



WROWADZENIE

Wyprowadzanie danych: `cout` jest **obiekt**em służącym do wypisywania na ekranie znaków i zmiennych. Operator `<<` oznacza wprowadzenie danych do instrukcji `cout`, która następnie wyświetli je na ekranie.

Na przykład: `cout << "Pole prostokąta wynosi = " << Pole;`

Wprowadzanie danych: `cin` jest **obiekt**em reprezentującym standardowe **wejście** programu. Operator `>>` oznacza wprowadzenie danych do odpowiedniej zmiennej, podanej po jego prawej stronie.

Na przykład: `cin >> a >> b;`

Deklarowanie zmiennych:

`int a, b;` deklaracja zmiennych `a` i `b` typu całkowitego (w przypadku zmiennych tego samego typu można je wymienić po przecinku),

`float wzrost, wynik;` deklaracja zmiennych `wzrost` i `wynik` typu rzeczywistego.

Instrukcja iteracyjna `for`:

`for (wyrażenie_początkowe; warunek; wyrażenie_pętli) instrukcja;`

Zastosowanie instrukcji `for` do wprowadzenia n liczb z klawiatury:

```
for(i=0; i<n; i++)
{
    cout << "Podaj liczbę: ";
    cin >> a;
}
```

Instrukcja warunkowa `if`:

<code>if(wyrażenie)</code>	lub	<code>if(wyrażenie)</code>
<code>instrukcja1;</code>		<code>instrukcja1;</code>
<code>else instrukcja2;</code>		

Zastosowanie instrukcji `if` do sprawdzania, czy podana liczba jest dodatnia, czy niedodatnia:

```
if(a>0)
    cout << "liczba dodatnia";
else cout << "liczba niedodatnia";
```

Funkcje:

W języku C++ wszystkie podprogramy nazywane są **funkcjami**. Dzielimy je na dwie grupy:

- funkcje niezwracające wartości (odpowiednik procedury w języku Pascal):

```
void nazwa_funkcji(lista_parametrów_formalnych)
{
    lista_instrukcji;
}
```

- funkcje zwracające wartość (odpowiednik funkcji w języku Pascal):

```
opis_typu nazwa_funkcji(lista_parametrów_formalnych)
{
    lista_instrukcji;
    return wartość;
}
```

Tablice:

Aby utworzyć zmienne indeksowane w języku C++, musimy zadeklarować specjalną strukturę danych – tablicę.

`opis_typu nazwa_tablicy [liczba_elementów_tablicy];`

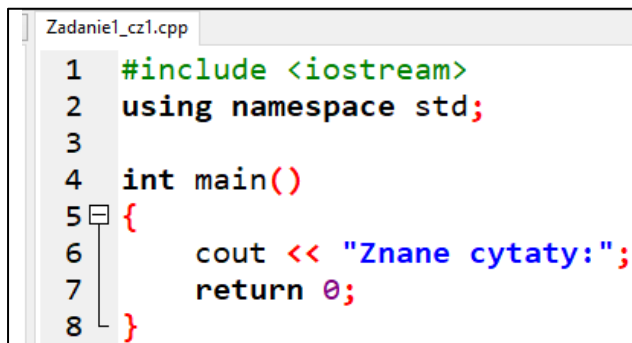
Do elementów tablicy odwołujemy się, podając nazwę tablicy i indeks elementu umieszczony w nawiasach kwadratowych, np.: `a[0]`, `a[1]`, ..., `a[n - 1]` – dla tablicy n -elementowej o nazwie `a`. W języku C++ pierwszy indeks jest zawsze równy 0. Liczbę elementów tablicy musimy określić wcześniej, np. wpisać w deklaracji tablicy jako konkretną wartość. Na przykład: `int dane[50];` oznacza zadeklarowanie tablicy o nazwie `dane` składającej się z pięćdziesięciu elementów o indeksach od 0 do 49. Do elementów tablicy odwołujemy się przez zmienne: `dane[0]`, `dane[1]`, ..., `dane[49]`.

ZADANIA

Uwaga: Utwórz na pulpicie swój folder do zapisywania rozwiązań poniższych zadań.

Zadanie 1. Wyprowadzanie napisów

1. Napisz program wyprowadzający na ekran napis: "Znane cytaty". Zapisz program w pliku pod nazwą *Zadanie1_cz1* (rys. 1.). Skompiluj i uruchom program.



```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Znane cytaty:";
7     return 0;
8 }
```

Rys. 1. Kod źródłowy programu *Zadanie1_cz1*

2. Uzupełnij program: w drugim wierszu wyświetlaj cytat "To be, or not to be ", a w trzecim imię i nazwisko autora cytatu "William Shakespeare" i po przecinku tytuł sztuki "Hamlet". Zapisz program w pliku pod nazwą *Zadanie1_cz2*. Skompiluj i uruchom program. Dodatkowo zadбай, aby cytat i tytuł utworu były wyświetlone w cudzysłowach (rys. 2.).

Wskazówki: Aby po wyświetleniu napisu kursor przeszedł do drugiego wiersza, należy umieścić sekwencję znaków "\n" po pierwszej części napisu, np.:

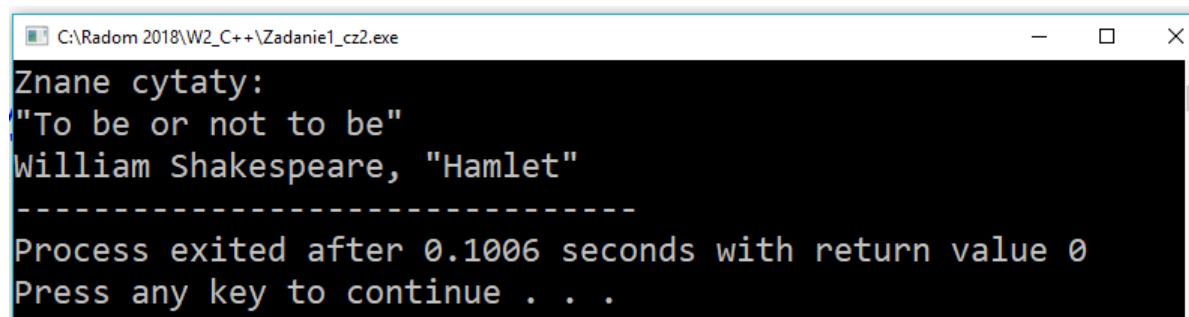
```
cout << "Znane cytaty" << endl;
```

lub

```
cout << "Znane cytaty\n";
```

Aby tekst został wyświetlony w cudzysłowach należy dodać wyświetlanie cudzysłowu: "\""

```
cout << "\"" << "To be or not to be" << "\"" << endl;
```



```
C:\Radom 2018\W2_C++\Zadanie1_cz2.exe
Znane cytaty:
'To be or not to be'
William Shakespeare, 'Hamlet'
-----
Process exited after 0.1006 seconds with return value 0
Press any key to continue . . .
```

Rys. 2. Wynik działania programu *Zadanie1_cz2*

Zadanie 2. (zad. 1., str. 89 tylko, że w tym zadaniu są obliczenia dla trzech liczb)**Deklarowanie zmiennych, wprowadzanie i wyprowadzanie danych, wykonywanie obliczeń – obliczanie sumy i średniej arytmetycznej**

1. Napisz program umożliwiający wprowadzenie dwóch liczb rzeczywistych *a* i *b*, obliczenie ich sumy (*suma*) i wyprowadzenie wyniku (*suma*) na ekran monitora. Skompiluj i uruchom program. Zapisz program w pliku pod nazwą *Zadanie2_cz1* (rys. 3.).
2. Uzupełnij program o obliczenie średniej arytmetycznej liczb *a* i *b*. Zadeklaruj nową zmienną *średnia*. Wyświetlaj też napis "Średnia wynosi: ". Napis i wynik wyświetlaj w nowym wierszu. Zapisz program w pliku pod nazwą *Zadanie2_cz2*.

```

Zadanie2_cz1.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      float a,b,suma;
7      cout << "Podaj dwie liczby" <<endl;
8      cin >> a >> b;
9      suma = a + b;
10     cout << "Suma wynosi: " << suma;
11     return 0;
12 }

```

Wskazówki: Zmienne można deklarować w dowolnym miejscu programu, ale zawsze przed pierwszym użyciem. W instrukcji przypisania używamy znaku „=”.

Rys. 3. Kod źródłowy programu Zadanie2_cz1

Zadanie 3. (zad. 4., str. 90) Stosowanie instrukcji warunkowej – algorytm sprawdzania, która liczba jest większa

Napisz program sprawdzający, która z dwóch różnych liczb całkowitych x i y wprowadzanych z klawiatury jest większa. Wyświetl na ekranie wartość większej liczby (rys. 4.). Zapisz program w pliku pod nazwą *Zadanie 3*.

```

9      if(x>y)
10         cout << "Większa jest liczba pierwsza: " << x;
11     else
12         cout << "Większa jest liczba druga: " << y;

```

Rys. 4. Stosowanie instrukcji warunkowej – fragment kodu źródłowego programu Zadanie3

Zadanie 4. Stosowanie instrukcji iteracyjnej `for` – przykład algorytmu iteracyjnego

Zmodyfikuj program z zadania 3. tak, aby sprawdzał n par liczb wprowadzanych z klawiatury. Zadeklaruj zmienną sterującą i i liczbę wprowadzanych liczb n jako `int` (rys. 5. *Ten rzut nie obejmuje deklaracji zmiennych*). Wartość n wprowadzaj z klawiatury. Dodaj przed wprowadzeniem n napis "Podaj liczbę sprawdzanych par liczb". Zapisz program w pliku pod nazwą *Zadanie4*.

```

10     for(i=0; i<n; i++)
11     {
12         cout << "Podaj dwie liczby" <<endl;
13         cin >> x >> y;
14         if(x>y)
15             cout << "Większa jest liczba pierwsza: " << x << endl;
16         else
17             cout << "Większa jest liczba druga: " << y << endl;
18     }
19     return 0;

```

Rys. 5. Stosowanie instrukcji iteracyjnej `for` – fragment kodu źródłowego programu Zadanie4

Warsztaty dla nauczycieli

Zadanie 5.* (zad. 3., str. 106) Stosowanie funkcji niezwracającej wartości, bez parametrów

1. Napisz program umożliwiający wyprowadzenie na ekran monitora napisu „Gwiazdkowe pozdrowienia”, a pod nim dwudziestu trzech gwiazdek (znaków *). Zdefiniuj funkcję *Gwiazdki* jako niezwracającą wartości, czyli typu `void` i bez parametrów. Zadaniem funkcji jest wyprowadzenie w jednym wierszu dwudziestu trzech gwiazdek. Funkcję wywołaj w programie głównym (rys. 6.). Zapisz program w pliku pod nazwą *Zadanie5_cz1*.

```

4 void Gwiazdki()
5 {
6     int i;
7     for(i=0; i<23; i++)
8     {
9         cout << "*";
10    }
11 }
12
13 int main()
14 {
15     cout << "Gwiazdkowe pozdrowienia" << endl;
16     Gwiazdki();
17     return 0;
18 }

```

Rys. 6. Definicja funkcji *Gwiazdki* i wywołanie jej w programie głównym – fragment kodu programu *Zadanie5_cz1*

2. Uzupełnij program, wyświetlając w kolejnym wierszu napis „z konferencji w Radomiu”, pod którym również wyświetlaj 23 gwiazdki. Zapisz program w pliku pod nazwą *Zadanie5_cz2*.

Zadanie 6.* (zad. 5., str. 106) Stosowanie funkcji zwracającej wartość, z jednym parametrem

Napisz program umożliwiający obliczenie objętości sześcianu. Zdefiniuj funkcję *Szescian* z jednym parametrem typu całkowitego o nazwie *liczba*, obliczającą sześcian liczby *liczba* i zwracającą do programu głównego wynik obliczenia. Wywołaj funkcję w programie głównym z parametrem aktualnym *bok*, który zadeklaruj jako `int`, a wartość wprowadzaj z klawiatury (rys. 7.). Zapisz program w pliku pod nazwą *Zadanie6*.

Uwaga: Rozwiązanie pokazane na rysunku 7. można zmodyfikować, rezygnując ze zmiennej *objetosc_szescianu*. Spróbuj poprawić program.

```

4 int Szescian(int liczba)
5 {
6     return liczba*liczba*liczba;
7 }
8
9 int main()
10 {
11     int bok, objetosc_szescianu;
12     cout << "Podaj dlugosc boku " << endl;
13     cin >> bok;
14     objetosc_szescianu = Szescian(bok);
15     cout << "Objetosc szescianu wynosi: " << objetosc_szescianu;
16     return 0;
17 }

```

Rys. 7. Definicja funkcji *Szescian* i wywołanie jej w programie głównym – fragment kodu programu *Zadanie6*

Warsztaty dla nauczycieli

Zadanie 7.* (ćw. 6. i 7., str. 98-99) Stosowanie tablicy do wprowadzania i wyprowadzania danych

1. Napisz funkcję o nazwie *WprowadzDane* wprowadzającą dane do tablicy *a*, składającej się z dziesięciu liczb całkowitych. Zapisz program w pliku pod nazwą *Zadanie7*. Skompiluj i uruchom program.
2. Zdefiniuj funkcję o nazwie *WyprowadzDane* wyprowadzającą w kolejnych wierszach dane z tablicy na ekran. W funkcji głównej *main()* po poleceniu wywołania funkcji *WprowadzDane* dodaj wywołanie funkcji *WyprowadzDane*. Zapisz program w pliku pod tą samą nazwą. Skompiluj i uruchom program.

```

4  const int N = 10;
5  int a[N];
6
7  void WprowadzDane()
8  {
9      for( int i = 0; i < N; i++)
10     {
11         cout << "Podaj dana nr " << i << ": ";
12         cin >> a[i];
13     }
14 }
15
16 void WyprowadzDane()
17 {
18     for(int i = 0; i < N; i++)
19         cout << "a[" << i << "] = " << a[i] << endl;
20 }
21
22 main ()
23 {
24     WprowadzDane();
25     WyprowadzDane();
26     return 0;
27 }

```

Rys. 8. Stosowanie tablicy do wprowadzania i wyprowadzania danych – fragment kodu programu *Zadanie7*

3. Zmodyfikuj program *Zadanie7* tak, aby dane wyświetlały się w odwrotnym porządku. Zapisz program pod nazwą *Zadanie7_odwrotnie*.

Wskazówka: Operator inkrementacji zapisujemy *i++*, a dekrementacji jako *i--*. Należy odpowiednio zmienić wyrażenia w instrukcji *for*.

Środowisko programowania dostępne bezpłatnie w Internecie:

Kompilator C++ <https://sourceforge.net/projects/orwelldevcpp/>

Podręczniki zawierające tematy z programowania w języku C++:

- G. Koba, *Teraz bajty. Informatyka dla szkoły podstawowej. Klasa VIII*,
- G. Koba, *Teraz bajty. Informatyka dla szkół ponadpodstawowych. Klasa I. Zakres podstawowy*,
- G. Koba, *Z nowym bitem. Informatyka dla szkół ponadgimnazjalnych – zakres podstawowy*,
- G. Koba, *Informatyka dla szkół ponadgimnazjalnych – zakres podstawowy*,
- G. Koba, *Informatyka dla szkół ponadgimnazjalnych – zakres rozszerzony*.

Poradniki metodyczne:

- G. Koba, *Poradnik metodyczny. Informatyka dla szkół ponadgimnazjalnych – zakres podstawowy*,
- G. Koba, *Poradnik metodyczny. Informatyka dla szkół ponadgimnazjalnych – zakres rozszerzony*.

Materiały metodyczne do pozostałych podręczników dostępne poprzez wyszukiwarkę:

<https://www.migra.pl/wyszukiwarka>