

Pojem třídy a struktury v C++ (o – Objektová vlastnost)

- struct a class jsou složené datové typy – vychází ze struct jazyka C
 - pomocí struct a class se definuje nový datový typ (má „stejné“ vlastnosti a možnosti co do použití jako standardní typy – int, double ...)
 - struct v C++ je rozšířena o (objektové vlastnosti) možnost přidat metody a přístupová práva
 - struct a class v C++ se liší pouze minimálně (leč v důležité věci)
 - nebude-li dále řečeno jinak, platí uvedené pro struct i class
 - třída tvoří jmenný prostor
-
- v jazyce C struct obsahuje pouze data
 - v jazyce C++ obsahuje data, metody (funkce pracující s třídou)
 - v jazyce C++ lze nastavit přístupová práva pro data a metody (lze je používat pouze „uvnitř“ třídy, nebo je může používat i „uživatel“ třídy)
 - v C++ platí, že „data jsou to nejcennější, co máme“ a proto jsou v class data implicitně „schována“ – private
 - struktura kvůli zpětné kompatibilitě s jazykem C musí implicitně umožňovat uživatelský přístup k datům - public

Pojem třídy a struktury v C++

- pro deklaraci/definici slouží klíčové slovo class/struct
- deklarace třídy - pouze popis třídy – nevyhrazuje paměť – uvádíme v hlavičkovém souboru
- deklarací struktury/třídy vzniká nový datový typ, který se chová (je rovnocenný) jako základní datové typy (pokud autor třídy odpovídající činnosti popíše)
- objektové vlastnosti rozšířeny i pro union

Na základě znalostí o struktuře jazyka C napište pro jazyk C++:

- oznámení jména struktury, třídy
- definice struktury, třídy s vyznačením, kde budou parametry (data a metody)
- nadefinujte dva objekty (proměnné) dané třídy a ukazatel na objekt dané třídy, který inicializujete tak aby ukazoval na první z objektů

Pozn.: postupujte na základě znalostí o strukturách

Pro nový datový typ platí stejná pravidla jako pro základní typy

- deklarace - oznámení názvu/nového typu. Vlastní definice dat a metod je později.
- následně lze použít pouze ukazatel (ne instanci), jelikož není známa velikost typu
- netvoří kód -> je v hlavičkovém souboru

```
// deklarace typu - netvoří kód  
class jméno_třída;  
struct jméno_struktury;
```

- členská data a metody jsou uvedeny za názvem třídy v {} závorkách, **musí** být zakončeno středníkem
- nepoužívat typedef – většinou není nutný
- popis dat a hlaviček metod netvoří kód -> píšeme do hlavičkového souboru

```
// definice typu - netvoří kód - předpis  
class jméno_třída { parametry, tělo třídy };  
struct jméno_struktury {parametry, tělo      };
```

- definice proměnné, vyhradí paměť -> v cpp zdrojovém souboru
- klíčové slovo class, struct není nutné uvádět, stačí název typu
- zápis totožný jako pro int, nejste-li si zápisem jistí, napište pro typ int a následně typ změňte

```
// definice proměnných daného typu a práce s nimi  
jméno_třída a, b, *pc; // zabere paměť pro proměnné  
//obdoba int a,b,*pc  
// 2x objekt, 1x ukazatel  
pc = &a; // inicializace ukazatele
```

Pojem třídy a struktury v C++ - postup při psaní programu

- vytvoříme dva soubory pro třídu – hlavičkový a zdrojový
- součástí vývoje třídy by měl být i testovací kód (testovací funkce) – může být součástí zdrojového souboru („odstranitelnou“ například pomocí #if - #endif), lépe vytvořit soubor zvláštní (například s funkcí main)
- hlavičkový soubor ošetřit proti vícenásobnému načtení
- ve zdrojovém souboru třídy (a souborech kde budeme třídu používat) naincludovat hlavičkový soubor
- v hlavičkovém souboru napsat deklaraci třídy včetně tělíčka (nezapomenout ukončit středníkem)

hlavičkový soubor třídy

```
#ifndef CCOMPLEX_H123  
#define CCOMPLEX_H123  
class CComplex {
```

```
};
```

```
#endif
```

zdrojový soubor třídy

```
#include "ccomplex.h"
```