

Temat: Obsługa plików w języku C++

Klasa `fstream`:

- ▶ Klasa **`ifstream`** umożliwia obsługę strumienia wejściowego.
- ▶ Klasa **`ofstream`** umożliwia obsługę strumienia wyjściowego.
- ▶ Klasa **`fstream`** umożliwia obsługę strumienia wejściowego i wyjściowego. (**`#include <fstream>`**)
 - ▶ Otwarcie pliku:
`plik.open("nazwa_pliku",tryb_otwarcia_pliku);`
 - ▶ Nazwa pliku może być podana z pełną ścieżką dostępu do pliku.
 - ▶ Jeśli ścieżka do pliku nie zostanie podana, to będzie on poszukiwany w katalogu roboczym.
 - ▶ Odczyt i zapis do pliku przypomina korzystanie z obiektów **`cin`** oraz **`cout`**.
 - ▶ Dokumentacja: <http://www.cplusplus.com/reference/fstream/>

Tryb otwarcia pliku:

- ▶ **`ios::app`** – dołączanie do pliku,
- ▶ **`ios::ate`** – ustawienie bieżącej pozycji na końcu pliku,
- ▶ **`ios::trunc`** – usunięcie całej zawartości pliku ,
- ▶ **`ios::out`** – umożliwia zapis do pliku,
- ▶ **`ios::in`** – umożliwia odczyt z pliku,
- ▶ **`ios::binary`** – umożliwia operacje binarne.

Opcje dotyczące zmiany pozycji w pliku:

- ▶ **`ios::beg`** – początek pliku,
- ▶ **`ios::cur`** – pozycja bieżąca pliku,
- ▶ **`ios::end`** – koniec pliku.

Zapisywanie do pliku:

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  int main()
6  {
7      fstream plik;
8      int liczba=133;
9      plik.open("a.txt", ios::out);
10     plik<<liczba<<endl;
11     plik.close();
12     cout<<"Zapisano liczbe do pliku \"a.txt\".";
13     return 0;
14 }
```

Zapisano liczbe do pliku "a.txt".

a.txt	
1	133

Dopisywanie do pliku:

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  int main()
6  {
7      fstream plik;
8      int a1=11, a2=222;
9      plik.open("a.txt", ios::out|ios::app);
10     plik<<a1<<endl<<a2<<endl;
11     plik.close();
12     cout<<"Dopisano liczby do pliku \"a.txt\".";
13     return 0;
14 }
```

Dopisano liczby do pliku "a.txt".

a.txt	
1	133
2	11
3	222

Odczytywanie z pliku dwóch liczb:

```
1 #include <iostream>
2 #include <fstream>
3 #include <cstdlib> //exit()
4 using namespace std;
5
6 int main()
7 {
8     fstream plik;
9     int a1, a2;
10    plik.open("a.txt", ios::in);
11    if(plik.good() == false) {
12        cout<<"Brak pliku \"a.txt\".";
13        exit(0);
14    }
15    plik>>a1;
16    plik>>a2;
17    plik.close();
18    cout<<"Odczytano "<<a1<<" i "<<a2<<". "<<endl;
19    return 0;
20 }
```

Odczytano 133 i 11.

Odczytywanie z pliku wszystkich liczb:

```
1 #include <iostream>
2 #include <fstream>
3 #include <cstdlib> //exit()
4 using namespace std;
5
6 int main()
7 {
8     fstream plik;
9     int a;
10    plik.open("a.txt", ios::in);
11    if(plik.good() == false) {
12        cout<<"Brak pliku \"a.txt\".";
13        exit(0);
14    }
15    while(plik>>a) { //dopoki jest co czytac
16        cout<<a<< ", ";
17    }
18    plik.close();
19    return 0;
20 }
```

133, 11, 222,

Odczytywanie z pliku linia po linii:

```
1  #include <iostream>
2  #include <fstream>
3  #include <cstdlib> //exit()
4  using namespace std;
5
6  int main()
7  {
8      fstream plik;
9      string linia;
10     plik.open("b.txt", ios::out);
11     plik<<"Jan Adam Kot\n20\nEwa Kos\n21\n";
12     plik.close();
13     plik.open("b.txt", ios::in);
14     if(plik.good()==true) {
15         while(getline(plik, linia))
16             cout<<linia<<endl;
17     }
18     plik.close();
19     return 0;
20 }
```

```
Jan Adam Kot
20
Ewa Kos
21
```

Odczytywanie z pliku liczb rzeczywistych:

```
1  #include <iostream>
2  #include <fstream>
3  #include <cstdlib> //atof()
4  using namespace std;
5
6  int main()
7  {
8      fstream plik;
9      string linia;
10     double liczba;
11     plik.open("c.txt", ios::out|ios::trunc);
12     plik<<"1.2\n3.5\n5.6\n1.8\n2.5\n3.2";
13     plik.close();
14     plik.open("c.txt", ios::in);
15     if(plik.good()==true) {
16         while(!plik.eof()) {
17             getline(plik, linia);
18             liczba=atof(linia.c_str());
19             cout<<liczba<<endl;
20         }
21     }
22     plik.close();
23     return 0;
24 }
```

```
1.2
3.5
5.6
1.8
2.5
3.2
```

Odczytywanie z pliku linia po linii:

```
1  #include <iostream>
2  #include <fstream>
3  #include <cstdlib> //atoi()
4  using namespace std;
5
6  int main()
7  {
8      fstream plik;
9      string linia;
10     int liczba, nr=1;
11     plik.open("b.txt", ios::out|ios::trunc);
12     plik<<"Jan Emil Kot\n20\nEwa Kos\n22\n";
13     plik.close();
14     plik.open("b.txt", ios::in);
15     if(plik.good()==true) {
16         while(getline(plik, linia)) {
17             if(nr++%2==0) {
18                 liczba=atoi(linia.c_str());
19                 cout<<liczba<<endl;
20             } else
21                 cout<<linia<<endl;
22         }
23     }
24     plik.close();
25     return 0;
26 }
```

```
Jan Emil Kot
20
Ewa Kos
22
```


Zadania przykładowe:

1. Do pliku tekstowego o nazwie *liczby.txt* wpisz pewną ilość liczb naturalnych (jednak nie mniej niż 100 i nie więcej niż 1000). Wpisywane liczby mają być z zakresu od 1 do 100. Użyj generatora liczb pseudolosowych.

```

1  #include <iostream>
2  #include <fstream>
3  #include <cstdlib>
4  #include <ctime>
5  using namespace std;
6
7  int main()
8  {
9      fstream plik;
10     srand(time(0));
11     int ile=rand()%901+100;
12     //ios::out|ios::trunc - zawartosc pliku zostanie usunieta przed zapisem
13     //ios::out|ios::app - dane zostana dopisane na koncu istniejacego pliku
14     plik.open("liczby.txt",ios::out|ios::trunc);
15     if(plik.good()==true) {
16         for(int i=0; i<ile; i++)
17             plik<<(int)rand()%101<<" ";
18         plik.close();
19     }
20     cout<<"Utworzono plik \"liczby.txt\"."<<endl;
21     return 0;
22 }
```

Utworzono plik "liczby.txt".

```

1  53 1 77 39 97 66 30 78 60 51 90 92 32 33 99 67 54 16 98 75 29 74 28 7 41 44 87 84 56 15 74 89 98 58 54 5 52 39 47 2
20 40 22 82 99 18 13 70 40 89 18 67 89 37 31 44 72 20 37 16 1 26 36 1 45 73 26 36 54 16 23 69 53 4 14 3 33 100 36 81
87 3 39 79 75 44 80 95 13 70 29 53 36 58 72 15 70 77 96 39 23 28 7 19 51 55 75 22 16 89 31 30 68 35 69 59 65 35 38
33 37 23 82 74 40 0 57 23 62 25 5 70 5 52 37 92 32 98 21 78 19 79 54 49 30 23 69 13 14 37 18 21 84 93 69 40 79 15 72
9 55 14 65 31 68 81 99 54 49 50 41 60 98 31 64 96 67 2 42 86 16 40 22 62 40 14 93 58 95 97 98 61 55 49 42 50 29 85
67 2 79 11 5 52 85 45 54 44 84 18 47 100 0 78 59 57 8 94 80 27 55 14 97 80 70 16 30 60 17 19 82 61 49 63 98 2 49 76
68 78 33 0 54 26 65 87 24 68 94 86 70 28 34 11 90 24 10 28 60 12 23 62 42 97 40 51 13 33 100 39 49 43 96 0 31 72 12
9 8 82 13 45 28 88 72 65 92 79 13 91 7 70 36 59 41 46 52 63 30 21 9 84 65 96 98 97 87 79 11 32 4 99 89 21 89 13 33 8
48 86 94 8 88 69 65 95 59 75 40 54 41 3 3 33 14 19 66 45 26 39 71 58 52 20 56 29 0 61 43 83 68 60 48 81 69 84 71 21
26 35 6 1 61 10 28 42 2 0 57 87 99 23 20 23 15 77 96 68 92 64 28 71 50 12 77 31 47 38 66 86 33 75 69 67 20 38 68 25
64 95 57 73 47 50 98 24 85 10 16 4 5 63 19 19 85 87 25 9 59 91 30 20 35 74 58 93 31 43 87 45 13 70 84 36 14 59 92 60
18 64 61 99 9 19 25 74 52 79 68 87 25 34 92 91 60 97 7 31 5 62 30 33 100 53 99 4 41 56 59 77 34 65 41 65 87 89 83 17
96 0 66 63 5 12 2 36 72 77 54 29 49 68 92 89 88 100 32 58 51 65 4 34 98 85 85 15 86 73 62 46 33 92 97 15 43 90 52 56
25 92 47 48 26 82 20 51 20 62 65 62 79 61 31
```

2. Z pliku *liczby.txt* wczytaj wszystkie liczby (nie wiemy ile jest liczb), oblicz ich ilość i sumę. Wynik wypisz na ekranie.

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  int main()
6  {
7      int liczba, ile=0, suma=0;
8      fstream plik;
9      plik.open("liczby.txt", ios::in);
10     if(plik.good()==true) {
11         while(!plik.eof()) {
12             plik>>liczba;
13             suma+=liczba;
14             ile++;
15         }
16     }
17     plik.close();
18     cout<<"W pliku jest "<<ile<<" liczb."<<endl;
19     cout<<"Suma liczb z pliku wynosi "<<suma<<". "<<endl;
20     return 0;
21 }
```

```
W pliku jest 534 liczb.
Suma liczb z pliku wynosi 26964.
```

3. W pliku *macierz.txt* znajduje się macierz o rozmiarze 2x10. Wczytaj elementy macierzy z pliku do tablicy dwuwymiarowej: `int m[2][10]`. Następnie wypisz elementy macierzy na ekranie.

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  int main()
6  {
7      int m[2][10];
8      fstream plik;
9      plik.open("macierz.txt", ios::in);
10     if(plik.good() == true) {
11         for(int i=0; i<2; i++)
12             for(int j=0; j<10; j++)
13                 plik>>m[i][j];
14     }
15     plik.close();
16     cout<<"Macierz odczytana z pliku \"macierz.txt\":\n";
17     for(int i=0; i<2; i++) {
18         for(int j=0; j<10; j++)
19             cout<<m[i][j]<<" ";
20         cout<<endl;
21     }
22     return 0;
23 }
```

```
Macierz odczytana z pliku "macierz.txt":
1 3 5 7 9 0 2 4 6 8
0 9 8 7 6 5 4 3 2 1
```


4. Dany jest plik o nazwie *odszyfrujkodcezara.txt* z zakodowanym tekstem przy użyciu szyfru cezara z przesunięciem 3. Odszyfrowany tekst zapisz w pliku *odkodowanytekst.txt*. Odkoduj tylko znaki a-z i A-Z, pozostałe znaki pozostaw bez zmian.

Tekst do odkodowania (plik odszyfrujkodcezara.txt):

Lq fubswrjudskb, d Fdhvdu flskhu, dovr nqrzq dv Fdhvdu'v flskhu, wkh vkliv flskhu, Fdhvdu'v frgh ru Fdhvdu vkliv, lv rqh ri wkh vlpsohvw dqg prvw zlghob nqrzq hqfubswlrq whfkqltxhv. Lw lv d wbsb ri vxevwlwxwlrq flskhu lq zkfk hdkf ohwwhu lq wkh sodlqwhaw lv uhsodfhg eb d ohwwhu vrph ilahg qxpehu ri srlwlrqv grzq wkhdoskdehw. Iru hadpsoh, zlwk d ohiw vkliv ri 3, G zrxog eh uhsodfhg eb D, H zrxog ehfrph E, dqg vr rq. Wkh phwkrg lv qdphg diwhu Mxolxv Fdhvdu, zkr xvhg lw lq klv sulydwh fruuhvsrqghqfh.

```

1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  char szyfrCezaraKoduj(int p, char z) {
6      if ((z>=65 && z<=90) || (z>=97 && z<=122)) z=int(z)+p%26;
7      else return z;
8      if (!(z>=65 && z<=90) || (z>=97 && z<=122)) z=int(z)-26;
9      return z;
10 }
11 char szyfrCezaraDekoduj(int p, char z) {
12     if ((z>=65 && z<=90) || (z>=97 && z<=122)) z=int(z)-p%26;
13     else return z;
14     if (!(z>=65 && z<=90) || (z>=97 && z<=122)) z=int(z)+26;
15     return z;
16 }
17
18 int main()
19 {
20     char znak;
21     fstream plikWe, plikWy;
22     plikWe.open("odkodujszyfrcezara.txt",ios::in);
23     plikWy.open("odkodowanytekst.txt",ios::out|ios::trunc);
24     if(plikWe.good()==true && plikWy.good()==true) {
25         while(plikWe.get(znak)) {
26             plikWy<<szyfrCezaraDekoduj(3,znak);
27         }
28     }
29     plikWe.close();
30     plikWy.close();
31     cout<<"Odkodowano tekst, zobacz plik \"odkodowanytekst.txt\".";
32     return 0;
33 }

```

Odkodowano tekst, zobacz plik "odkodowanytekst.txt".

1 In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on. The method is named after Julius Caesar, who used it in his private correspondence.

- [illegible]