Cvičení C++

5.11.2018

faltin@ksi.mff.cuni.cz

Ukazatele a reference

```
struct coordinate {
  int x, y;
} c;
```

```
coordinate *ptr = &c;                    coordinate &ref = c;
cout << ptr->x << (*ptr).y << endl;      cout << ref.x << ref.y << endl;
coordinate cc = *ptr;                    coordinate  = ref;

coordinate d;                            coordinate d;
ptr = &d;                                coordinate &ref2 = d;
```

Dynamická alokace

1. unique_ptr<T> + make_unique<T>(…);
2. shared_ptr<T> + make_shared<T>(…);
3. new/new[] + delete/delete[]

Unique_ptr + make_unique

- r-value/move semantic
- **Jeden vlastník**

```
struct printer_interface {};
struct pretty_printer : printer_interface {…};
struct ugly_printer : printer_interface {…};

int x; std::cin >> x;

unique_ptr<printer_interface> printer_ptr;
if (x == 0) {
  printer_ptr = make_unique<pretty_printer>();
} else {
  printer_ptr = make_unique<ugly_printer>();
}

do_cool_stuff(*printer_ptr);
```

Shared_ptr + make_shared

- Počítání referencí
- **Více vlastníků**

```
vector<shared_ptr<int>> data1;
my_fn_to_fill_data(data1);

vector<shared_ptr<int>> data2(data1.size());
std::copy(begin(data1), end(data1), begin(data2)); // copy

// Zjednodušený kód!
std::thread other_thread(do_things_in_this_thread, data2); // spawn thread
do_thing_in_this_thread(data2);
other_thread.join();
```

New/new[] + delete/delete[]

- C-style/low level
- **Programátor chce sám řešit alokaci**

```
class my_int_unique_ptr {
public:
  my_int_unique_ptr(int value) : data_ptr(new int(value)) {}

~my_int_unique_ptr() { delete data_ptr; }

  my_int_unique_ptr(const my_int_unique_ptr &other) {
    other.data = this->data;
    this->data = nullptr;
  }

  my_int_unique_ptr &operator=(const my_int_unique_ptr &other) { … }

private:
  int *data;
};

struct my_int_vector {
  explicit my_int_vector(size_t size) : data_ptr(new int[size]) {}

  ~my_int_vector() { delete[] data_ptr; }

  …
  …

  int *data_ptr;
}
```

RAII

- **R**esource **a**cquisition **i**s **i**nitialization
- https://en.wikipedia.org/wiki/Resource_acquisition_is_initialization
- https://en.cppreference.com/w/cpp/language/raii
- https://en.wikibooks.org/wiki/More_C%2B%2B_Idioms/Resource_Acquisition_Is_Initialization

Úkol Databáze (z posledně)

1. Problém: Jak uložit více typů do jednoho pole
   - Dědičnost
2. Vypsat typovanou hodnotu z x-tého sloupce
   - cout << db.get_col(3).get_row(5);
3. Implementovat std::shared_ptr

```
struct vector_holder {};
struct int_vector_holder : vector_holder {
  std::vector<int> data;
};
struct double_vector_holder : vector_holder {
  std::vector<double> data;
};

vector<std::unique_ptr<vector_holder>> all_data;
```