

KURS JĘZYKA C++

SORTOWANIE TABLIC

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie.

Zdefiniuj szablon klasy, którego obiekty będą pełnić rolę funkcji sortujących za pomocą porównań i zamian. Aby obiekty mogły zachowywać się jak funkcje, muszą one być obiektami funkcyjnymi, zwanymi też funkto-rami. Definiowanie własnych obiektów funkcyjnych jest możliwe dzięki możliwości przeładowania operatora wywołania `operator()(...)`. Szablon tej klasy powinien mieć parametr określający strategię porównywania (domyślnie niech będzie to strategia $\leq$).

Zaprojektuj i zdefiniuj taką hierarchię klas, która umożliwi sortowanie kilkoma różnymi metodami (sortowa-nie bąbelkowe, przez wstawianie, przez scalanie itp). Udostępniij użytkownikowi–programiście gotowe obiekty tych klas ale zabroń ich tworzenia: w programie nie będzie można utworzyć obiektu takiej klasy, za to będzie można skorzystać z dostarczonych obiektów.

Na koniec napisz zestaw kilku programów testowych, sprawdzających działanie sortujących obiektów funk-cyjnych (posortuj tablicę liczb całkowitych `int []`, tablicę łańcuchów znakowych `string []`, tablicę wskaźni-ków do liczb rzeczywistych `double * []` i tablicę ciągów znakowych `const char * []`).

Wskazówka.

Zdefiniuj szablon funkcji `exch()` do zamiany wartości dwóch obiektów tego samego typu i jego specja-lizowaną wersję dla wskaźników (operacja zamiany jest wykonywana na wskazywanych obiektach) i ciągów znakowych `const char *` (w tym przypadku zamieniamy tylko wskaźniki).

```
template <typename T>
comp (T &, T &);
```

Uwaga.

Porównywanie dwóch obiektów tego samego typu powinno zwracać wartość `bool`. Specjalizowana wersja porównań dotyczy wskaźników (należy porównywać wartości wskazywanych obiektów a nie ich adresy) oraz dla ciągów znakowych typu `const char *` (porównywanie leksykograficzne).

Dobrze przemyśl sposób przekazywania strategii porównywania elementów do obiektu sortującego!