

KURS PROGRAMOWANIA W C++

BEZPIECZNE PLIKI I MANIPULATORY

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie 1

W oparciu o technikę *zdobycia zasobów poprzez inicjalizację* zaimplementuj bezpieczne klasy opakowujące pliki: `PlikWejsciowy` dla plików tekstowych do czytania (opakowanie dla obiektu `ifstream`) oraz `PlikWyjsciowy` dla plików tekstowych do pisania (opakowanie dla obiektu `ofstream`). Plik należy otworzyć w konstruktorze (jeśli okaże się to niemożliwe zgłoś wyjątek) a zamknąć w destruktorze. Zadbaj o to, by ustawienie flagi błędu `ios_base::badbit` lub `ios_base::failbit` powodowało automatyczne zgłoszenie wyjątku `ios_base::failure`.

Klasa `PlikWejsciowy` powinna umieć odczytać całą linię danych (wyciągając ze strumienia znak przejścia do nowej linii bez umieszczania go w wynikowym łańcuchu) zwracającą obiekt typu `string`, pojedynczy znak `char` a także liczby całkowitą `int` i rzeczywistą `double` z pominięciem początkowych białych znaków. Klasa `PlikWyjsciowy` powinna umieć zapisać obiekt typu `string`, pojedynczy znak `char`, liczbę całkowitą `int` i rzeczywistą `double` a także znak przejścia do nowej linii. W klasie `PlikWejsciowy` zdefiniuj zaprzyjaźnione operatory do czytania `operator>>` a w klasie `PlikWyjsciowy` operatory do pisania `operator<<`.

Na koniec napisz program, który przetestuje zachowanie się obiektów obu klas (również w sytuacjach wyjątkowych) — program powinien odczytać z pliku ciąg liczb całkowitych i zapisać go w odwrotnej kolejności do pliku tekstowego o tej samej nazwie. Z początku nie wiadomo ile liczb jest zapisanych w pliku z danymi, dlatego posłuż się kontenerem `vector<>`.

Zadanie 2

Zdefiniuj własny manipulator bezparametrowy `clearline` dla strumienia wejściowego, który będzie usuwał wszystkie znaki, aż do napotkania znaku przejścia do nowej linii (ten znak także należy usunąć ze strumienia) lub znaku końca pliku. Zdefiniuj również manipulator z parametrem `ignore (int x)`, którego zadaniem będzie pominięcie `x` znaków ze strumienia wejściowego, chyba że wcześniej zostanie wyjęty znak przejścia do nowej linii lub strumień się skończy.

Dla strumienia wyjściowego zdefiniuj bezparametrowe manipulatory `comma` wypisujący przecinek i odstęp `", "` oraz `colon` wypisujący dwukropek i odstęp `": "`. Zdefiniuj także manipulator z parametrem `index (int x)`, który wypisze liczbę `x` w nawiasach kwadratowych.

Na koniec napisz program testujący zdefiniowane przez siebie manipulatory — program powinien odczytać wszystkie linie danych zapamiętując je w kontenerze `vector<>`. Następnie posortuj odczytane linie leksykograficznie i wypisz je wraz z pierwotnymi numerami linii w nawiasach kwadratowych dołączonymi na początku.