

NPRG041: Programování v C++

<http://www.ksi.mff.cuni.cz/~svoboda/courses/211-NPRG041/>

Cvičení 9

Chytré ukazatele, algoritmy, funktory

Martin Svoboda

martin.svoboda@matfyz.cuni.cz

8. 12. 2021

Univerzita Karlova, Matematicko-fyzikální fakulta

P1: Databáze filmů: Data

Vytvořte jednoduchou aplikaci pro databázi filmů

- Rozlišovat budeme dva druhy titulů: film a seriál
 - Každý titul má název, rok uvedení a číselné hodnocení
 - Film má navíc jméno režiséra
 - Seriál naopak počet sérií a celkový počet epizod
- Vstupní data najdeme v CSV souboru s následující strukturou
 - Na každém řádku je záznam pro jeden titul
 - Typ titulu je určen klíčovým písmenem na začátku
 - Jednotlivé údaje oddělujeme pomocí středníku
 - Pro ilustraci můžeme předpokládat například tato data
 - F;Vratné lahve;2007;76;Jan Svěrák
 - S;Game of Thrones;2011;91;8;73
 - S;Squid Game;2021;82;1;9
 - F;Medvídek;2007;53;Jan Hřebejk

P1: Databáze filmů: Data

Pokračování...

- Jednotlivé tituly reprezentujte vhodnými třídami
 - Např. `Title`, `Movie` a `Series`
 - Aneb využijte dědičnost
 - Konstruktor předka se volá následujícím způsobem
 - `Movie(...)` : `Title(...)` { ... }
- Vstupní data načtěte a zpracujte takto
 - Nejprve si připravte kontejner typu vektor
 - Zpracování vstupního souboru umístěte do samostatné funkce
 - Objekty titulů vytvářejte pomocí dynamické alokace
 - Zatím opět používejte jen obyčejné ukazatele
- Nakonec zařídte vypsání načtených titulů na standardní výstup
 - `Movie {title: "Vratné lahve", year: 2007, rating: 76, director: "Jan Svěrák"}`

P2: Databáze filmů: Ukazatele

Upravte předchozí kód následujícím způsobem

- Namísto obyčejných začněte používat chytré ukazatele
 - Konkrétně budeme potřebovat ty sdílené
 - `#include <memory>`
 - `std::shared_ptr<Item>`
 - `std::make_shared<Item>(...)`
 - Předejte argumenty odpovídající zamýšlenému konstrukturu

P3: Databáze filmů: Dotaz

Rozšiřte náš kód o vyhodnocení jednoduchého dotazu

- Najděte seriály s hodnocením alespoň 80 a nejméně 10 díly
 - `#include <algorithm>`
 - `std::copy_if(itBegin, itEnd, itTarget, pred)`
 - `bool operator()(const T& item) { ... }`
 - `#include <typeinfo>`
 - `typeid(*item) == typeid(T)`
 - `dynamic_cast<Item*>(&*item)`
 - `std::vector::erase(itBegin, itEnd)`
- Nalezené tituly vypište vzestupně podle roku uvedení
 - `std::sort(itBegin, itEnd)`
 - `bool operator<(const T& i1, const T& i2) { ... }`
 - `std::sort(itBegin, itEnd, comp)`
 - `bool operator()(const T& i1, const T& i2) { ... }`