

Odpočet

Termín odevzdání: 21.03.2021 23:59:59

Zbývá 6 dní, 22h, 43m a 44s

Pozdní odevzdání s penalizací: 16.05.2021 23:59:59 (Penále za pozdní odevzdání: 100.0000 %)**Hodnocení:** 0.0000**Max. hodnocení:** 1.0000 (bez bonusů)**Odevzdaná řešení:** 0 / 25**Nápovědy:** 0 / 25

V poslední době se zábavou plebsu stalo především odpočítávání času, ať už do konce izolace, karantény, nouzového stavu nebo funkčního období. Pro tyto úkoly se ukázalo, že ve standardní knihovně C++ chybí třída pro práci s časem v uživatelsky přívětivé podobě. Už žádné vteřiny do konce!

Vaším úkolem je realizovat třídu `CTime`, která bude reprezentovat čas. Realizovaná třída musí splňovat následující rozhraní:

konstruktor implicitní

Inicializuje prázdný objekt.

konstruktor s 2 parametry typu `int` reprezentující hodiny a minuty

Inicializuje objekt s předanými parametry a sekundy nastaví na 0.

V případě nevalidních parametrů vyhodí výjimku `std::invalid_argument`, výjimka je součástí standardní knihovny, její deklarace je v hlavičkovém souboru `stdexcept`.

konstruktor s 3 parametry typu `int` reprezentující hodiny, minuty a sekundy

Inicializuje objekt s předanými parametry.

V případě nevalidních parametrů vyhodí výjimku `std::invalid_argument`, výjimka je součástí standardní knihovny, její deklarace je v hlavičkovém souboru `stdexcept`.

kopírující konstruktor, operátor přiřazení, destruktor

Budou implementovány pokud to vnitřní struktury Vaší třídy vyžadují.

operator+, operator+=

Umožní přičíst k předanému času specifikovaný počet sekund (`int`).

operator-, operator-=

Umožní odečíst od předaného času specifikovaný počet sekund (`int`).

operator-

Umožní odečíst od sebe dva objekty typu `CTime`. Výsledek je minimální počet sekund, který potřebujeme přičíst k jednomu z předaných časů, abychom získali čas druhý.

operator++, operator--

Umožní přičítat (resp. odečítat) sekundu k danému času.

relační operátory (<, <=, >, >=, ==, !=)

Umožní porovnávat dva časy mezi sebou.

operator<<

Bude umožňovat výstup objektu do C++ streamu (formát výstupu HH:MM:SS, jednociferné hodiny jsou doplněny mezerou).

operator>>

Bude umožňovat načtení ze vstupního C++ streamu, ignorujte počáteční bílé znaky.

V případě nevalidního vstupu, nastaví `std::ios::failbit` a původní objekt nemění.

Odevzdávejte zdrojový soubor, který obsahuje Vaši implementaci třídy `CTime`. V odevzdávaném souboru ponechte hlavičkové soubory, třídu `CTimeTester` a funkci `main` v bloku podmíněného překladu.

V tomto příkladu není poskytnutý předpis pro požadované rozhraní třídy. Z textového popisu, ukázky použití níže a znalostí přetěžování operátorů byste měli být schopni toto rozhraní vymyslet.

Testovací skript očekává, že třída bude používat členské proměnné `m_Hour`, `m_Minute` a `m_Second`. Stejně tak očekává, že třída `CTimeTester` bude mít `friend` přístup. Pokud tuto část změníte, váš kód nepůjde zkompileovat.

Nápověda

- Jako validní akceptujte hodiny od 0 do 23, minuty a sekundy od 0 do 59.
- Předchozí bod aplikujte i při výpočtech. Pokud by v rámci výpočtu došlo k přetečení / podtečení času do dalšího / předchozího dne, uvažujte výsledek modulo 24 hodin.
- Pokud Vám program nejde zkompileovat, ujistěte se, že máte správně přetížené operátory. Zejména si kontrolujte kvalifikátory `const`.
- Operátor pro výstup implementujte správně -- neposílejte data na `cout`, posílejte je do předaného výstupního proudu. Za výstupem čísla do proudu nepřidávejte odřádkování ani jiné bílé znaky.
- Načítání ze streamu musí správně manipulovat s příznakem chyby (`failbit`). Chování `CTime` má podobné chování jako pro desetinná čísla, úvodní bílé znaky se přeskakují. Pro nastavení příznaku chyby se hodí `is . setstate (ios::failbit)`.
- Použití třídy předpokládá mnoho přetížených operátorů. Navrhněte rozhraní tak, abyste neměli zbytečně dlouhou implementaci. Využijte vlastností C++ (implicitní parametry), rozumný návrh dokáže ušetřit mnoho práce.

Vzorová data:

Download

Odevzdat:

Choose File

No file chosen

Odevzdat

Referenční řešení