
JĘZYK PROGRAMOWANIA C++

SCHOWEK NA LICZBĘ

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zdefiniuj klasę `WartoscStala`, która będzie służyć do skojarzenia wartości z określoną nazwą. Klasa ta powinna zawierać dwa niepubliczne pola: `nazwa` typu `string` i `wartosc` typu `double`. Zdefiniuj w niej odpowiednie konstruktory (w tym konstruktor bezparametrowy) oraz publiczne metody pozwalające odczytać zawartości tych pól.

Następnie napisz program, który wczyta ze standardowego wejścia liczbę całkowitą $n > 0$ (typ `int`) oznaczającą rozmiar danych, a potem n par *nazwa-wartość* (każda taka para w osobnej linii a nazwa oddzielona od wartości pojedynczą spacją) i stablicuje je w dynamicznie przydzielonej tablicy obiektów. Pamiętaj, żeby na końcu zwolnić przydzieloną pamięć.

```
WartoscStala *tab;
int n;
// ...
cin>>n;
tab = new WartoscStala[n];
// ...
delete[] tab;
```

Kiedy dane będą już stablicowane w pamięci, posortuj je leksykograficznie najpierw względem wartości a potem względem nazw. Po każdym sortowaniu wypisz wszystkie elementy tablicy na standardowe wyjście.

Do porównywania elementów typu `WartoscStala` zdefiniuj oddzielne funkcje, które będą zwracały wartość `true` tylko wtedy, gdy ich argumenty będą podane w kolejności niemalejącej. W samej funkcji sortującej wykorzystaj też oddzielnie zdefiniowaną funkcję `zamienMiejscami` do zamiany dwóch elementów w tablicy.

```
void zamienMiejscami (WartoscStala &pierwszy, WartoscStala &drugi);
bool porownajNazwy (const WartoscStala &pierwszy, const WartoscStala &drugi);
void sortujWgNazw (WartoscStala tab[], int n);
bool porownajWartosci (const WartoscStala &pierwszy, const WartoscStala &drugi);
void sortujWgWartosci (WartoscStala tab[], int n);
```

W samej klasie `WartoscStala` dopisz metodę `opis()`, która będzie potrafiła zaprezentować zawartość obiektu w postaci łańcucha znaków i wykorzystaj ją przy wypisywaniu wyników.

Przetestuj program na danych zapisanych w plikach tekstowych (przetwarzanie potokowe); wyniki także przekieruj do pliku. Zadbaj o to, by w pliku wynikowym nie pojawiały się zbędne komentarze (odpowiednio używaj strumieni wyjściowych `cout` i `cerr`).