

Praca na plikach:

Biblioteki do zadań z plikami:

Aby pracować z plikami na dysku, musimy dołączyć do programu bibliotekę obsługującą pliki:

```
#include <fstream>
```

W niej utworzone są klasy dzięki którym możesz tworzyć obiekty plikowe.

Tworzenie obiektów plikowych

Obiekt plikowy tworzymy przez stworzenie zmiennej plikowej, dla której typem zmiennej jest nazwa klasy

```
ifstream plik;
```

typ zmiennej

nazwa zmiennej

Własności obiektu lub funkcje możliwe do wykonania na obiekcie wymieniane są po kropce za nazwą zmiennej.

```
plik.open( "C:\\odczyt.txt" );
```

przykład funkcji otwierania pliku z parametrem ścieżki dostępu.

Tryby dostępu do pliku

Tryb dostępu określa nam uprawnienia programu, np.: tylko do odczytu, możliwość zapisywania itp.:

```
plik.open( "nazwa_pliku.txt", std::ios::in | std::ios::out );
```

ścieżka dostępu

tryb odczytu

tryb zapisu

Ilość uprawnień (trybów) można zwiększać przez łączenie ich znakiem „|”

Tryb	Opis trybu
ios::app	(append - dopisywanie danych do pliku) Ustawia wewnętrzny wskaźnik zapisu pliku na jego koniec. Plik otwarty w trybie tylko do zapisu. Dane mogą być zapisywane tylko i wyłącznie na końcu pliku.
ios::ate	(at end) Ustawia wewnętrzny wskaźnik pliku na jego koniec w chwili otwarcia pliku.
ios::binary	(binary) Informacja dla kompilatora, aby dane były traktowane jako strumień danych binarnych, a nie jako strumień danych tekstowych.
ios::in	(input - wejście/odczyt) Zezwolenie na odczytywanie danych z pliku.
ios::out	(output - wyjście/zapis) Zezwolenie na zapisywanie danych do pliku.
ios::trunc	(truncate) Zawartość pliku jest tracona, plik jest obcinany do 0 bajtów podczas otwierania.

Otwieranie i zamykanie pliku:

Operację tę można wykonać w momencie gdy plik istnieje i mamy do niego uprawnienia. Po otwarciu i wykonaniu operacji na plikach plik powinien być zamknięty, by niepotrzebnie nie blokował pamięci. Aby uniknąć zamykania programu z błędem krytycznym możemy stosować następującą procedurę:

```
1  #include <iostream>
2  #include <fstream>
3
4  using namespace std;
5
6  int main()
7  {
8      fstream plik;
9      plik.open( "nazwa_pliku.txt", ios::in | ios::out );
10     //przykładowe ustawienie trybów otwarcia
11     if( plik.good() == true )
12     {
13         cout << "Uzyskano dostep do pliku!" << endl;
14         //operacje na pliku (zapis/odczyt)
15
16         plik.close(); // zamykanie pliku
17     } else cout << "Brak dostepu!" <<endl;
18     return( 0 );
19 }
20
```

W wierszu **11** używamy funkcji **good** która zwraca na wartość **True** w przypadku prawidłowego otworzenia pliku.

W wierszy **16** zamykamy plik po wykonaniu operacji plikowych.

Pobieranie danych z pliku tekstowego i wypisywanie ich na ekranie:

```
while(!plik.eof()){
    getline( plik, dane );
    cout << dane <<endl;
}
```

Za pomocą pętli dopóki nie wystąpi znak końca pliku **eof()** pobieramy linia po linii i wypisujemy na ekranie.

Dopisywanie do pliku tekstowego.

```
string tekst;  
cin >> tekst;  
if( plik.good() == true )  
{  
    getline( cin, tekst );  
    plik.write( & tekst[ 0 ], tekst.length() );  
  
    plik.close(); // zamykanie pliku  
-} else cout << "Brak dostępu!" <<endl;
```

Do zapisu danych blokami wykorzystana zostanie funkcja **write()**. W poniższym przykładzie zmienna **tekst** (typu string) pełni rolę bufora (&tekst[0] do pierwszego elementu zmiennej) natomiast **tekst.length()** mówi kompilatorowi, żeby skopiował ilość danych równą długości bufora tekst.