Lukáš Brchl

Řešení 3 úlohy

Část 1

Od standardu verze C++11 je možné při návratu z funkce využít konstruktu “copy-list-initialization”, který se při volání snaží incializovat návratovou hodnotu funkce. Problém je, že tato konstrukce dokáže vyvolat pouze neexplicitní konstruktor a proto není možné tímto způsobem inicializovat std::tuple, který do standardu verze C++17 má pouze explicitní konstruktor. Nyní v C++17 je to již opraveno (má neexplicitní konstruktor) a je jej tedy možné při návratu incilizovat stejně jako std::pair.

Zdroje:
http://en.cppreference.com/w/cpp/language/return
http://en.cppreference.com/w/cpp/language/list_initialization
http://en.cppreference.com/w/cpp/utility/tuple/tuple
http://en.cppreference.com/w/cpp/utility/pair/pair
https://stackoverflow.com/questions/32084706/why-initialization-doesnt-work-for-tuple
https://stackoverflow.com/questions/13461027/why-does-the-standard-differentiate-between-direct-list-initialization-and-copy/13461382#13461382

Část 2

Funkce std::tie zkonstruuje a vrací std::tupple obsahující reference na argumenty volání. Dále má std::tuple přetížený operátor =, který přijímá tupple a přiřazuje jeho hodnoty do referencí na pravé straně, kterých musí být stejný počet.

Příklad lze tedy upravit na:

```
std::tuple<int&, int&>(q, r) = quot_and_rem(10, 3);
```

Zdroje:
http://en.cppreference.com/w/cpp/utility/tuple/tie
http://en.cppreference.com/w/cpp/utility/tuple/operator%3D
https://stackoverflow.com/questions/43762651/how-does-stdtie-work
https://stackoverflow.com/questions/19800303/what-is-the-difference-between-assigning-to-stdtie-and-tuple-of-references
https://marblemice.wordpress.com/2007/03/07/c-how-the-tie-function-works/