

zaawansowane techniki C++ i STL**operacje w lokalnym systemie plików**

Instytut Informatyki
Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie 1 (1 pkt)

Przygotuj plik z danymi `liczby.txt`, w którym umieścisz ciąg dodatnich liczb rzeczywistych pooddzielanych białymi znakami (co najmniej 10). Następnie napisz program, który używając iteratora strumieniowego `istream_iterator` odczyta wszystkie liczby z tego pliku, wstawi je do wektora a potem policzy średnią arytmetyczną i odchylenie standardowe dla tego zbioru danych. Wyniki wypisz na standardowym wyjściu z dokładnością do 3 miejsc po kropce dziesiętnej.

Zadanie 2 (1 pkt)

Niech $\Phi(n)$ oznacza funkcję Eulera (tocjent), która dla każdej dodatniej liczby naturalnej zwraca liczbę liczb naturalnych nie większych od n i względnie pierwszych z n . Dla zadanej wartości k stabiluj kolejne wartości tocjenta $\Phi(1), \Phi(2), \dots, \Phi(k)$ a następnie zapisz je do pliku `phi.txt`, używając iteratora strumieniowego `ostream_iterator` z separatorem w postaci średnika i spacji ";" ". Wartość k przekaż do programu poprzez argumenty wywołania.

Zadanie 3 (1 pkt)

Napisz program, który sprawdzi czy w lokalnym systemie plików istnieją podane pliki lub katalogi. Jeśli tak, to wypisz ich najważniejsze parametry (ścieżka kanoniczna, data ostatniej modyfikacji, rozmiar dla pliku itp.). Nazwy plików lub katalogów do sprawdzenia przekaż do programu poprzez argumenty wywołania.

Zadanie 4 (1 pkt)

Napisz program, który wypisze zawartość wskazanego katalogu w lokalnym systemie plików. Jeśli podany przez użytkownika katalog nie istnieje, to wypisz stosowny komunikat na standardowe wyjście dla błędów; jeśli podany katalog istnieje, to wypisz jego zawartość w kolejności alfabetycznej (najpierw wypisz katalogi a potem pliki). Wypisując zawartość katalogu użyj iteratora plikowego `directory_iterator`. Nazwę katalogu pobierz ze standardowego wejścia (można to robić w pętli).

Zadanie 5 (2 pkt)

Napisz program, który policzy i wypisze sumę rozmiarów wszystkich plików we wskazanym katalogu i jego podkatalogach. Jeśli podany przez użytkownika katalog nie istnieje, to wypisz stosowny komunikat na standardowe wyjście dla błędów; jeśli podany katalog istnieje, to policz ile jest w nim plików i posumuj ich rozmiary. Analizując zawartość katalogu i podkatalogów postuż się rekurencyjnym iteratorem plikowym `recursive_directory_iterator`. Nazwę katalogu pobierz ze standardowego wejścia (można to robić w pętli).

Zadanie 6 (2 pkt)

Napisz program, który przeglądnie wskazany katalog i jego podkatalogi w poszukiwaniu projektów realizowanych w Code::Blocks (stare IDE do C++). Jeśli w jakimś podkatalogu znajdzie się plik z rozszerzeniem .cbp, to z tego podkatalogu należy usunąć katalog bin wraz z całą jego zawartością. Analizując zawartość katalogu i podkatalogów posłuż się rekurencyjnym iteratorem plikowym recursive_directory_iterator. Przed każdym usunięciem wypisz komunikat na standardowe wyjście dla błędów cerr o lokalizacji usuwanego katalogu. Nazwę katalogu pobierz ze standardowego wejścia (można to robić w pętli).

Zadanie 7 (2 pkt)

Zaprojektuj i zaimplementuj prosty mechanizm serializacji i deserializacji dla wartości podstawowych i prostych obiektów w C++. Serializacja i deserializacja ma pracować na strumieniach binarnych. Zaimplementuj zestaw funkcji do serializacji i deserializacji dla:

1. wszystkich typów podstawowych (bool, char, short, int, long long, float, double i long double);
2. łańcuchów znakowych typu string;
3. tablic wymienionych powyżej typów danych – przy serializacji określ rozmiar tablicy a przy deserializacji należy utworzyć nową tablicę na stercie operatorem new[], wypełnić ją danymi i zwrócić parę: wskaźnik do tablicy i jej rozmiar.