

Poštovní schránka

Termín odevzdání:	19.04.2020 23:59:59
Pozdní odevzdání s penalizací:	30.06.2020 23:59:59 (Penále za pozdní odevzdání: 100.0000 %)
Hodnocení:	3.5750
Max. hodnocení:	7.1500 (bez bonusů)
Odevzdaná řešení:	2 / 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 vytvořit třídy `CMailBox` a `CMail`, které budou simulovat poštovní schránku na poštovním serveru.

Celý program bude tvořen několika třídami. Některé třídy budete vyvíjet sami, jiné jsou hotové v testovacím prostředí. Pro účely testování si je budete muset v odlehčené verzi také vyvinout. Jedná se o postup nazývaný "mocking" - pro odladění a otestování Vašeho kódu si připravíte implementaci cizích tříd. Takto implementované třídy mají stejné rozhraní jako třídy produkční, ale jejich implementace je nějak zjednodušená nebo upravená pro snazší testování.

`CMailBox`

Tato třída reprezentuje jednu e-mailovou schránku. schránka může obsahovat došlou poštu (instance třídy `CMail`), pošta může být rozdělena do složek. Tuto třídu budete implementovat celou.

`CMail`

Tato třída reprezentuje jeden e-mail. E-mail je popsán datem a časem doručení, odesilatelem, obsahem a přílohami. Pro reprezentaci těchto náležitostí budete používat třídy `CTimeStamp`, `std::string`, `CMailBody` a `CAttach`. Třídu `CMail` budete implementovat celou.

`CTimeStamp`

Je jednoduchá třída reprezentující datum a čas. Tato třída je implementovaná v testovacím prostředí, budete ji implementovat pouze pro účely testování. Vaše implementace této třídy musí zůstat v bloku podmíněného překladu (viz přiložený vzor).

`CMailBody`

Tato třída zabaluje tělo e-mailu. Je implementovaná v testovacím prostředí, Vy budete opět implementovat pouze odlehčenou variantu pro testování. Tato Vaše implementace opět musí zůstat v bloku podmíněného překladu.

`CAttach`

Tato třída reprezentuje přílohy. Je implementovaná v testovacím prostředí a v přiložené ukázce máte její kompletní podobu. Implementaci opět ponechte v bloku podmíněného překladu.

CMailBox

Tuto třídu implementujete celou. Implementace musí být MIMO blok podmíněného překladu. Požadované rozhraní obsahuje:

konstruktor

vytvoří prázdnou schránku. Ve schránce je jediná složka `inbox` pro příchozí poštu.

destruktor, kopírující konstruktor a operátor =

pokud jsou potřeba,

`Delivery (mail)`

Metoda přidá zadaný e-mail do složky `inbox`. Metoda vrací hodnotu `true` pro úspěch, `false` pro neúspěch (ale selhat by neměla).

`NewFolder (name)`

Metoda vytvoří novou složku zadaného jména. Nově vytvořená složka bude prázdná. Metoda vrací `true` pro úspěch, `false` pro neúspěch (složka stejného jména již existuje). Předpokládáme pouze jednoúrovňové složky (nelze vnořovat).

`MoveMail (fromFld, toFld)`

Metoda přesune poštu ze zadané složky `fromFld` do cílové složky `toFld`. Přesunutí znamená, že složka `fromFld` bude po úspěšném dokončení prázdná. Metoda vrací `true` pro úspěch, `false` pro neúspěch (některá ze složek neexistuje).

`ListMail (fld, from, to)`

Metoda vrátí seznam e-mailů ze zadané složky `fld` v zadaném rozmezí času (`from` - `to`, obě meze včetně). Pokud zadaná složka neexistuje, vrátí prázdný seznam.

`ListAddr (from, to)`

Metoda vrátí množinu se adresami odesílatelů e-mailů, které došly v zadaném rozmezí času (`from` - `to`, obě meze včetně). Hledá se ve všech složkách. Pokud v zadaném intervalu nedošel žádný e-mail, bude výsledkem prázdná množina.

CMail

Tuto třídu implementujete celou. Implementace musí být MIMO blok podmíněného překladu. Požadované rozhraní obsahuje:

konstruktor

který vytvoří instanci se složkami nastavenými podle parametrů. Parametr attach může být NULL, pokud příloha neexistuje,

destruktor, kopírující konstruktor, operátor =, ...

pokud jsou potřeba,

přístupové metody (getter)

pro jednotlivé složky (čas, odesílatel, tělo mailu, přílohy),

operátor pro výstup

který zobrazí datum, odesílatele, tělo mailu a přílohu (pokud existuje). Pro všechny složky jsou připravené operátory pro výstup, složky jsou oddělené jednou mezerou (případně znakem + a informací o příloze, viz ukázka). Testovací prostředí výstupy nekontroluje, jsou zde pouze pro ladění a testování.

CTimeStamp

Tuto třídu implementujete pouze pro účely ladění a testování. Implementace MUSÍ být v bloku podmíněného překladu. Produkční rozhraní obsahuje:

konstruktor

který vytvoří instanci ze zadaných složek rok/měsíc/den/hodina/minuta/sekunda,

Compare (x)

Metoda porovná instanci a parametr x. Vráť zápornou hodnotu pokud je instance menší než x, 0 pokud jsou si rovné a kladnou hodnotu pokud je instance větší než x.

operátor pro výstup

zobrazí reprezentovaný časový údaj v podobě YYYY-MM-DD HH24:MI:SS.

další

Pro skutečný běh se bude používat implementace v testovacím prostředí. I když si do své implementace CTimeStamp přidáte nějaké další metody, tyto nebudou v testovacím prostředí dostupné. Proto doporučujeme používat pouze toto rozhraní a další metody nepřidávat.

CMailBody

Tuto třídu implementujete pouze pro účely ladění a testování. Implementace MUSÍ být v bloku podmíněného překladu. Produkční rozhraní obsahuje:

konstruktor

který vytvoří instanci ze zadané informace o délce a obsahu,

destruktor, kopírující konstruktor, operátor =, ...

instance obsahuje dynamicky alokovaná data, v implementaci proto je k dispozici kopírující konstruktor/op=/..., aby šlo objekty snadno přesouvat a kopírovat,

operátor pro výstup

v dodané podobě, opět určený pro testování a ladění.

další

Pro skutečný běh se bude používat implementace v testovacím prostředí. I když si do své implementace CMailBody přidáte nějaké další metody, tyto nebudou v testovacím prostředí dostupné. Proto doporučujeme používat pouze toto rozhraní a další metody nepřidávat. V testovacím prostředí je dále úprava, která zamezuje vytváření instancí CMailBody dynamicky (pomocí new).

CAttach

Tato třída je implementovaná v testovacím prostředí a její zdrojový kód je v příloze (v bloku podmíněného překladu). Třída reprezentuje přílohu e-mailu (velmi zjednodušenou). Protože se přílohy často opakují (ty samé soubory se rozesílají více uživatelům, řetězové maily), jsou přílohy řešené pomocí počítaných referencí. Rozhraní proto je:

konstruktor

který vytvoří instanci ze zadaném obsahu (zde velmi zjednodušenou). Konstruktor inicializuje proměnnou m_RefCnt, tato proměnná drží počet odkazů na tento objekt. Objekt zanikne v okamžiku, kdy na něj nebude nikdo odkazovat (m_RefCnt==0).

AddRef

metoda zvedne počet odkazů. Každý, kdo získá odkaz na objekt, zavolá tuto metodu. Tedy například konstruktor CMail nebo getter Attach budou volat tuto metodu, aby počet odkazů souhlasil.

Release

metoda sníží počet odkazů a pokud dosáhne 0, objekt zruší. Tuto metodu zavolá každý, kdo dříve získal odkaz na objekt a již jej nepotřebuje. Tedy např. destruktor CMail nebo kód, který odkaz na objekt dříve získal při volání gettru Attach.

destruktor

uvolní případné prostředky (zde velmi zjednodušeno). Všimněte si, že destruktory je privátní, efektivně tedy nelze vytvořit staticky alokovanou instancí a vytvořenou instancí nelze smazat pomocí `delete`. To má dobrý význam, k výmazu dojde, až instance nebude odkazovaná z jiného místa, objekt nelze snadno zničit když na něj odkazuje někdo jiný.

kopírující konstruktor, operátor `=`, ...

jsou přesunuté do privátní sekce, tedy pro okolní svět zakázané.

operátor pro výstup

v dodané podobě, opět určený pro testování a ladění.

mutable `m_RefCnt`

tuto složku lze modifikovat i z `const` metod.

další

Pro skutečný běh se bude používat implementace v testovacím prostředí. I když si do své implementace `CAttach` přidáte nějaké další metody, tyto nebudou v testovacím prostředí dostupné. Proto doporučujeme používat pouze toto rozhraní a další metody nepřidávat.

Odevzdávejte soubor, který obsahuje implementaci požadovaných tříd. Třídy musí splňovat veřejné rozhraní podle ukázky - pokud Vámi odevzdané řešení nebude obsahovat popsané 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 destruktory. Je jedno, zda jsou metody implementované inline nebo odděleně. Odevzdávaný soubor nesmí obsahovat vkládání hlavičkových souborů a funkci `main`. Funkce `main`, vkládání hlavičkových souborů a třídy `CTimeStamp`, `CMailBody` a `CAttach` mohou zůstat, ale pouze obalené direktivami podmíněného překladu jako v příložené ukázce.

Při řešení úlohy využijte STL. Využijte STL tak, abyste byli schopni rychle zpřístupňovat složky dle jmen, dokázali rychle nalézt e-mail podle data/času a co nejvíce omezte zbytečné kopírování e-mailů. Úloha má povinnou, nepovinnou a bonusovou část. Povinnými testy projde i naivní řešení. Nepovinným testem projde vylepšená verze, která dokáže pracovat s velkým množstvím e-mailů (omezte počet potřebných porovnávání při hledání e-mailů). Bonusovým testem projde řešení, které dokáže rychle kopírovat/přesouvat e-maily (provádí se mnoho přesunů mezi složkami, budete muset použít přesouvací konstruktor/operátor `=` z C++11 a deklaraci `noexcept`).

Řešení této úlohy nemůže být použito pro code review, protože by nešlo rozumně hodnotit návrh řešení (rozhraní je do značné míry dáno zadáním).

Vzorová data:

[Download](#)

☐ Referenční řešení

2

15.04.2020 21:56:14

[Download](#)

Stav odevzdání:

Ohodnoceno

Hodnocení:

3.5750

• **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: 4.000 s)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test nahodnymi daty': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.124 s (limit: 4.000 s)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test rychlosti (mnoho e-mailu)': Program překročil přidělenou maximální dobu běhu
 - Vyčerpání limitu na celý test, program násilně ukončen po: 6.006 s (limit: 6.000 s)
 - Neúspěch v nepovinném testu, hodnocení: 50.00 %
- Test 'Test rychlosti (mnoho presunu)': Nebylo testováno
 - Neúspěch v bonusovém testu, hodnocení: Bonus nebude udělen
- Celkové hodnocení: 50.00 % (= 1.00 * 1.00 * 0.50)
- Celkové procentní hodnocení: 50.00 %
- Celkem bodů: 0.50 * 7.15 = 3.58

SW metriky:

	Celkem	Průměr	Maximum	Jméno funkce
Funkce:	28	--	--	--
Řádek kódu:	309	11.04 ± 17.68	100	main
Cyklomatická složitost:	54	1.93 ± 1.51	7	CTimeStamp::Compare

Stav odevzdání: Ohodnoceno

Hodnocení: 0.0000

• **Hodnotitel: automat**

- Program zkompileován
- Test 'Zakladni test s parametry podle ukazky': Neúspěch
 - Dosaženo: 72.41 %, požadováno: 100.00 %
 - Celková doba běhu: 0.000 s (limit: 4.000 s)
 - Neúspěch v závazném testu, hodnocení: 0.00 %
 - Nesprávný výstup **[Zpřístupnit nápovědu (32 B)]**
 - Nesprávný výstup **[Zpřístupnit nápovědu (39 B)]**
 - Nesprávný výstup **[Zpřístupnit nápovědu (39 B)]**
 - Nesprávný výstup **[Zpřístupnit nápovědu (32 B)]**
 - Nesprávný výstup **[Zpřístupnit nápovědu (32 B)]**
 - Nesprávný výstup **[Zpřístupnit nápovědu (32 B)]**
 - Nesprávný výstup **[Zpřístupnit nápovědu (32 B)]**
 - Nesprávný výstup **[Zpřístupnit nápovědu (32 B)]**
- Celkové hodnocení: 0.00 %
- Celkové procentní hodnocení: 0.00 %
- Celkem bodů: $0.00 \cdot 7.15 = 0.00$

		Celkem	Průměr	Maximum	Jméno funkce
SW metriky:	Funkce:	27	--	-- --	
	Řádek kódu:	307	11.37 ± 18.71	104	main
	Cyklomatická složitost:	51	1.89 ± 1.52	7	CTimeStamp::Compare