

## Datové typy II

Termín odevzdání:	01.05.2022 23:59:59	292141.510 sec
Pozdní odevzdání s penalizací:	15.05.2022 23:59:59 (Penále za pozdní odevzdání: 100.0000 %)	
Hodnocení:	0.0000	
Max. hodnocení:	2.0000 (bez bonusů)	
Odevzdaná řešení:	0 / 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 sadu tříd, které budou implementovat popis některých datových typů v C/C++. Jedná se o rozšíření jednodušší varianty úlohy, proto doporučujeme začít s řešením jednodušší varianty a řešení následně rozšířit.

Rozšíření spočívá v přidání dalších datových typů:

- ukazatel,
- pole,
- možnost neomezeně skládat typy podle platných C/C++ pravidel. Tedy např. půjde vytvořit ukazatel na strukturu a přidat jej jako další složku do struktury.

Očekává se, že implementujete následující třídy s následujícím rozhraním:

### CDataTypeInt

Třída reprezentuje primitivní typ `int`. Má následující rozhraní:

implicitní konstruktor

inicializuje instanci třídy,

`getSize()`

metoda vrátí velikost typu, zde vždy 4,

operátor `==`

porovná tento typ s jiným typem, vrátí `true`, pokud jsou oba typy shodné (oba jsou typy `int`),

operátor `!=`

porovná tento typ s jiným typem, vrátí `true`, pokud jsou typy rozdílné,

operátor `<<`

zobrazí název typu do zadaného proudu.

### CDataTypeDouble

Třída reprezentuje primitivní typ `double`. Má následující rozhraní:

implicitní konstruktor

inicializuje instanci třídy,

`getSize()`

metoda vrátí velikost typu, zde vždy 8,

operátor `==`

porovná tento typ s jiným typem, vrátí `true`, pokud jsou oba typy shodné (oba jsou typy `double`),

operátor `!=`

porovná tento typ s jiným typem, vrátí `true`, pokud jsou typy rozdílné,

operátor `<<`

zobrazí název typu do zadaného proudu.

### CDataTypeEnum

Třída reprezentuje výčtový typ. Má následující rozhraní:

implicitní konstruktor

inicializuje instanci třídy,

`getSize()`

metoda vrátí velikost typu, zde vždy 4,

`add()`  
metoda přidá další hodnotu do výčtu. Pokud stejná hodnota ve výčtu již existuje, je vyvolaná výjimka (viz ukázkový běh),  
operátor `==`  
porovná tento typ s jiným typem, vrátí `true`, pokud jsou oba typy shodné (oba jsou výčtové typy a mají stejný výčet hodnot ve stejném pořadí),  
operátor `!=`  
porovná tento typ s jiným typem, vrátí `true`, pokud jsou typy rozdílné,  
operátor `<<`  
zobrazí název typu do zadaného proudu. Pozor, hodnoty výčtu musí být zobrazené v pořadí zadávání.

## CDataTypeStruct

Třída reprezentuje typ struktura. Má následující rozhraní:

implicitní konstruktor  
inicializuje instanci třídy,  
`getSize()`  
metoda vrátí velikost typu (dle obsažených složek),  
`addField(name, type)`  
metoda přidá další složku zadaného jména a typu (`int/double/enum`) na konec seznamu složek. Pokud je jméno složky duplicitní, vyhlásí výjimku (viz ukázkový běh),  
`field(name)`  
zpřístupní složku zadaného jména, případně vyhodí výjimku pokud složka takového jména neexistuje (viz ukázkový běh). Složka zpřístupněná touto metodou bude pouze čtena,  
operátor `==`  
porovná tento typ s jiným typem, vrátí `true`, pokud jsou oba typy shodné (oba typy jsou struktury, mají stejný počet složek, složky na každé pozici mají stejný typ, jména složek se ale mohou lišit),  
operátor `!=`  
porovná tento typ s jiným typem, vrátí `true`, pokud jsou typy rozdílné,  
operátor `<<`  
zobrazí název typu do zadaného proudu. Pořadí složek odpovídá pořadí jejich přidání metodou `addField`.

## CDataTypeArray

Třída reprezentuje typ pole. Má následující rozhraní:

konstruktor (`size,type`)  
inicializuje instanci třídy, pole má velikost `size`, prvkem pole mají typ `type`,  
`getSize()`  
metoda vrátí velikost typu (podle typu a velikosti),  
`element()`  
zpřístupní typ prvku pole (viz ukázkový běh),  
operátor `==`  
porovná tento typ s jiným typem, vrátí `true`, pokud jsou oba typy shodné (oba typu pole, stejná velikost, stejný typ prvku),  
operátor `!=`  
porovná tento typ s jiným typem, vrátí `true`, pokud jsou typy rozdílné,  
operátor `<<`  
zobrazí název typu do zadaného proudu.

## CDataTypePtr

Třída reprezentuje typ ukazatel. Má následující rozhraní:

konstruktor  
inicializuje instanci třídy, ukazatel odkazuje na datový typ předaný v parametru,  
`getSize()`  
metoda vrátí velikost typu (zde vždy 8),  
`element()`  
zpřístupní typ odkazovaný ukazatelem (viz ukázkový běh),  
operátor `==`  
porovná tento typ s jiným typem, vrátí `true`, pokud jsou oba typy shodné (oba typy ukazatele a odkazují na stejný datový typ),  
operátor `!=`  
porovná tento typ s jiným typem, vrátí `true`, pokud jsou typy rozdílné,  
operátor `<<`

zobrazí název typu do zadaného proudu.

Odevzdávejte zdrojový kód s implementací tříd `CDataTypeInt`, `CDataTypeDouble`, `CDataTypeEnum`, `CDataTypeStruct`, `CDataTypeArray` a `CDataTypePtr`. Třídy musejí implementovat rozhraní popsané výše. Do tříd si ale můžete doplnit další metody, které Vám implementaci zjednoduší.

#### Poznámky:

- Při implementaci můžete využít STL a typ `c++ string`. Základem implementace je ale dobrý návrh tříd a využití polymorfismu. Vaše řešení nebude uznáno, pokud nebude využívat polymorfismus. Zvláštní pozornost si zaslouží typy ukazatel a pole, zejména pak umístění závorek v deklaracích.
- Při porovnání zápisu typů (výstupy operátoru `<<`) se nekontrolují bílé znaky (viz příložená funkce `whitespaceMatch`).
- Ukázka použití tříd je v příloženém archivu.
- Pro návratovou hodnotu metody `getSize` použijte typ `size_t`.
- Řešení tohoto problému, které projde všemi testy na 100%, lze použít pro code review.
- Bodová hodnocení úloh Datové typy I a Datové typy II se sčítají. Nezapomeňte proto odevzdat své řešení do obou zadání (řešení úlohy Datové typy II je rozšířením řešení problému Datové typy I, tedy zdrojový kód řešení rozšířené úlohy bude fungovat i jako řešení problému zjednodušeného).
- **Dodatek 9.4.2022:** pokud je vstupním parametrem metody řetězec, pak umožněte použití jak C-řetězce (`const char *`), tak i C++ řetězce (`std::string`).

Vzorová data:

Download

Odevzdat:

Choose File

No file chosen

Odevzdat



Referenční řešení