

Regstřík firem I

Termín odevzdání:	31.03.2013 23:59:59
Pozdní odevzdání s penalizací:	12.05.2013 23:59:59 (Penále za pozdní odevzdání: 100.0000 %)
Hodnocení:	4.4000
Max. hodnocení:	4.0000 (bez bonusů)
Odevzdaná řešení:	3 / 20 Volné pokusy + 20 Penalizované pokusy (-2 % penalizace za každé odevzdání)
Nápovědy:	1 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu)

Úkolem je realizovat třídu CCompanyIndex, která bude implementovat databázi firemního rejstříku.

Uvažovaná databáze obsahuje jméno a adresu vlastníka a jméno a adresu firmy (firma může sídlit jinde než majitel). Předpokládáme, že všechny hodnoty jsou reprezentovány řetězci. Dále předpokládáme, že dvojice jméno vlastníka, adresa vlastníka je jednoznačnou identifikací. Tedy jedno jméno (např. Jan Novák) může být v databázi vícekrát, jedna adresa (např. Dlouhá 25) může být v databázi použita také vícekrát, ale Jan Novák bydlící v Dlouhé 25 je v databázi nejvýše jeden a vlastnící jen jednu firmu.

Jméno a adresa vlastníka je tedy v databázi unikátní. Naopak, jméno a adresa firmy unikátní není. Tedy jedna firma může být spoluvlastněna více vlastníky.

Veřejné rozhraní je uvedeno níže. Obsahuje následující:

- Konstruktor bez parametrů. Tento konstruktor inicializuje instanci třídy tak, že vzniklá instance je zatím prázdná (neobsahuje žádné záznamy).
- Destruktor. Uvolňuje prostředky, které instance alokovala.
- Metoda Add(oName, oAddr, cName, cAddr) přidá do existující databáze další záznam. Metoda vrací hodnotu true, pokud byl do záznam přidán, nebo hodnotu false, pokud přidán nebyl (protože již v databázi existoval záznam se stejným jménem a adresou majitele). Parametry oName a oAddr reprezentují jméno a adresu vlastníka, parametry cName a cAddr udávají název a adresu firmy.
- Metoda Del(oName, oAddr) odstraní záznam z databáze. Pokud byl záznam skutečně odstraněn, vrátí metoda hodnotu true. Pokud záznam neodstraní (protože neexistoval záznam majitele udaného jména a adresy), vrátí metoda hodnotu false.
- Metoda Search(oName, oAddr, cName, cAddr) zjistí, zda je v databázi uložen záznam pro zadané jméno a adresu vlastníka (oName a oAddr). Pokud je, metoda informací o firmě vrátí prostřednictvím výstupních parametrů (cName a cAddr) a návratovou hodnotou bude true. Pokud záznam vlastníka zadaného jména a adresy neexistuje, metoda vrátí false a výstupní parametry ponechá beze změny.

Odevzdávejte soubor, který obsahuje implementovanou třídu CCompanyIndex. Třída musí splňovat veřejné rozhraní podle ukázky - pokud Vámi odevzdané řešení nebude obsahovat popsání rozhraní, dojde k chybě při kompilaci. Do třídy si ale můžete doplnit další metody (veřejné nebo i privátní) a členské proměnné. Odevzdávaný soubor musí obsahovat jak deklaraci třídy (popis rozhraní) tak i definice metod, konstruktoru a destrukturu. Je jedno, zda jsou metody implementované inline nebo odděleně. Odevzdávaný soubor nesmí kromě implementace třídy CCompanyIndex obsahovat nic jiného, zejména ne vkládání hlavičkových souborů a funkci main (funkce main a vkládání hlavičkových souborů může zůstat, ale pouze obalené direktivami podmíněného překladu jako v ukázce níže).

Implementovaná třída nesmí používat datové struktury z STL (vector, list, ...). Jejich použití povede k chybě při kompilaci. Třída je testovaná v omezeném prostředí, kde je limitovaná dostupná paměť (dostačuje k uložení seznamu) a je omezena dobou běhu.

Implementovaná třída se nemusí zabývat kopírujícím konstruktorem ani přetěžováním operátoru =. V této úloze ProgTest neprovádí testy této funkčnosti.

Implementace třídy musí být efektivní z hlediska nároků na čas i nároků na paměť. Jednoduché lineární řešení nestačí (pro testovací data vyžaduje čas přes 10 minut, limit je 10s). Předpokládejte, že operace dotazu je cca 10x častější než vkládání a mazání. Proto vyhovuje řešení s polem seřazených hodnot, kde vyhledávání probíhá půlením intervalu s logaritmickou složitostí. Méně časté operace vkládání a výmaz je pak doporučeno implementovat postupem, který v logaritmickém čase najde pozici vkládaného/odebíraného prvku a pak v lineárním čase posune obsah pole.

Pro uložení hodnot alokujte pole dynamicky. Počáteční velikost volte malou (např. tisíc prvků) a velikost zvětšujte/zmenšujte podle potřeby. Při zaplnění pole není vhodné alokovat nové pole větší pouze o jednu hodnotu, takový postup má obrovskou režii na kopírování obsahu. Je rozumné pole rozšiřovat s krokem řádově tisíců prvků, nebo geometrickou řadou s kvociemem ~1.5 až 2.

Požadované veřejné rozhraní třídy:

```
#ifndef __PROGTEST__
#include <iostream>
#include <iomanip>
#include <string>
#include <cstring>
#include <cstdlib>
#include <cstdio>
using namespace std;
#endif /* __PROGTEST__ */

class CCompanyIndex
{
public:
    CCompanyIndex ( void );
    ~CCompanyIndex ( void );

    bool Add ( const string & oName,
               const string & oAddr,
               const string & cName,
               const string & cAddr );

    bool Del ( const string & oName,
               const string & oAddr );

    bool Search ( const string & oName,
                  const string & oAddr,
                  string & cName,
                  string & cAddr ) const;

    // todo
};
```

Ukázka použití třídy:

```
bool status;
string cName, cAddress;
CCompanyIndex b1;
status = b1 . Add ( "Smith", "Oak road", "ACME, Ltd.", "One ACME road" );
// status = true
status = b1 . Add ( "Brown", "Second street", "MakroHard, Inc.", "Soft street" );
// status = true
status = b1 . Add ( "Hacker", "5-th avenue", "Forks and Knives, Ltd.", "Cutlery avenue" );
// status = true
status = b1 . Add ( "Hacker", "7-th avenue", "Child toys, Inc.", "Red light district" );
// status = true
status = b1 . Search ( "Brown", "Second street", cName, cAddress );
// status = true, cName = "MakroHard, Inc.", cAddress="Soft street"
status = b1 . Search ( "Hacker", "Oak road", cName, cAddress );
// status = false
status = b1 . Search ( "Smith", "Oak road", cName, cAddress );
// status = true, cName = "ACME, Ltd.", cAddress="One ACME road"
status = b1 . Del ( "Smith", "Oak road" );
// status = true
status = b1 . Search ( "Smith", "Oak road", cName, cAddress );
// status = false

CCompanyIndex b2;
status = b2 . Add ( "Smith", "Michigan avenue", "ACME, Ltd.", "One ACME road" );
// status = true
status = b2 . Search ( "Smith", "Michigan avenue", cName, cAddress );
// status = true, cName = "ACME, Ltd.", cAddress="One ACME road"
status = b2 . Del ( "Smith", "Michigan avenue" );
// status = true
status = b2 . Search ( "Smith", "Michigan avenue", cName, cAddress );
// status = false
status = b2 . Del ( "Smith", "Michigan avenue" );
// status = false
status = b2 . Add ( "Smith", "Michigan avenue", "Forks and Knives, Ltd.", "Cutlery avenue" );
// status = true
status = b2 . Search ( "Smith", "Michigan avenue", cName, cAddress );
// status = true, cName = "Forks and Knives, Ltd.", cAddress="Cutlery avenue"
status = b2 . Add ( "Smith", "Michigan avenue", "Child toys, Inc.", "Red light district" );
```

```
// status = false
status = b2 . Add ( "Smith", "Michigan avenue", "MakroHard, Inc.", "Soft street" );
// status = false
status = b2 . Del ( "Smith", "Michigan avenue" );
// status = true
status = b2 . Search ( "Smith", "Michigan avenue", cName, cAddress );
// status = false
```

Referenční řešení

3	18.03.2013 23:00:12	Download
Stav odevzdání:		Ohodnoceno
Hodnocení:		4.4000
<ul style="list-style-type: none"> • Hodnotitel: automat <ul style="list-style-type: none"> ◦ Program zkompileován ◦ Test 'Zakladni test s parametry podle ukazky': Úspěch <ul style="list-style-type: none"> ▪ Dosaženo: 100.00 %, požadováno: 100.00 % ▪ Celková doba běhu: 0.000 s (limit: 2.000 s) ▪ Využití paměti: 12556 KiB (limit: 15314 KiB) ▪ Úspěch v závazném testu, hodnocení: 100.00 % ◦ Test 'Test nahodnymi vstupy (Add, Search)': Úspěch <ul style="list-style-type: none"> ▪ Dosaženo: 100.00 %, požadováno: 50.00 % ▪ Celková doba běhu: 0.025 s (limit: 2.000 s) ▪ Využití paměti: 13084 KiB (limit: 15314 KiB) ▪ Úspěch v závazném testu, hodnocení: 100.00 % ◦ Test 'Test nahodnymi vstupy (Add, Del, Search)': Úspěch <ul style="list-style-type: none"> ▪ Dosaženo: 100.00 %, požadováno: 50.00 % ▪ Celková doba běhu: 0.027 s (limit: 1.975 s) ▪ Využití paměti: 13084 KiB (limit: 15314 KiB) ▪ Úspěch v závazném testu, hodnocení: 100.00 % ◦ Test 'Test nahodnymi hodnotami + test prace s pameti': Úspěch <ul style="list-style-type: none"> ▪ Dosaženo: 100.00 %, požadováno: 25.00 % ▪ Celková doba běhu: 4.931 s (limit: 5.000 s) ▪ Využití paměti: 25784 KiB (limit: 31319 KiB) ▪ Úspěch v nepovinném testu, hodnocení: 100.00 % ◦ Test 'Test rychlosti (nahodne hodnoty)': Úspěch <ul style="list-style-type: none"> ▪ Dosaženo: 100.00 %, požadováno: 50.00 % ▪ Celková doba běhu: 2.977 s (limit: 10.000 s) ▪ Využití paměti: 17340 KiB (limit: 23126 KiB) ▪ Úspěch v nepovinném testu, hodnocení: 100.00 % ◦ Test 'Test rychlosti (nahodna jmena, fixni adresa)': Úspěch <ul style="list-style-type: none"> ▪ Dosaženo: 100.00 %, požadováno: 75.00 % ▪ Celková doba běhu: 2.962 s (limit: 7.023 s) ▪ Využití paměti: 17340 KiB (limit: 23126 KiB) ▪ Úspěch v nepovinném testu, hodnocení: 100.00 % ◦ Test 'Test rychlosti (fixni jmena, nahodna adresa)': Úspěch <ul style="list-style-type: none"> ▪ Dosaženo: 100.00 %, požadováno: 75.00 % ▪ Celková doba běhu: 3.319 s (limit: 4.061 s) ▪ Využití paměti: 17340 KiB (limit: 23126 KiB) ▪ Úspěch v nepovinném testu, hodnocení: 100.00 % ◦ Všechny paměťové bloky byly uvolněné - ok. ◦ Celkové hodnocení: 100.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00) • Použité nápovědy: 1 • Penalizace za vyčerpané nápovědy: Není (1 <= 2 limit) • Celkové procentní hodnocení: 100.00 % • Bonus za včasné odevzdání: 0.40 • Celkem bodů: 1.00 * (4.00 + 0.40) = 4.40 		

SW metriky:		Celkem	Průměr	Maximum	Jméno funkce
Funkce:	13	--	--	--	
Řádek kódu:	159	12.23 ± 11.13	41	main()	
Cyklomatická	32	2.46 ±	10	binaryGetIndex(const string	

složitost:

3.08

&,const string &,int,int)

2	18.03.2013 22:25:00	Download
Stav odevzdání: Ohodnoceno		
Hodnocení: 3.1504		
<ul style="list-style-type: none"> • Hodnotitel: automat <ul style="list-style-type: none"> ◦ Program zkompileován ◦ Test 'Zakladni test s parametry podle ukazky': Úspěch <ul style="list-style-type: none"> ▪ Dosaženo: 100.00 %, požadováno: 100.00 % ▪ Celková doba běhu: 0.000 s (limit: 2.000 s) ▪ Využití paměti: 12556 KiB (limit: 15314 KiB) ▪ Úspěch v závazném testu, hodnocení: 100.00 % ◦ Test 'Test nahodnymi vstupy (Add, Search)': Úspěch <ul style="list-style-type: none"> ▪ Dosaženo: 100.00 %, požadováno: 50.00 % ▪ Celková doba běhu: 0.048 s (limit: 2.000 s) ▪ Využití paměti: 13084 KiB (limit: 15314 KiB) ▪ Úspěch v závazném testu, hodnocení: 100.00 % ◦ Test 'Test nahodnymi vstupy (Add, Del, Search)': Úspěch <ul style="list-style-type: none"> ▪ Dosaženo: 96.93 %, požadováno: 50.00 % ▪ Celková doba běhu: 0.043 s (limit: 1.952 s) ▪ Využití paměti: 13084 KiB (limit: 15314 KiB) ▪ Úspěch v závazném testu, hodnocení: 96.93 % ▪ <input type="checkbox"/> Nesprávný výstup ▪ Nesprávný výstup [Zpřístupnit nápovědu (112 B)] ▪ Nesprávný výstup [Zpřístupnit nápovědu (54 B)] ▪ Nesprávný výstup [Zpřístupnit nápovědu (108 B)] ▪ Nesprávný výstup [Zpřístupnit nápovědu (108 B)] ▪ Nesprávný výstup [Zpřístupnit nápovědu (56 B)] ▪ Nesprávný výstup [Zpřístupnit nápovědu (106 B)] ▪ Nesprávný výstup [Zpřístupnit nápovědu (56 B)] ◦ Test 'Test nahodnymi hodnotami + test prace s pameti': Úspěch <ul style="list-style-type: none"> ▪ Dosaženo: 98.49 %, požadováno: 25.00 % ▪ Celková doba běhu: 4.341 s (limit: 5.000 s) ▪ Využití paměti: 25780 KiB (limit: 31319 KiB) ▪ Úspěch v nepovinném testu, hodnocení: 98.49 % ▪ Nesprávný výstup [Zpřístupnit nápovědu (54 B)] ▪ Nesprávný výstup [Zpřístupnit nápovědu (108 B)] ▪ Nesprávný výstup [Zpřístupnit nápovědu (91 B)] ▪ Nesprávný výstup [Zpřístupnit nápovědu (56 B)] ▪ Nesprávný výstup [Zpřístupnit nápovědu (112 B)] ▪ Nesprávný výstup [Zpřístupnit nápovědu (58 B)] ▪ Nesprávný výstup [Zpřístupnit nápovědu (110 B)] ▪ Nesprávný výstup [Zpřístupnit nápovědu (58 B)] ◦ Test 'Test rychlosti (nahodne hodnoty)': Úspěch <ul style="list-style-type: none"> ▪ Dosaženo: 100.00 %, požadováno: 50.00 % ▪ Celková doba běhu: 4.776 s (limit: 10.000 s) ▪ Využití paměti: 17340 KiB (limit: 23126 KiB) ▪ Úspěch v nepovinném testu, hodnocení: 100.00 % ◦ Test 'Test rychlosti (nahodna jmena, fixni adresa)': Úspěch <ul style="list-style-type: none"> ▪ Dosaženo: 100.00 %, požadováno: 75.00 % ▪ Celková doba běhu: 4.675 s (limit: 5.224 s) ▪ Využití paměti: 17340 KiB (limit: 23126 KiB) ▪ Úspěch v nepovinném testu, hodnocení: 100.00 % ◦ Test 'Test rychlosti (fixni jmena, nahodna adresa)': Program překročil přidělenou maximální dobu běhu <ul style="list-style-type: none"> ▪ Vyčerpání limitu na celý test, program násilně ukončen po: 0.553 s (limit: 0.549 s) ▪ Neúspěch v nepovinném testu, hodnocení: 75.00 % ◦ Celkové hodnocení: 71.60 % (= 1.00 * 1.00 * 0.97 * 0.98 * 1.00 * 1.00 * 0.75) • Celkové procentní hodnocení: 71.60 % • Bonus za včasné odevzdání: 0.40 • Celkem bodů: 0.72 * (4.00 + 0.40) = 3.15 		

SW metriky:

	Celkem	Průměr	Maximum	Jméno funkce
Funkce:	12	--	--	--
Řádek kódu:	172	14.33 ±	67	main()

			17.37	
Cyklomatická složitost:	31	2.58 ± 2.47	7	binarySearch(const string &,const string &,int,int)

1

18.03.2013 22:10:35

Download

Stav odevzdání:

Ohodnoceno

Hodnocení:

0.0000

Hodnotitel: automat

Program zkompileován

Test 'Zakladni test s parametry podle ukazky': Úspěch

Dosaženo: 100.00 %, požadováno: 100.00 %

Celková doba běhu: 0.000 s (limit: 2.000 s)

Využití paměti: 12556 KiB (limit: 15314 KiB)

Úspěch v závazném testu, hodnocení: 100.00 %

Test 'Test nahodnymi vstupy (Add, Search)': Program provedl neplatnou operaci a byl ukončen (Segmentation fault/Bus error/Memory limit exceeded/Stack limit exceeded)

Celková doba běhu: 0.043 s (limit: 2.000 s)

Neúspěch v závazném testu, hodnocení: 0.00 %

Celkové hodnocení: 0.00 % (= 1.00 * 0.00)

Celkové procentní hodnocení: 0.00 %

Bonus za včasné odevzdání: 0.40

Celkem bodů: 0.00 * (4.00 + 0.40) = 0.00

</