

# JĘZYK PROGRAMOWANIA C++

## MANIPULATORY I PLIKI

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

---

---

*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. Plik jest podstawową jednostką danych w każdym systemie plików.

### Zadanie 1.

Dla strumienia wejściowego `istream` zdefiniuj własny manipulator bezparametrowy `clearline`, 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.

### Zadanie 2.

Dla strumienia wyjściowego `ostream` zdefiniuj bezparametrowe manipulatory `comma` wypisujący przecinek z odstępem `,` oraz `colon` wypisujący dwukropek z odstępem `:`. Zdefiniuj także manipulator z parametrem `index (int x, int w)`, który wypisze liczbę `x` w nawiasach kwadratowych i na liczbę tą przeznaczy co najmniej w pozycji (dosuń liczbę do prawego nawiasu kwadratowego).

### Zadanie 3.

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. Numer linii umieść na początku wiersza w nawiasach kwadratowych (numeracja wierszy ma się zaczynać od 1).

### Zadanie 4.

W oparciu o technikę *zdobycia zasobów poprzez inicjalizację* zaimplementuj bezpieczne klasy opakowujące pliki: `wejscie` dla plików binarnych do czytania (opakowanie dla obiektu `ifstream`) oraz `wyjście` dla plików binarnych 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 `wejście` powinna umieć odczytać jeden bajt danych (wyciągając ze strumienia jeden znak) zwracając wartość typu `int`. Natomiast klasa `wyjście` powinna umieć zapisać jeden bajt, dostając wartość typu `int` z zakresu od 0 do 255 (bardziej znaczące bity ignorujemy).

W klasie `eejście` zdefiniuj zaprzyjaźniony operator do odczytania bajtu danych z pliku `operator>>` a w klasie `wyjście` operator do zapisania jednego bajtu `operator<<`.

### **Zadanie 5.**

Napisz program, który przetestuje zachowanie się obiektów obu klas plikowych (również w sytuacjach wyjątkowych) — program powinien odczytać zadany plik bajt po bajcie, dokonać transformacji tych bajtów (na przykład zanegować bity albo odwrócić bity, czyli zamienić pierwszy z ostatnim, drugi z przedostatnim itd.) i zapisać przekształcone bajty do drugiego pliku binarnego. Nazwy plików przekaż do programu jako parametry wywołania.

### **Uzupełnienie.**

Definicje manipulatorów i klas opakujących pliki umieść w przestrzeni nazw `strumienie`.

### **Elementy w programie, na które należy zwrócić szczególną uwagę.**

- Definicja manipulatorów bezparametrowych.
- Definicja manipulatorów z parametrami.
- Testowanie manipulatorów.
- Wrappery dla plików realizujące ideę zdobywania zasobów poprzez inicjalizację.
- Obsługa błędów w strumieniach za pomocą wyjątków.
- Testowanie wrapperów strumieniowych.