

kurs języka C++

szyfr Cezara

Instytut Informatyki
Uniwersytetu Wrocławskiego

Paweł Rzechonek

Prolog

Dane przetwarzane przez komputery mogą być cyfrową reprezentacją dowolnych informacji: tekstowych, obrazowych, dźwięków, filmów itp. Dane są pamiętane w postaci plików na nośnikach elektronicznych. Plik to uporządkowana sekwencja danych o skończonej długości. Dane w plikach mogą mieć reprezentację tekstową (czytelną i łatwą w edycji), albo binarną (przystosowaną do bezpośredniego przetwarzania przez procesor). Tak czy inaczej plik jest podstawową jednostką danych w systemie plików.

Zadanie

Zdefiniuj w oparciu o technikę RAII (zdobywanie zasobów poprzez inicjalizację) bezpieczne klasy opakujące pliki tekstowe: `wejscie` dla plików tekstowych do czytania (opakowanie dla obiektu `ifstream`) oraz `wyjście` dla plików tekstowych do pisania (opakowanie dla obiektu `ofstream`). Plik należy otworzyć w konstruktorze (jeśli okaże się to nie możliwe zgłoś wyjątek) a zamknąć w destruktorze. Ustawienie flagi błędu `ios_base::badbit` lub `ios_base::failbit` ma spowodować zgłoszenie wyjątku `ios_base::failure`.

Klasa `wejscie` powinna posiadać tylko jedną funkcję składową `nastlinia()` do odczytywania kolejnych linii tekstu z pliku; funkcja ta ma zwrócić łańcuch znakowy typu `string`. Natomiast klasa `wyjście` powinna posiadać tylko jedną funkcję składową `piszlinie()` do zapisywania kolejnych linii tekstu do pliku; funkcja ta ma otrzymać jako argument łańcuch znakowy typu `string`.

Na koniec napisz program testujący zdefiniowane przez siebie klasy, wykorzystując je do kopiowania plików połączonego z szyfrowaniem. Program ma zaszyfrować wskazany plik za pomocą szyfru Cezara – należy szyfrować tylko litery zapisane w pliku tekstowym za pomocą danego przesunięcia; przesunięcie może być dowolną liczbą całkowitą, którą należy znormalizować za pomocą operacji modulo 26, aby otrzymać przesunięcie o wartości z zakresu od 0 do 25. Parametrami programu powinny być: nazwa pliku do zaszyfrowania, nazwa pliku wynikowego i klucz szyfrujący (liczba całkowita).

Ważne elementy programu

- Podział programu na pliki nagłówkowe i źródłowe.
- Definicja klas opakujących strumienie wejście i wyjście zgodnie z techniką RAII.
- Obsługa błędów w strumieniach za pomocą wyjątków.
- Implementacja szyfru Cezara.
- W funkcji `main()` należy przetestować podane argumenty wywołania.