

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:	7.1500
Max. hodnocení:	7.1500 (bez bonusů)
Odevzdaná řešení:	4 / 20 Volné pokusy + 20 Penalizované pokusy (-2 % penalizace za každé odevzdání)
Nápovědy:	0 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu)

Úkolem je realizovat třídu `CTaxRegister`, která bude implementovat

Pro plánované důslednější potírání daňových úniků je potřeba vybudovat databázi všech obyvatel, ve které budou archivované jejich 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 Hlavní 60, ale Jan Novak bydlící na adrese Hlavní 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 vrátí 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, vrátí `true`, pro neúspěch vrátí `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, vrátí `true`, pro neúspěch vrátí `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`. Vracený 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` vrátí 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 prováděny žádné změny v registru, ze kterého iterátor vznikl (zejména není volána 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 destrukturu. 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 příloženém 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 kopírováním 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 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 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 příloženém archivu naleznete kostru implementace se základní sadou testů.
- Neodevzdávejte řešení obsahující cizí testy. Zkopírované cizí testy budou považované za plagiátorství (testy z příloženého archivu budeme tolerovat).
- Řešení této úlohy lze použít pro code review.

Vzorová data:

[Download](#)

☐ Referenční řešení

4**03.05.2020 22:39:24**[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: 8.000 s)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test meznich hodnot': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 0.081 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: 0.003 s (limit: 7.919 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: 0.003 s (limit: 7.916 s)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test rychlosti (nahodne hodnoty)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.407 s (limit: 7.913 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: 0.020 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: 0.410 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: 0.462 s (limit: 9.590 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 %
- Penalizace za odevzdání po termínu: 7.15
- Celkem bodů: 1.00 * (7.15 - 7.15) = 0.00

		Celkem	Průměr	Maximum	Jméno funkce
SW metriky:	Funkce:	23	--	--	--
	Řádek kódu:	192	8.35 ± 14.38	75	main
	Cyklomatická složitost:	49	2.13 ± 2.47	13	main

3	05.04.2020 20:17:31	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: 8.000 s)
 - Úspěch v závazném testu, hodnocení: 100.00 %
 - Test 'Test mezních hodnot': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 0.124 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: 0.005 s (limit: 7.876 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: 7.871 s)
 - Úspěch v závazném testu, hodnocení: 100.00 %
 - Test 'Test rychlosti (nahodne hodnoty)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.549 s (limit: 7.867 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: 0.020 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: 0.586 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: 0.464 s (limit: 9.414 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 %
- Penalizace za odevzdání po termínu: 7.15
- Celkem bodů: 1.00 * (7.15 - 7.15) = 0.00

		Celkem	Průměr	Maximum	Jméno funkce
SW metriky:	Funkce:	23	--	--	--
	Řádek kódu:	200	8.70 ± 14.33	75	main
	Cyklomatická složitost:	49	2.13 ± 2.47	13	main

2	24.03.2020 10:14:02	Download
Stav odevzdání:	Ohodnoceno	
Hodnocení:	7.1500	

- **Hodnotitel: automat**

- Program zkompileován
- Test 'Základní test s parametry podle ukázky': 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 mezních hodnot': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 0.115 s (limit: 8.000 s)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test náhodnými vstupy (Add, Search)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.005 s (limit: 7.885 s)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test náhodnými vstupy (Add, Del, Search)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.004 s (limit: 7.880 s)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test rychlosti (náhodné hodnoty)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.560 s (limit: 7.876 s)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test náhodnými hodnotami + test práce s pamětí': Úspěch
 - Dosaženo: 100.00 %, požadováno: 25.00 %
 - Celková doba běhu: 0.020 s (limit: 8.000 s)
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Test rychlosti (náhodná jména, fixní adresa)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 75.00 %
 - Celková doba běhu: 0.576 s (limit: 10.000 s)
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Test rychlosti (fixní jména, náhodná adresa)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 75.00 %
 - Celková doba běhu: 0.449 s (limit: 9.424 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 %
- Celkem bodů: 1.00 * 7.15 = 7.15

		Celkem	Průměr	Maximum	Jméno funkce
SW metriky:	Funkce:	23	--	--	--
	Řádek kódu:	200	8.70 ± 14.33	75	main
	Cykloomatická složitost:	49	2.13 ± 2.47	13	main

1	21.03.2020 01:56:42	Download
---	---------------------	----------

Stav odevzdání: Ohodnoceno

Hodnocení: 5.5055

- **Hodnotitel: automat**

- Program zkompileován
- Test 'Základní test s parametry podle ukázky': 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 mezních hodnot': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 0.162 s (limit: 8.000 s)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test náhodnými vstupy (Add, Search)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.005 s (limit: 7.838 s)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test náhodnými vstupy (Add, Del, Search)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.005 s (limit: 7.833 s)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test rychlosti (náhodné hodnoty)': Úspěch

- Dosaženo: 100.00 %, požadováno: 50.00 %
- Celková doba běhu: 0.575 s (limit: 7.828 s)
- Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test nahodnými hodnotami + test práce s pamětí': Úspěch
 - Dosaženo: 100.00 %, požadováno: 25.00 %
 - Celková doba běhu: 0.019 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: 0.580 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: 0.476 s (limit: 9.420 s)
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- Bylo detekováno celkem 417 neuvolněných bloků paměti - penalizace 30%
- Celkové hodnocení: 70.00 % (= (1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00) * 0.7)
- Celkové procentní hodnocení: 70.00 %
- Bonus za včasné odevzdání: 0.72
- Celkem bodů: $0.70 * (7.15 + 0.72) = 5.51$

		Celkem	Průměr	Maximum	Jméno funkce
SW metriky:	Funkce:	22	--	--	--
	Řádek kódu:	227	10.32 ± 14.68	75	main
	Cyklomatická složitost:	47	2.14 ± 2.53	13	main