NPRG041: **Programování v C++**

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

Cvičení 11

Kontejnery, iterátory

Martin Svoboda
martin.svoboda@matfyz.cuni.cz

22. 12. 2021

Univerzita Karlova, Matematicko-fyzikální fakulta

P1: Gumové pole

Naprogramujte vlastní kontejner gumového pole

- Pole bude šablonované
  - Jediný parametr bude určovat typ prvku
- Vnitřně data organizujte následujícím způsobem
  - Použijte vektor chytrých unikátních ukazatelů na bloky prvků
    - `std::unique_ptr<T[]>`
    - `std::make_unique<T[]>(block_size)`
  - Každý blok reprezentujte jako tradiční pole fixní velikosti
    - Tuto velikost specifikujte pomocí parametrického konstruktoru
  - Pamatujte si také počet aktuálně vložených prvků

P1: Gumové pole

Pokračování…

- Implementujte následující funkce
  - `T& operator[](size_t index) const;`
    - `…[index / block_size][index % block_size]`
  - `T& at(size_t index);`
  - `void push_back(const T& item);`
    - `(*this)[…]`
  - `std::ostream& operator<<(std::ostream& os, const Array<T>& array);`
- Všechny funkce experimentálně vyzkoušejte

P2: Gumové pole: Iterátory

Přidejte do našeho kontejneru implementaci dopředného iterátoru

- Realizujte jej pomocí vnitřní třídy
  - `#include <iterator>`
  - `class iterator;`
  - `template<typename T>`
    `class Array<T>::iterator { ... }`
  - Vnitřně uchovejte ukazatel na gumové pole a číslo pozice
- Přidejte následující tagy
  - `using iterator_category =`
    `std::forward_iterator_tag;`
  - `using difference_type = std::ptrdiff_t;`
  - `using value_type = T;`
  - `using pointer = T*;`
  - `using reference = T&;`

P2: Gumové pole: Iterátory

Pokračování…

- Definujte následující konstruktor
  - `iterator(Array<T>* array, size_t position);`
- Implementujte následující základní funkce
  - `T& operator*() const;`
  - `bool operator!=(const iterator& other) const;`
  - `iterator& operator++();`
    - Za posledním prvkem už hodnotu vnitřní pozice nezvyšujte
- Do třídy pole přidejte následující funkce
  - `iterator begin();`
  - `iterator end();`

P2: Gumové pole: Iterátory

Pokračování…

- Nové funkce opět experimentálně vyzkoušejte
  - `for (auto&& item : array) { ... }`
  - `for (auto it = array.begin(); it != array.end(); ++it) { ... }`
- Můžete přidat také následující funkce iterátoru
  - `T* operator->() const;`
  - `iterator operator++(int);`
  - `iterator operator+(const difference_type& movement) const;`