

# Zaawansowane C++

## Lista 3: Move semantics, r-,l-values

### Zadanie 1 Reguła trzech i reguła pięciu

W szablonie klasy `cpplab::vector` napisz

- Wielką trójkę tj. konstruktor kopiujący, kopiujący operator `=` z argumentami typu `const l-value reference` oraz destruktor
- Wielką piątkę, tj. wielka trójka plus konstruktor przenoszący oraz przenoszący operator `=` z argumentami typu `r-value reference`

Nie używaj `memcpy`. Jak zwykle w funkcji `main()` przetestuj działanie wszystkich konstruktorów.

### Zadanie 2 `emplace_back()`

Do klasy `cpplab::vector` dodaj metodę `emplace_back()`, której zadaniem będzie przyjmowanie argumentów mogących stworzyć obiekty typu `T`, gdzie `T` to typ elementów przechowywanych przez dany wektor. Wykorzystaj parameter pack i perfect forwarding, aby z przekazanych argumentów bez zbędnej kopii stworzyć w wektorze nowy, dodany na koniec obiekt typu `T`.

Funkcję `emplace_back()` przetestuj na wektorze składającym się z elementów klasy `pixel` o trzech polach `int r, int g, int b`; np. `v.emplace_back(3,4,6)`

### Materiały pomocnicze:

- r-,l-value references <https://www.youtube.com/watch?v=fbYknr-HPYE&t=26s>
- move semantics <https://www.youtube.com/watch?v=ehMg6zvXuMY&t=378s>
- przypominam o tym [https://www.youtube.com/watch?v=ryRf4Jh\\_YC0&t=1359s](https://www.youtube.com/watch?v=ryRf4Jh_YC0&t=1359s)
- move assignment operator <https://www.youtube.com/watch?v=OWNeCTd7yQE&t=630s>