ProgTest ► BI-PA2 (19/20 LS) ► Domácí úloha 02 ► Daňový registr

Daňový registr

Termín odevzdání: 29.03.2020 23:59:59

Pozdní odevzdání s penalizací: **30.06.2020 23:59:59** (Penále za pozdní odevzdání: 100.0000 %)

Hodnocení: 7.8650

Max. hodnocení: **7.1500** (bez bonusů) Odevzdaná řešení:

1 / 20 Volné pokusy + 20 Penalizované pokusy (-2 % penalizace za každé odevzdání) 0 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu) Nápovědy:

Úkolem je realizovat třídu CTaxRegister, která bude implementovat

Pro plánované důslednější potírání daňových úniků je potřeba vybudovat databázi všech obyvatel, ve které budou archivované jejích příjmy a výdaje. Předpokládáme, že v databázi je zaveden každý občan v okamžiku jeho narození. Od té doby se mu počítají všechny příjmy a všechny výdaje. Záznam je z databáze odstraněn v okamžiku jeho úmrtí. Občan je identifikován svým jménem, adresou a číslem účtu. Číslo účtu je unikátní přes celou databázi. Jména a adresy se mohou opakovat, ale dvojice (jméno, adresa) je opět v databázi unikátní. Tedy v databázi může být mnoho jmen Jan Novak, mnoho lidí může mít adresu Hlavni 60, ale Jan Novak bydlící na adrese Hlavni 60 může být v databázi pouze jeden.

Veřejné rozhraní třídy CTaxRegister 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 Birth (Name, Addr, Acct) přidá do existující databáze další záznam. Parametry Name a Addr reprezentují jméno a adresu občana, parametr Acct udává číslo účtu. Metoda vrací hodnotu true, pokud byl záznam přidán, nebo hodnotu false, pokud přidán nebyl (protože již v databázi existoval záznam se stejným jménem a adresou, nebo záznam se stejným číslem účtu).
- Metoda Death (Name, Addr) odstraní záznam z databáze. Parametrem je jednoznačná identifikace pomocí jména a adresy. Pokud byl záznam skutečně odstraněn, vrátí metoda hodnotu true. Pokud záznam neodstraní (protože neexistoval občan s touto identifikací), vrátí metoda hodnotu false.
- Metody Income zaznamenají na účet pro daného občana příjem ve výši Amount. Varianty jsou dvě občan je buď identifikován svým jménem a adresou, nebo identifikátorem jeho účtu. Pokud metoda uspěje, vrací true, pro neúspěch vrací false (neexistující občan/účet).
- Metody Expense zaznamenají na účet daného občana výdaj ve výši Amount. Varianty jsou dvě občan je buď identifikován svým jménem a adresou, nebo identifikátorem jeho účtu. Pokud metoda uspěje, vrací true, pro neúspěch vrací false (neexistující občan/účet).
- Metoda Audit (Name, Addr, Acct, SumIncome, SumExpense) vyhledá pro občana se zadaným jménem a adresou a všechny dostupné informace (účet, součet příjmů a součet výdajů). Nalezené informace uloží do zadaných výstupních parametrů. Metoda vrátí true pro úspěch, false pro selhání (neexistující dvojice jméno + adresa).
- Metoda ListByName vrátí instanci třídy CIterator. Vrácený objekt má rozhraní podle seznamu níže. Jeho pomocí půjde procházet záznamy v registru od počátku do konce. Při načítání budou záznamy procházené v pořadí zvětšujícího se jména (seřazená podle jména), pro stejná jména budou nejdříve vracené záznamy seřazené podle adresy.

Veřejné rozhraní třídy CIterator obsahuje následující:

- Metoda Atend vrací hodnotu true pokud bylo dosaženo konce registru (tedy iterátor již neodkazuje na platný záznam) nebo false pokud iterátor ještě nedosáhl konce registru (odkazuje na platný záznam).
- Metoda Next posune iterátor na další záznam.
- Metoda Name zjistí jméno osoby na aktuální pozici iterátoru.
- Metoda Addr zjistí adresu osoby na aktuální pozici iterátoru.
- Metoda Account zjistí identifikátor účtu osoby na aktuální pozici iterátoru.
- Můžete se spolehnout, že během používání instance iterátoru nejsou prováděné žádné změny v registru, ze kterého iterátor vznikl (zejména není volaná metoda CTaxRegister::Birth ani CTaxRegister::Death).

Odevzdávejte soubor, který obsahuje implementované třídy CTaxRegister a CIterator. Třídy musí splňovat veřejné rozhraní podle ukázky - pokud Vámi odevzdané řešení nebude obsahovat popsané rozhraní, dojde k chybě při kompilaci. Do tříd 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 destruktoru. Je jedno, zda jsou metody implementované inline nebo odděleně. Odevzdávaný soubor nesmí obsahovat vkládání hlavičkových souborů a funkci main (funkce main a vkládání hlavičkových souborů může zůstat, ale pouze obalené direktivami podmíněného překladu jako v přiloženém archivu).

Třída je testovaná v omezeném prostředí, kde je limitovaná dostupná paměť (dostačuje k uložení seznamu) a je omezena dobou běhu. Implementovaná třída CTaxRegister 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 čas i nároků na paměť. Jednoduché lineární řešení nestačí (pro testovací data vyžaduje čas přes 5 minut). Předpokládejte, že narození a úmrtí jsou řádově méně časté než ostatní operace. 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 případně použijte STL. Pozor, pokud budete pole alokovat ve vlastní režii, zvolte počáteční velikost malou (např. tisíc prvků) a velikost zvětšujte/zmenšujte podle potřeby. Při zaplnění pole není vhodné alokovat ve 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 kvocientem ~1.5 až 2.

Pokud budete používat STL, nemusíte se starat o problémy s alokací. Pozor - v STL máte úmyslně zakázané kontejnery map.

Poznámky:

- V přiloženém archivu naleznete kostru implementace se základní sadou testů.
- Neodevzdávejte řešení obsahující cizí testy. Zkopírované cizí testy budou považované za plagiátorství (testy z přiloženého archivu budeme tolerovat).
- Řešení této úlohy lze použít pro code review.

Vzorová data: Download

Referenční řešení

Hodnotitel: automat

- Program zkompilován
 - Test 'Zakladni test s parametry podle ukazky': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 0.000 s (limit: 8.000 s) Úspěch v závazném testu, hodnocení: 100.00 %
 - Test 'Test meznich hodnot': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 % Celková doba běhu: 0.072 s (limit: 8.000 s)
 - Úspěch v závazném testu, hodnocení: 100.00 % Test 'Test nahodnymi vstupy (Add, Search)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 % Celková doba běhu: 0.004 s (limit: 7.928 s)
 - Úspěch v závazném testu, hodnocení: 100.00 % Test 'Test nahodnymi vstupy (Add, Del, Search)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.004 s (limit: 7.924 s) Úspěch v závazném testu, hodnocení: 100.00 %
 - Test 'Test rychlosti (nahodne hodnoty)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 % Celková doba běhu: 0.378 s (limit: 7.920 s)
 - Úspěch v závazném testu, hodnocení: 100.00 % • Test 'Test nahodnymi hodnotami + test prace s pameti': Úspěch
 - Dosaženo: 100.00 %, požadováno: 25.00 % Celková doba běhu: 0.017 s (limit: 8.000 s) Úspěch v nepovinném testu, hodnocení: 100.00 %
 - Test 'Test rychlosti (nahodna jmena, fixni adresa)': Úspěch Dosaženo: 100.00 %, požadováno: 75.00 %
 - Celková doba běhu: 0.529 s (limit: 10.000 s)
 - Úspěch v nepovinném testu, hodnocení: 100.00 % Test 'Test rychlosti (fixni jmena, nahodna adresa)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 75.00 % Celková doba běhu: 0.440 s (limit: 9.471 s)
 - Úspěch v nepovinném testu, hodnocení: 100.00 % Celkové hodnocení: 100.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00)
- Celkové procentní hodnocení: 100.00 %
- Bonus za včasné odevzdání: 0.72
- Celkem bodů: 1.00 * (7.15 + 0.72) = 7.87

Průměr Maximum Jméno funkce Celkem

18

SW metriky: Řádek kódu: $179 \ 9.94 \pm 7.21$ 29 CTaxRegister::Birth 8 CTaxRegister::Birth Cyklomatická složitost: 49 2.72 ± 2.26

Funkce:

Stav odevzdání: Ohodnoceno

Download

7.8650 Hodnocení:

> Hodnotitel: automat Program zkompilován

- Test 'Zakladni test s parametry podle ukazky': Úspěch Dosaženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 0.000 s (limit: 8.000 s)

14.03.2020 01:32:05

- Úspěch v závazném testu, hodnocení: 100.00 % Test 'Test meznich hodnot': Úspěch Dosaženo: 100.00 %, požadováno: 100.00 %
- Celková doba běhu: 0.259 s (limit: 8.000 s) Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test nahodnymi vstupy (Add, Search)': Úspěch Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.015 s (limit: 7.741 s)
- Úspěch v závazném testu, hodnocení: 100.00 % Test 'Test nahodnymi vstupy (Add, Del, Search)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.013 s (limit: 7.726 s) Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test rychlosti (nahodne hodnoty)': Úspěch Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 1.206 s (limit: 7.713 s) Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test nahodnymi hodnotami + test prace s pameti': Úspěch
 - Dosaženo: 100.00 %, požadováno: 25.00 % Celková doba běhu: 0.073 s (limit: 8.000 s) Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Test rychlosti (nahodna jmena, fixni adresa)': Úspěch Dosaženo: 100.00 %, požadováno: 75.00 %
 - Celková doba běhu: 1.311 s (limit: 10.000 s) Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Test rychlosti (fixni jmena, nahodna adresa)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 75.00 % Celková doba běhu: 1.107 s (limit: 8.689 s)
- Úspěch v nepovinném testu, hodnocení: 100.00 % Celkové hodnocení: 100.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00)
- Celkové procentní hodnocení: 100.00 % Bonus za včasné odevzdání: 0.72
- Celkem bodů: 1.00 * (7.15 + 0.72) = 7.87

Funkce: 20 Řádek kódu: $99 + 4.95 \pm 2.67$ 11 CTaxRegister::Birth Cyklomatická složitost: 3 CTaxRegister::Contains $32\ 1.60\pm0.66$

Celkem Průměr Maximum Jméno funkce

SW metriky: