

Cel ćwiczenia: **utrwalenie wiedzy na temat konstruktorów i destruktorów,**
 zapoznanie się z przeładowaniem operatorów,
 zapoznanie się z funkcjami konwertującymi,
 poprawny podział projektu na pliki

Należy napisać klasę `ciag`, która będzie umożliwiała przechowywanie ciągu znakowego. Wewnątrz klasy ciąg znakowy ma być przechowywany w dynamicznie alokowanej tablicy znakowej utworzonej pod wskaźnikiem, który jest zmienną prywatną klasy. Klasa ma dysponować:

1. konstruktorem domyślnym,
2. konstruktorem z jednym argumentem typu `char*` (tablica znakowa),
3. konstruktorem kopiującym,
4. destruktorze,
5. bezargumentową funkcją `dl` zwracającą długość aktualnie przechowywanego ciągu,
6. przeładowanym operatorem przypisania,
7. przeładowanym operatorem `+=` (`A += B` oznacza `A = A + B`),
8. funkcję konwertującą na typ `char*` (aby obiekt klasy `ciag` mógł być przekazywany do funkcji oczekujących jako argumentu zwykłej tablicy znakowej),
9. statyczną bezargumentową funkcję `ile` zwracającą ilość obiektów klasy `ciag` aktualnie znajdujących się w pamięci (trzeba utworzyć odpowiednią zmienną statyczną i inkrementować/dekrementować ją w konstruktorze/destruktorze),
10. przeładowanym jako zaprzyjaźniona funkcja globalna operatorem `+` dodającym do siebie dwa ciągi znakowe i zwracającym ciąg powstały przez ich połączenie,
11. przeładowanymi jako zaprzyjaźnione funkcje globalne operatorami `<<` oraz `>>` obsługującymi wejście i wyjście.

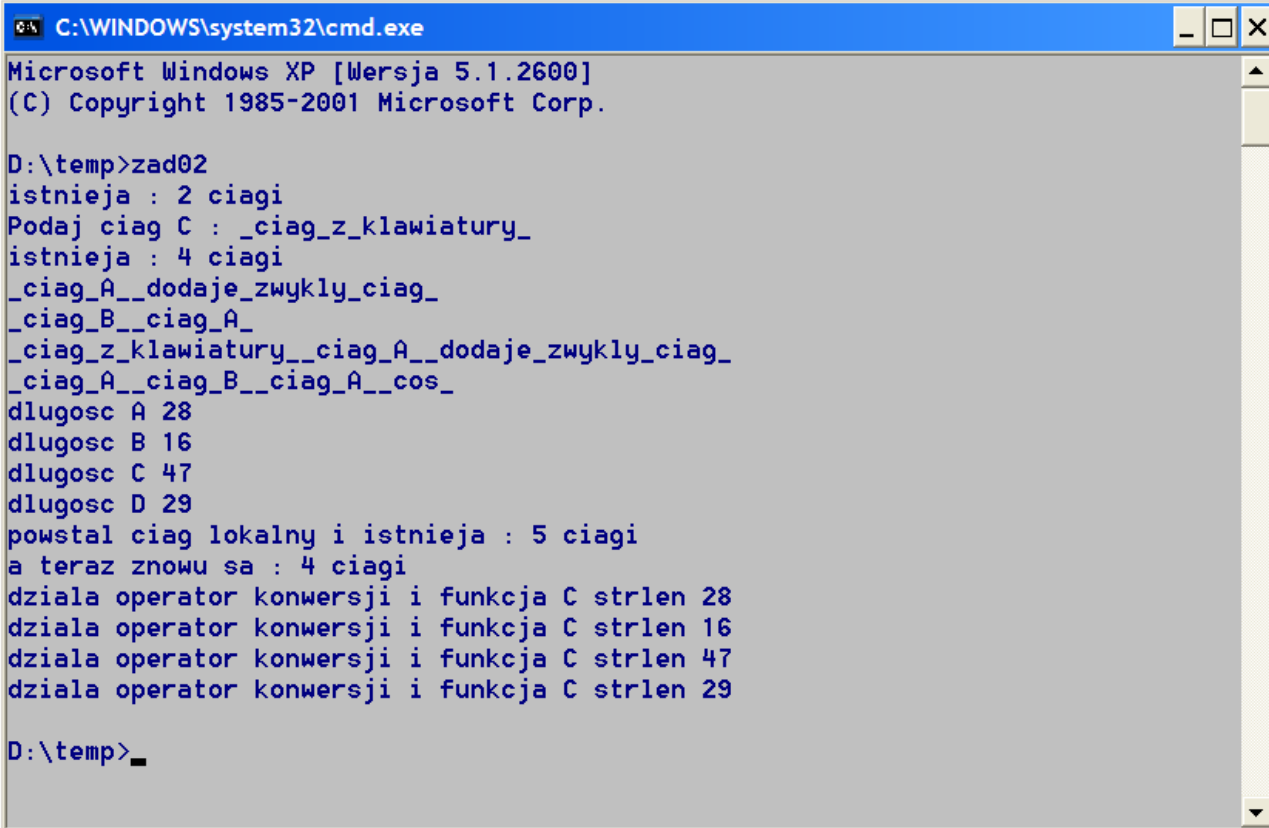
Program należy podzielić na trzy pliki:

1. plik `ciag.h` zawierający definicję klasy `ciag`
2. plik `ciag.cpp` zawierający definicje funkcji składowych klasy `ciag`
3. plik `ciag_main.cpp` zawierający funkcję główną (podaną poniżej) demonstrującą działanie klasy `ciag`

Poniżej podany jest kod funkcji `main`, który ma posłużyć jako test poprawności klasy `ciag` oraz zrzut ekranu przedstawiający poprawny rezultat zadziałania tej funkcji `main`:

```
/*  
Zadanie Drugie - Cwiczenie: konstruktory, destruktory, przeładowanie operatorow, funkcje  
konwertujace  
*/  
  
#include <iostream>  
#include <cstring>  
#include "zad02.h"  
using namespace std;
```

```
int main()
{
    ciag A("_ciag_A_");
    ciag B;
    B = "_ciag_B_";
    B += A;
    cout << "istnieje : " << ciag::ile() << " ciagi" << endl;
    ciag C;
    cout << "Podaj ciag C : ";
    cin >> C;
    ciag D;
    D = A + B + "_cos_";
    cout << "istnieje : " << ciag::ile() << " ciagi" << endl;
    A += "_dodaje_zwykly_ciag_";
    C += A;
    cout << A << endl << B << endl << C << endl << D << endl;
    cout << "dlugosc A " << A.dl() << endl;
    cout << "dlugosc B " << B.dl() << endl;
    cout << "dlugosc C " << C.dl() << endl;
    cout << "dlugosc D " << D.dl() << endl;
    {
        ciag ZZ;
        cout << "powstal ciag lokalny i istnieje : " << ciag::ile()
            << " ciagi" << endl;
    }
    cout << "a teraz znowu sa : " << A.ile() << " ciagi" << endl;
    cout << "dziala operator konwersji i funkcja C strlen "
        << strlen(A) << endl;
    cout << "dziala operator konwersji i funkcja C strlen "
        << strlen(B) << endl;
    cout << "dziala operator konwersji i funkcja C strlen "
        << strlen(C) << endl;
    cout << "dziala operator konwersji i funkcja C strlen "
        << strlen(D) << endl;
    return 0;
}
```



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Wersja 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\temp>zad02
istnieje : 2 ciagi
Podaj ciag C : _ciag_z_klawiatury_
istnieje : 4 ciagi
_ciad_A__dodaje_zwykly_ciag_
_ciad_B__ciad_A_
_ciad_z_klawiatury__ciad_A__dodaje_zwykly_ciag_
_ciad_A__ciad_B__ciad_A__cos_
dlugosc A 28
dlugosc B 16
dlugosc C 47
dlugosc D 29
powstal ciag lokalny i istnieje : 5 ciagi
a teraz znowu sa : 4 ciagi
dziala operator konwersji i funkcja C strlen 28
dziala operator konwersji i funkcja C strlen 16
dziala operator konwersji i funkcja C strlen 47
dziala operator konwersji i funkcja C strlen 29

D:\temp>
```