

Podstawy Programowania Komputerów

Standard Template Library – krótkie wprowadzenie

Krzysztof Simiński

2018

Spis treści

1	Streams	1
1.1	<ios>	1
1.1.1	Klasa <code>std::ios_base</code>	1
1.1.2	Klasa <code>std::ios</code>	4
1.1.3	Funkcje	4
1.2	<code>iostream</code> , <code>fstream</code> , <code>sstream</code>	4
1.3	<iomanip>	4

1 Streams

Ten rozdział opisuje pokrótce podstawowe cechy strumieni.

Strumień jest abstrakcyjną reprezentacją źródła lub celu nieskończonego ciągu bajtów. Zwykle strumienie kojarzy się z urządzeniami fizycznymi, np. plikami, ekranem, klawiaturą, drukarką, gniazdem sieciowym itd.

Strumienie są zdefiniowane w następujących bibliotekach:

- <ios>, <iostream>, <ostream>, <streambuf> i <iosfwd> definiują podstawowe klasy strumieniowe, ale zwykle nie są dołączane w programach bezpośrednio – są dołączane przez inne biblioteki strumieniowe.
- <iostream> jest podstawową biblioteką operacji wejścia-wyjścia. Biblioteka definiuje standardowe strumienie: wejścia (**cin**) i wyjścia (**cout**, **cerr**, **clog**).
- <fstream> definiuje strumienie plikowe.
- <sstream> definiuje strumienie napisowe.
- <iomanip> definiuje standardowe manipulatory do modyfikacji flag i formatowania strumieni.

Rys. 1 przedstawia hierarchię klas strumieniowych.

1.1 <ios>

Biblioteka definiuje klasy `ios_base` i `ios`, i funkcje.

1.1.1 Klasa `std::ios_base`

Klasa `ios_base` definiuje wspólne składowe strumieni. Klasa nie ma publicznych konstruktorów, więc nie można zdefiniować obiektu klasy `ios_base`. Klasa definiuje szereg klas – tab. 1. Podstawowe metody klasy są zebrane w tab. 2.

Tablica 1: Flagi zdefiniowane w bibliotece **ios_base**.

flagi	
std::ios_base::boolalpha	wyświetlanie wartości logicznych jako true lub false
std::ios_base::showbase	poprzedzanie liczb całkowitych podstawą systemu
std::ios_base::showpoint	wyświetlanie zawsze kropki dziesiętnej
std::ios_base::showpos	wyświetlanie zawsze znaku + dla liczb dodatnich
std::ios_base::skipws	pomijanie białych znaków
std::ios_base::uppercase	użycie wielkich liter
flagi	
std::ios_base::oct	użycie systemu ósemkowego dla liczb całkowitych
std::ios_base::dec	użycie systemu dziesiętnego dla liczb całkowitych
std::ios_base::hex	użycie systemu szesnastkowego dla liczb całkowitych
flagi dla liczb zmiennoprzecinkowych	
std::ios_base::fixed	użycie notacji niewykładniczej
std::ios_base::scientific	użycie notacji wykładniczej
wyrównanie	
std::ios_base::internal	wypełnienie przestrzeni między znakiem a liczbą znakami wypełniającymi
std::ios_base::left	wyrównanie do lewej
std::ios_base::right	wyrównanie do prawej
kombinacje flag	
std::ios_base::adjustfield	left right internal
std::ios_base::basefield	oct dec hex
std::ios_base::floatfield	scientific fixed

Tablica 2: Metody klasy **ios_base**.

flags	pobierz lub usta flagi formatu (tab. 1)
setf	ustaw wartość flagi (tab. 1)
unsetf	wyczyść wartość flagi (tab. 1)
precision	pobierz lub ustaw precyzję liczb zmiennoprzecinkowych
width	pobierz lub ustaw szerokość pola wypisywania

Tablica 3: Metody klasy **ios**.

good	czy stan strumienia jest good
eof	sprawdzenie flagi eofbit
fail	sprawdzenie flagi failbit lub badbit
bad	sprawdzenie flagi badbit
operator!	true , jeżeli failbit lub badbit jest ustawiony, w przeciwnym razie false
rdstate	odczytanie stanu strumienia
setstate	ustaw stan strumienia
clear	wyczyść flagi stanu strumienia
fill	pobierz lub ustaw znak wypełniający

Tablica 4: Flagi strumienia

flaga	przyczyna	good()	eof()	fail()	bad()
goodbit	brak błędów	true	false	false	false
eofbit	koniec strumienia	false	true	false	false
failbit	błąd logiczny	false	false	true	false
badbit	błąd odczytu / zapisu	false	false	true	true

1.1.2 Klasa **std::ios**

Klasa definiuje flagi formatowania strumieni. Niektóre z nich są wymienione poniżej:

- **eofbit** – ostatnia operacja strumieniowa dotarła do końca strumienia
- **failbit** – błąd logiczny, dalsze operacje na strumieniu mogą być możliwe
- **badbit** – błąd odczytu lub zapisu (zwykle utrata spójności strumienia), dalsze operacje na strumieniu niemożliwe
- **goodbit** równy zero oznacza, że żaden z innych bitów błędów nie jest ustawiony.

Metody klasy przedstawia tab. 3.

1.1.3 Funkcje

Podstawowe funkcje są zebrane w tablicy tab. 5.

1.2 **iostream**, **fstream**, **sstream**

Klasa **iostream** odpowiada zarówno za operacje wyjścia i wejścia. Podstawowe metody przedstawia tab. 6. Klasy **fstream** i **stringstream** dziedziczą wszystkie metody po klasie **iostream**.

1.3 **<iomanip>**

Biblioteka dostarcza manipulatory strumienia – tab. 7.

Tablica 5: Metody zdefiniowane w `<ios>`.

flagi		
<code>boolalpha</code>	<code>noboolalpha</code>	ustaw / wyczyść flagę <code>std::ios_base::boolalpha</code>
<code>showbase</code>	<code>noshowbase</code>	ustaw / wyczyść flagę <code>std::ios_base::showbase</code>
<code>showpoint</code>	<code>noshowpoint</code>	ustaw / wyczyść flagę <code>std::ios_base::showpoint</code>
<code>showpos</code>	<code>noshowpos</code>	ustaw / wyczyść flagę <code>std::ios_base::showpos</code>
<code>skipws</code>	<code>noskipws</code>	ustaw / wyczyść flagę <code>std::ios_base::skipws</code>
<code>uppercase</code>	<code>nouppercase</code>	ustaw / wyczyść flagę <code>std::ios_base::uppercase</code>
flagi		
<code>oct</code>		użyj zapisu ósemkowego
<code>dec</code>		użyj zapisu dziesiętnego
<code>hex</code>		użyj zapisu szesnastkowego
flagi dla formatowania liczb zmiennoprzecinkowych		
<code>fixed</code>		nie używaj notacji wykładniczej
<code>scientific</code>		użyj notacji wykładniczej
wyrównanie		
<code>internal</code>		wypełnij przestrzeń między znakiem a liczbą znakami wypełniającymi
<code>left</code>		wyrównaj do lewej
<code>right</code>		wyrównaj do prawej

Tablica 6: Metody klasy `iostream`.

operacje wejściowe	
<code>operator>></code>	operator sformatowanego wejścia
<code>gcount</code>	pobierz liczbę znaków
<code>get</code>	pobierz znak
<code>getline</code>	pobierz linię
<code>ignore</code>	pobierz i usuń znaki
<code>peek</code>	pobierz następny znak
<code>read</code>	wczytaj blok danych
<code>tellg</code>	pobierz pozycję w strumieniu wejściowym
<code>seekg</code>	ustaw pozycję w strumieniu wejściowym
operacje wyjściowe	
<code>operator<<</code>	operator sformatowanego wyjścia
<code>put</code>	wstaw znak do strumienia
<code>write</code>	zapisz blok danych
<code>tellg</code>	pobierz pozycję w strumieniu wyjściowym
<code>seekg</code>	ustaw pozycję w strumieniu wyjściowym
<code>flush</code>	opróżnij bufor do strumienia

Tablica 7: Funkcje zdefiniowane w bibliotece `<iomanip>`.

<code>setiosflags</code>	ustaw flagi formatowania (tab. 1)
<code>resetiosflags</code>	wyczyść flagi formatowania (tab. 1)
<code>setbase</code>	ustaw podstawę dla liczb całkowitych (8, 10, 16)
<code>setfill</code>	ustaw znak wypełniający
<code>setprecision</code>	ustaw precyzję dla liczb zmiennoprzecinkowych
<code>setw</code>	ustaw szerokość pola
<code>get_money</code>	wczytaj liczbę w formacie pieniądza
<code>put_money</code>	zapisz liczbę w formacie pieniądza
<code>get_time</code>	wczytaj liczbę w formacie czasu
<code>put_time</code>	zapisz liczbę w formacie czasu