`
Az index neve legyen `abstraction`

1.1. Példa a range függvényhez

Az előző könyvtárat használva a függvényhívások és a visszaadott programok:

- `getRangeFromIndex("difficulty", CPPHelloWorld, BrainfuckHelloWorld)` → `CPPHelloWorld, BrainfuckHelloWorld`
- `getRangeFromIndex("abstraction", BrainfuckHelloWorld, BrainfuckHelloWorld)` → `BrainfuckHelloWorld`

2. Követelmények

A feladathoz használjuk az előadáson és laboron tanult adatszerkezeteket! A megoldás pontozásánál számít a kódminőség, a használt adatszerkezetek és algoritmusok, valamint az átláthatóság/dokumentáltság.

2.1. Minimum követelmények

A beadott házi feladatoknak minden alapkövetelményt teljesítenie kell, különben a feladat 0 pontos eredményű lesz.

- A kódnak a megadott flagekkel (**-Werror -Wall -Wextra -pedantic**) fordulnia kell.
- A könyvtárba való beillesztés és törlés időkomplexitása legfeljebb $O(k \cdot log_2(n))$ lehet, ahol k az indexek száma.
- Az új index beszúrásának $O(n \cdot log_2(n))$ lehet legfeljebb a komplexitása.
- A könyvtárba akkor is lehet programokat illeszteni, ha az nem tartalmaz egyetlen indexet sem.
- Minden kiadott tesztnek sikeresen le kell futnia.
- A programban nem lehet memóriaszivárgás.

2.2. További követelmények/javaslatok

- A kódba kerüljenek magyarázó kommentek, amik leírják a működését.
- Az ellenőrzés során fontos a helyes kivételkezelést is tesztelni.
- A megoldáshoz érdemes további teszteket írni, mert a kiadott tesztek nem feltétlenül fednek le minden esetet.

A feladat megoldásához lehet STL-t használni.

Hivatkozások

[1] Steve Poulson Diarmuid J Pigott, Bruce M Axtens. Hopl. https://hopl.info/. Accessed: 2024-10-13.

[2] GeeksforGeeks. Indexing in databases | set 1 - geeksforgeeks. https://www.geeksforgeeks.org/indexing-in-databases-set-1/, Jul 2016. Accessed: 2024-11-04.