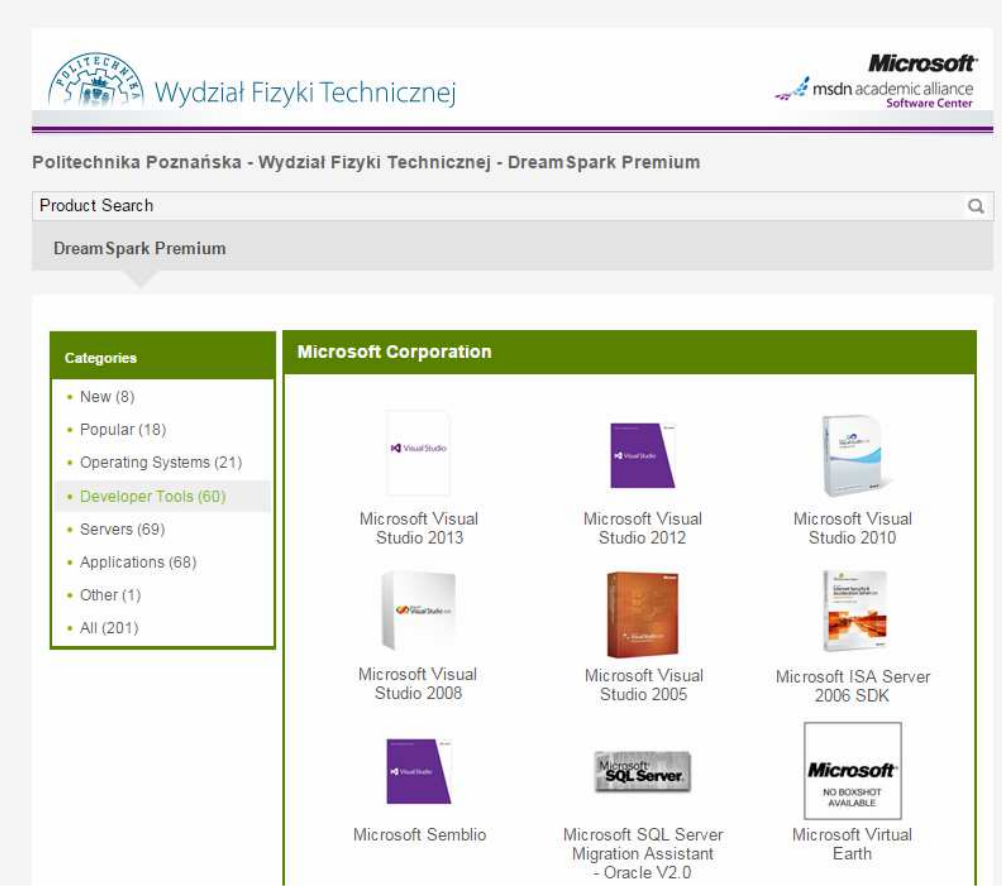
Podstawy informatyki

Prowadzący cz. 2 (C++) wykładu:
dr inż. Sylwester Przybył
Instytut Fizyki WFT
tel. (0-61) 665-3246;
e-mail : sylwester.przybyl@put.poznan.pl
Konsultacje: środa, 11:45 – 13:15, pok. 537.

W ramach **DreamSpark Premium** możliwe są do pobrania poniższe kompilatory C++

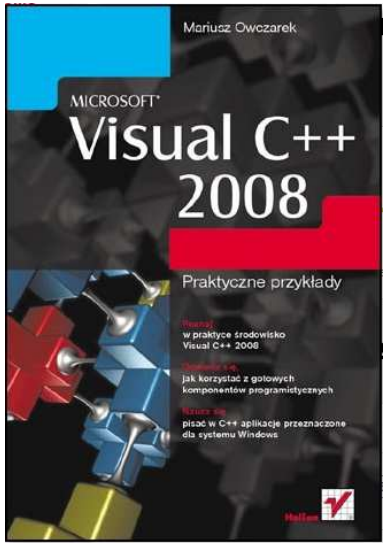
- Microsoft Visual Studio 2013,
- Microsoft Visual Studio 2012,
- Microsoft Visual Studio 2010,
- Microsoft Visual C++ 2008,
- Microsoft Visual C++ 2005,



LITERATURA

Bjarne Stroustrup (twórca języka C++) Język C++
Bjarne Stroustrup Programowanie. Teoria i praktyka z wykorzystaniem C++ (wyd.II popr.)
Jerzy Grębosz Symfonia C++ Standard (C++03)
Jerzy Grębosz Pasja C++ (niestety stare)
Siddhartha Rao C++. Dla każdego. Wydanie VII
Nicholas A. Solter, Scott J. Kleper C++ Zaawansowane programowanie
Stephen Prata Język C++. Szkoła programowania. Wydanie VI
Anthony Williams Język C++ i przetwarzanie współbieżne w akcji
D. Ryan Stephens C++ Receptury (O'Reilly)
Nicolai M. Josuttis C++. Biblioteka standardowa. Podręcznik programisty
David Vandevoorde, Nicolai M. Josuttis C++ szablony. Vademecum profesjonalisty

Mariusz Owczarek Microsoft Visual C++ 2008, Praktyczne przykłady
Radosław Sokół Microsoft Visual Studio 2012. Programowanie w C i C++



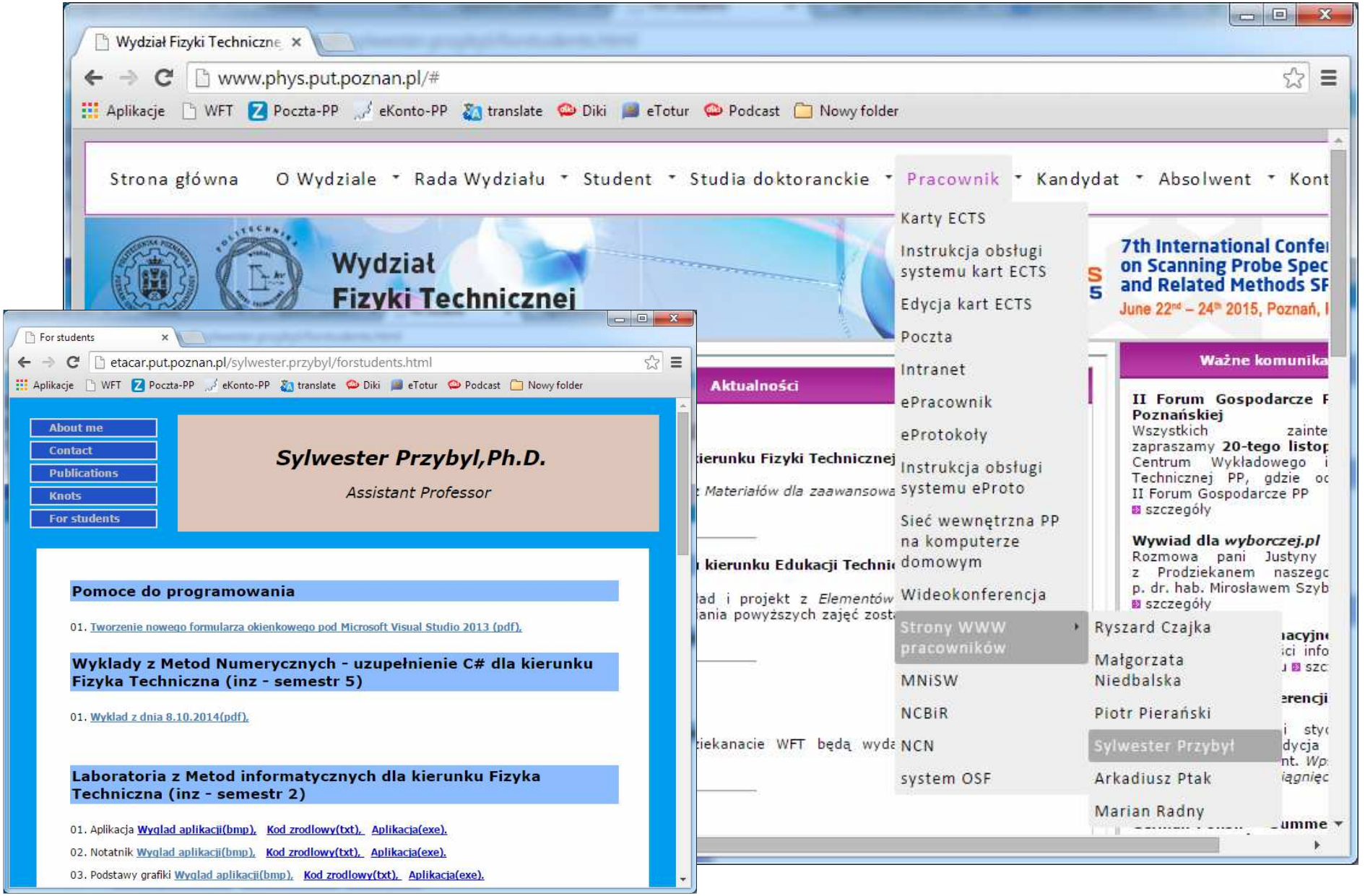
Cel dydaktyczny:

Zapoznanie studentów z podstawami języka C++ umożliwiającego tworzenie programów numerycznych rozwiązujących zagadnienia pojawiające się w fizyce i technice.

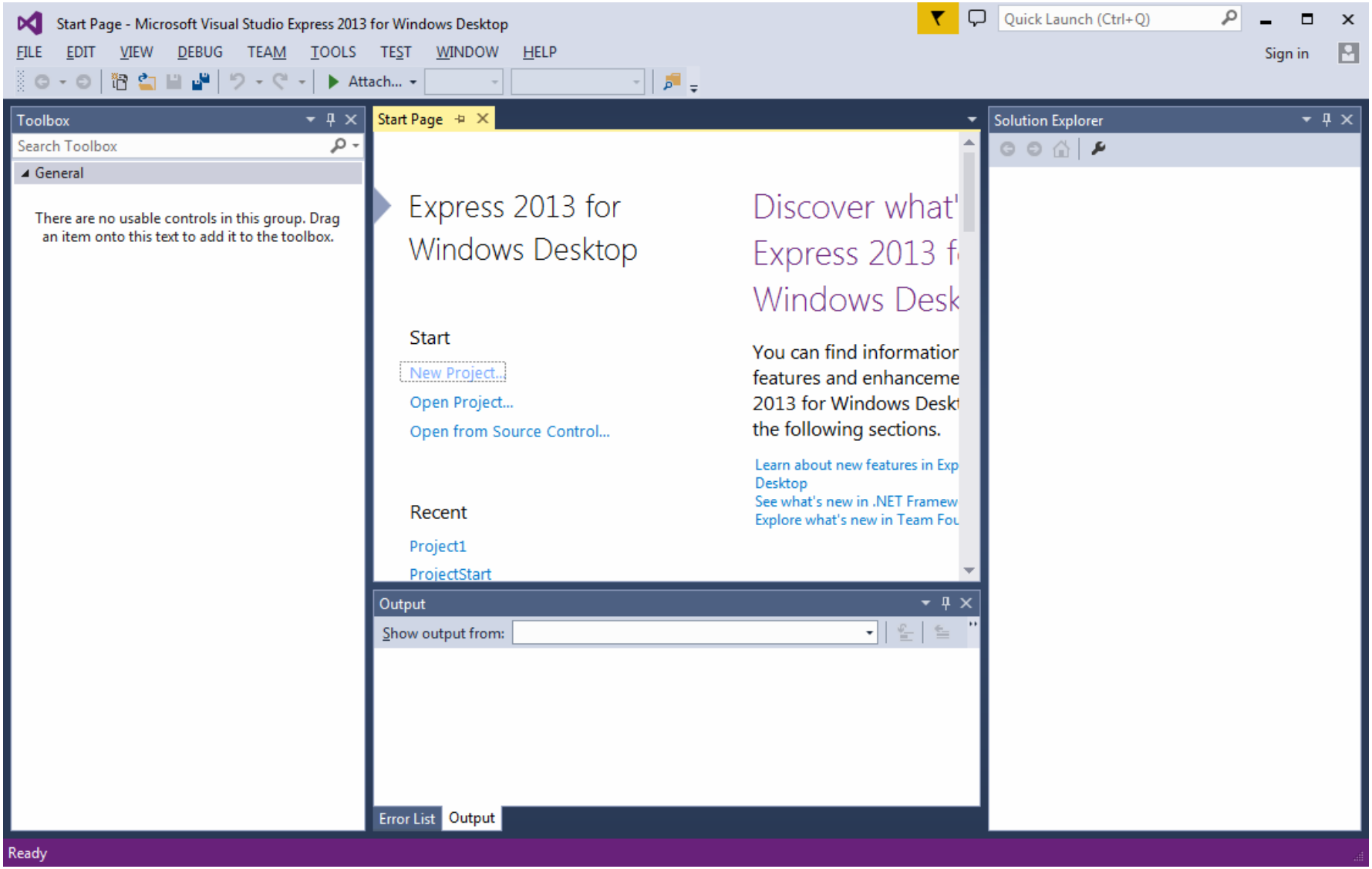
Realizowane zagadnienia:

- Programowanie w języku C++: środowisko Visual C++.
- Operacje wejścia-wyjścia. Typy i deklaracje. Wyrażenia i instrukcje.
- Operacje arytmetyczne i logiczne. Funkcje. Wskaźniki i tablice.
- Klasy i obiekty. Zasady programowania strukturalnego i obiektowego.
- Przeładowanie operatorów.
- Dziedziczenie.
- Obsługa plików tekstowych i binarnych.

Możliwe do pobrania z tego wykładu prezentacje



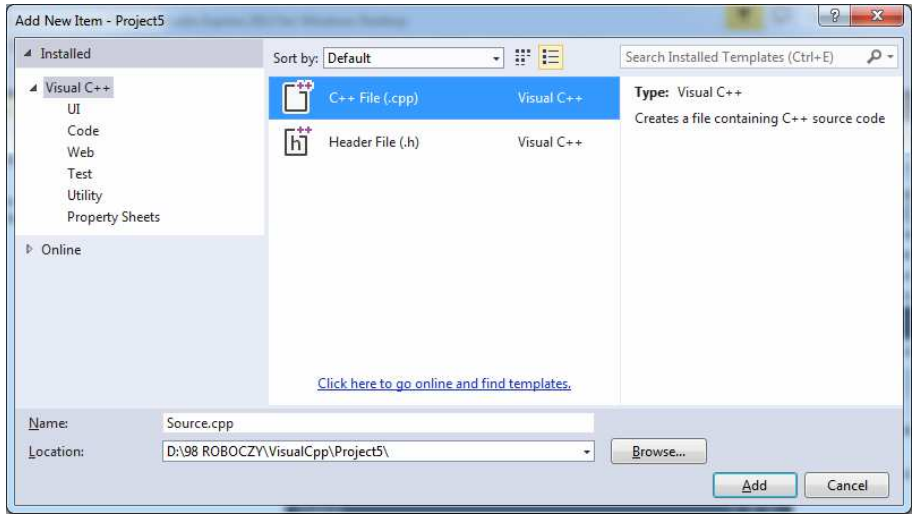
Wygląd środowiska Microsoft Visual Studio Express 2013



Tworzenie nowego projektu konsolowego

Z menu:
FILE / New Project
następnie z powstałego okienka:
Templates / Visual C++ / CLR / Empty Project

Ponownie z menu:
PROJECT / Add New Item (CTrl+Shift+A)
i znów z powstałego okienka:
Installed / Visual C++ / C++File (.cpp)
(lub) Installed / Visual C++ / Code / C++File (.cpp)

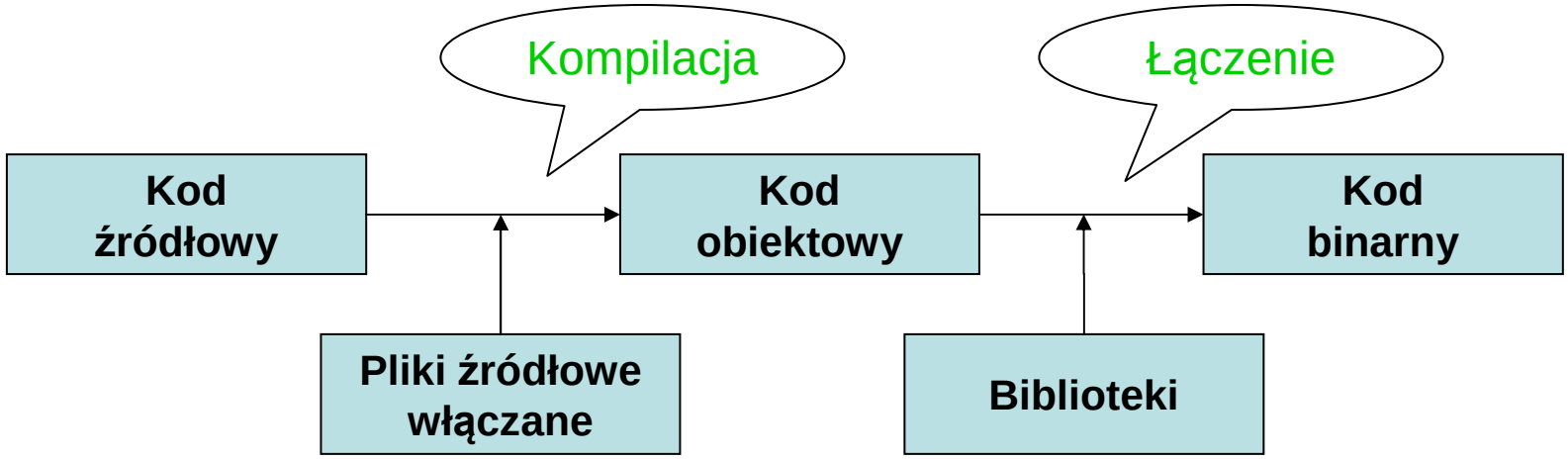


PROCES TWORZENIA PROGRAMU

```
#include <iostream>
using namespace std;
#define _USE_MATH_DEFINES
#include <math.h>
int main(void)
{
	cout << "Pierwiastek z 10 wynosi ";
	cout << sqrt(10.0);
	getchar();
}
```

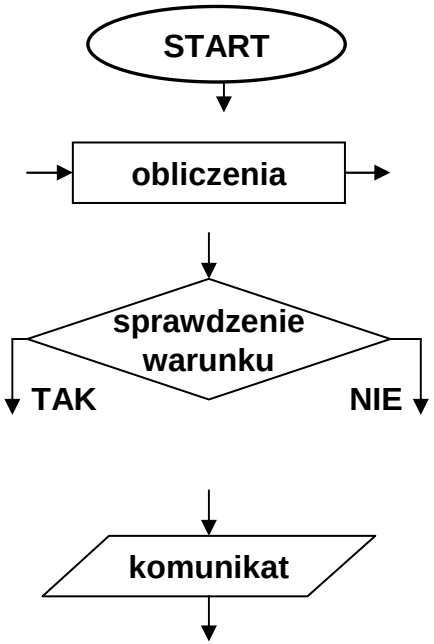
PROCES TWORZENIA PROGRAMU

edytor	→	(*.cpp)	*kod zódłowy*
kompilator	→	(*.obj)	*kod obiektowy (wynikowy)*
linker	→	(*.exe)	*kod wynikowy połączony z bibliotekami*
debugger	→	(swep/watch)	*śledzenie działania, usuwanie błędów*

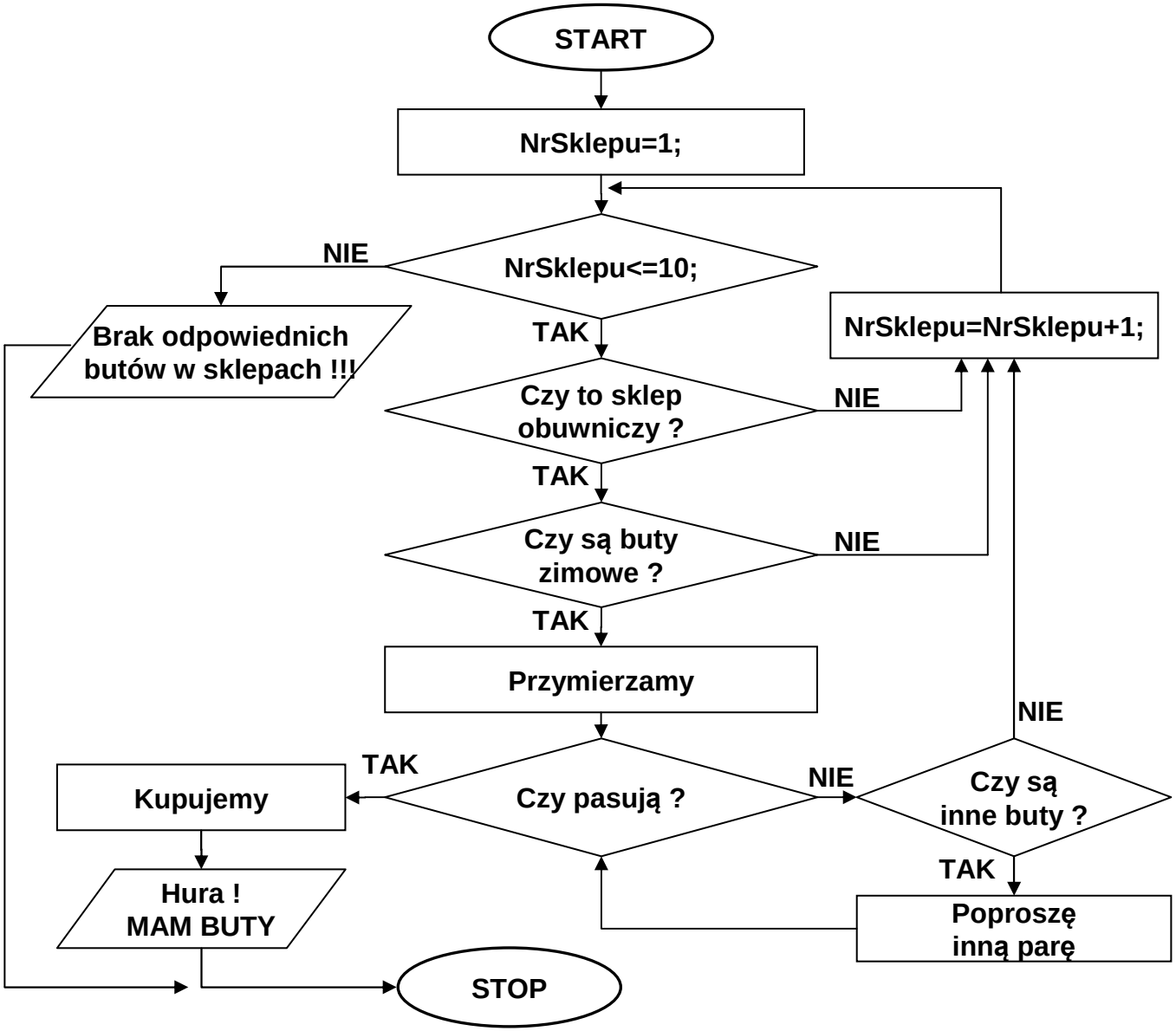


Kod źródłowy – program napisany w języku takim jak Pascal lub C++ czytelny dla programisty

Kod wynikowy - program zapisany jako ciąg rozkazów i danych w kodzie maszynowym procesora w postaci czytelnej dla komputera



Schemat blokowy zakupu butów zimowych w mieście, w którym jest 10 sklepów



Najprostszy program

```
int main(void)
{
}
```

Przechowywanie oraz zmiana danych w zmiennych.

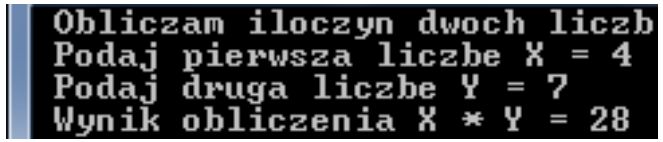
```
#include <iostream>
using namespace std;

int main(void)
{
  int x=5;
  cout<<x<<endl;
  x=12;
  cout<<x<<endl;
  x=4+3;
  cout<<x<<endl;
  x=x+4;
  cout<<x<<endl;
   getchar();
}
```

Proste obliczenia - iloczyn liczb

```
#include <iostream>
using namespace std;

int main(void)
{
  int liczbaNr1, liczbaNr2 ;
  int wynik ;
  cout << " Obliczam iloczyn dwóch liczb " << endl ;
  cout << " Podaj pierwsza liczbe X = " ;
  cin >> liczbaNr1 ;
  cout << " Podaj druga liczbe Y = " ;
  cin >> liczbaNr2 ;
  wynik = liczbaNr1 * liczbaNr2 ;
  cout << " Wynik obliczenia X * Y = " << wynik << endl ;
}
```



Proceduralna i obiektowa komunikacja z użytkownikiem

```c
/* proceduralnie: C / C++ */
#include <stdio.h>

int main( void)
{
  printf(" Dzien ");
  printf(" dobry\n ");
}
```

```cpp
// obiektowo: C++
#include <iostream>

int main( void)
{
  std::cout << " Dzien " ;
  std::cout << " dobry " << std::endl ;
}
```

#include → dyrektywa dołączenia tekstu zawartego w pliku
stdio.h → (**St**andar**dI**nput**O**utput) plik definicji funkcji Wej/Wyj
iostream → (**I**nput**O**utput**Stream**) plik definicji strumieni obiektowych
main → zastrzeżona nazwa głównej funkcji programu
void → typ danej “pustej”
\n → przejście do nowego wiersza
\t → znak tabulacji
\” → znak cudzysłowu
**** → jeden znak \
endl → manipulator przejścia do nowej linii

KOMENTARZE

```
/* ------------------------------------------------------------------------------------------
  Do programu można wprowadzić opisy  - komentarze.
  Komentarze są ignorowane przez kompilator.
  To jest komentarz – pierwszy sposób.
  UWAGA: taki komentarz nie może być zagnieżdżony
  ------------------------------------------------------------------------------------------ */
#include <iostream>
using namespace std;
//------------------------------------------------------------------------------------------
int  main(void)
{
  cout<< "Plik o nazwie \"win.ini\" znajduje";          // Komentarz – drugi sposób.
  cout<< " sie w C:\\WINDOWS. "<<endl;                  // Kompilator ignoruje resztę
}                                                       // znaków do końca linijki.
//------------------------------------------------------------------------------------------
```

```
Plik o nazwie "win.ini" znajduje sie w C:\WINDOWS.
```

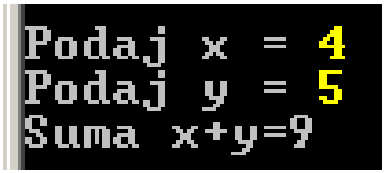
Proceduralna i obiektowa komunikacja z użytkownikiem cd.

```c
/* proceduralnie: C / C++ */
#include <stdio.h>
#include <stdlib.h>

int main( void)
{
  int x, y, s;
  system( "cls");
  printf ( "Podaj x = ");
  scanf ( "%d" , &x );
  printf ("Podaj y = ");
  scanf ("%d" , &y );
  s = x+y;
  printf("Suma x+y = %d\n", s );
}
```

```c
// obiektowo: C++
#include <iostream>
using namespace std;

int main( void)
{
  int x,y,s;
  system( "cls");
  cout << "Podaj x = ";
  cin >> x ;
  cout <<"Podaj y = ";
  cin >> y ;
  s = x+y;
  cout << "Suma x+y="<< s << '\n' ;
}
```



SŁOWA KLUCZOWE

Następujące słowa, zwane słowami kluczowymi, są zastrzeżone dla kompilatora języka C++ i **nie mogą być używane** jako nazwy zmiennych, funkcji lub etykiety:

alignas (od C++11)
alignof (od C++11)
and
and_eq
asm
auto(*)
bitand
bitor
bool
break
case
catch
char
char16_t (od C++11)
char32_t (od C++11)
class
compl
const
constexpr (od C++11)
const_cast
continue
decltype (od C++11)
default(1)
delete(1)
do
double
dynamic_cast
else
enum
explicit
export(*)
extern
false
float
for
friend
goto
if
inline
int
long
mutable
namespace
new
noexcept (od C++11)
not
not_eq
nullptr (od C++11)
operator
or
or_eq
private
protected
public
register
reinterpret_cast
return
short
signed
sizeof
static
static_assert (od C++11)
static_cast
struct
switch
template
this
thread_local (od C++11)
throw
true
try
typedef
typeid
typename
union
unsigned
using(*)
virtual
void
volatile
wchar_t
while
xor
xor_eq

DEFINIOWANIE ZMIENNYCH – USTALENIE TYPU, NAZWY

```
#include <iostream>
using namespace std;

int main( void)
{
  int i = 30;
  int j, Wiek_Ireny, Irena = i, irena =34,
          WiekIreny;                                    // to dalszy ciąg linii
  double h = 3.5, w = 4.7e6;
  float war = 3.3f;
  char symbol = 'a';
  /* ale nie
   char tekst  = „Litwo, Ojczyzno moja”;
   int Wiek Oli =35;
   double 5objetosci = 35.6;
  */
   cout << "Irena=" << Irena << endl;
   cout << "irena=" << Irena << endl;
   cout << "w=" << w << endl;
   cout << "symbol=" << symbol << endl;
}
```

PODSTAWOWE TYPY ZMIENNYCH

Nazwa typu	Zawartość	Przedział wartości	Zajęta pamięć
char	znak	- 128 ÷ 127	1 bajt
int	liczba całkowita	- 2 147 mln ÷ 2 147 mln	4 bajty
float	liczba rzeczywista	10e-38 ÷ 10e38 (7cyfr)	4 bajty
double	liczba rzeczywista	10e-308 ÷ 10e308 (15cyfr)	8 bajtów

MODYFIKATORY TYPU

signed	→ ze znakiem	int	char	-
unsigned	→ bez znaku	int	char	-
short	→ krótka (mniejsza)	int	-	-
long	→ długa (większa)	int	-	double

np.

```
unsigned long int dlugaLiczbaBezZnaku;
```

13!=6 227 020 800
20! =2 432 902 008 176 640 000

TYPY ZMIENNYCH

Typ	Wielkość w bajtach	Zakres/Precyzja
bool	1	-128 ÷ 127
char	1	całe ASCII (-128 ÷ 127)
unsigned char	1	całe ASCII (0 ÷ 255)
short int	2	-32 768 ÷ 32 767
unsigned short int	2	0 ÷ 65 535
int, long int	4	-2 147 483 648 ÷ 2 147 483 647
unsigned int (long int)	4	0 ÷ 4 294 967 295
long long	8	(-9223 ÷ 9223) 10^{15}
unsigned long long	8	(0 ÷ 18446) 10^{15}
float	4	1.2E38 ÷ 3.4E38 / 7 cyfr znacz
double	8	2.2E-308 ÷ 1.8E308/ 15 cyfr znacz.
long double	10	3.4E-4932 ÷ 1.2E4932 / 19 cyfr zn.

OPERATORY ARYTMETYCZNE

+ dodawanie
- odejmowanie
* mnożenie
/ dzielenie
% reszta z dzielenia

```
#include <iostream>
using namespace std;

int main( void)
{
  int a = 17, i;
  cout << a+3 << endl;
  i = a % 6;
  cout << i << endl;
}
```

OPERATORY PRZYPISANIA

= zwykłe przypisanie	x = 2;		
+= przypisanie sumy	x+=2;	→	x = x + 2;
-= przypisanie różnicy	x-=2;	→	x = x - 2;
*= przypisanie iloczynu	x*=2;	→	x = x * 2;
/= przypisanie ilorazu	x /=2;	→	x = x / 2;
%= przypisanie reszty	x%=2;	→	x = x % 2;

OPERATORY PRZYPISANIA - przykład

```
#include <iostream>
using namespace std;

int main( void)
{
  int x=5;
  cout << x << endl;
  x += 2;
  cout << x << endl;
  x -= 4 + 1;
  cout << x << endl;
  x *= x;
  cout << x << endl;
  x /= 5;
  cout << x << endl;
}
```

OPERATORY INKREMENTACJI I DEKREMENTACJI

zmienna++ - inkrementacja zmiennej po wyliczeniu wyrażenia
++zmienna - inkrementacja zmiennej przed wyliczeniem wyrażenia
zmienna-- - dekrementacja zmiennej po wyliczeniu wyrażenia
--**zmienna** - dekrementacja zmiennej przed wyliczeniem wyrażenia

objaśnienie:
i++; → i+=1; → i=i+1;

np.

```
 int i = 0;
 i++;
 int j = i;

 int x = 1;
 int y = ++ x ;        /* rezultat: x=2, y=2;  kolejność wykonywania ++x;  y=x; */

 int a =1;
 int b = a ++ ;        /* rezultat: a=2, b=1; kolejność wykonywania   b=a; a++; */
```

PRIORYTETY OPERATORÓW - przykład

```
#include <iostream>
using namespace std;

int main( void)
{
  int x;
  x = 2 + 3 * 10;
  cout << x << endl;
}
```

PRIORYTETY OPERATORÓW W JĘZYKU C++

Operator	Opis	Przykład
( )	wywołanie funkcji	sin()
[ ]	element tablicy	tab[10]
.	element struktury	osoba.nazwisko
–>	wskazanie elemenu struktury	wsk_osoby–>nazwisko
!	negacja logiczna	if(! (x >max)) kontynuuj;
~	negacja bitowa	~(001101) ≡ (110010)
–	zmiana znaku (negacja)	x = 10 * (– y)
++	inkrementacja (zwiększenie o 1)	x + + + y ≡ (x+ +) + y
– –	dekrementacja (zmiejszenie o 1)	– – y ≠ – – y ≡ – (– y)
&	operator referencji (adres elementu)	wsk_x = &x
*	operator dereferencji	*wsk_x = 10
(type)	zmiana typu (*typecast*)	(double) 10 ≡ 10.0
sizeof	rozmiar zmiennej lub typu (w bajtach)	sizeof(int) ≡ 2
*	mnożenie	
/	dzielenie	
%	operacja modulo (reszta z dzielenia)	if(x%2 == 0) parzyste;
+	dodawanie	
–	odejmowanie	

PRIORYTETY OPERATORÓW W JĘZYKU C++, CD

Operator	Opis	Przykład
<<	przesunięcie bitowe w lewo	1 << 2 ≡ (0001) << 2 ≡ (0100)
>>	przesuniecie bitowe w prawo	x = 4 >>1 ≡ x = 2
<	mniejszy niż	if(liczba < max) max = liczba;
<=	mniejszy lub równy	
>	wiekszy niż	
>=	wiekszy lub równy	
==	równy	
!=	nierówny (różny od)	
&	iloczyn bitowy	
^	suma bitowa modulo (różnica symetryczna)	
|	suma bitowa	
&&	iloczyn logiczny	
||	suma logiczna	
? :	wyrażenie warunkowe	
=	przypisanie	
*= /= %= +=	przypisania arytmetyczne	
–= <<= >>=		
&= ^= |=		
,	operator przecinka	

OPERATORY RELACYJNE

== równe
!= różne
< mniejsze
> większe
<= mniejsze lub równe
>= większe lub równe

```
#include <iostream>
using namespace std;

int main( void)
{
  int a;
  cout << "Podaj liczbe a = ";
  cin >> a;
  if( a < 0 )  cout << "Jest to liczba ujemna";
  else         cout << "Jest to liczba nieujemna";
}
```

```
Podaj liczbe a = -5
Jest to liczba ujemna
```

Przykłady:

wyrażenie	**wynik**
5 == 7	fałsz
3.14 >= 3.14	prawda
3.14 < 3.14	fałsz
7 != 5	prawda
0	FAŁSZ
2.7174	PRAWDA

wartość 0 (zero) jest traktowana jako FAŁSZ, dowolna inna jako PRAWDA

INSTRUKCJA WARUNKOWA IF

-usunąć łącznik else
-zostawić podwójny warunek
-zostawić pojedynczy warunek
-usunąć klamerki
-zamiana if'ów 1-2

```
#include <iostream>
using namespace std;

int main( void)
{
    double waga;
    cout << " Przekazujemy na poczcie ciężki list do wysłania .\n";
    cout << " Waga pokazuje następującą masę listu (w gramach):";
    cin >> waga;
    cout << "\n\n Odpowiedź obsługi :\n\n";
    if (waga > 100)
    {
        cout << " Waga przekracza 100 gram." << endl;
        cout << " List może być przesłany tylko jako paczka ." << endl;
    }
    else if (waga >0)
    {
        cout << " Cena znaczka na list wynosi 1,80 zł " << endl;
    }
    else
    {
        cout << " Przepraszam, zepsuła nam się waga .\n …" << endl;
    }
}
```

-**a=-2,b=3**

OPERATORY LOGICZNE

&& koniunkcja (AND)
|| alternatywa (OR)
! negacja (NOT)

przykład:

```
#include <iostream>
using namespace std;

int main( void)
{
   int a = -2, b = -2;

   if ( a > 0 && b > 0)
      cout << "\n Spelniony pierwszy warunek ";
   if ( a > 0 || b > 0)
      cout << "\nSpelniony drugi warunek ";
   if ( ! ( a == b) )
      cout << "\nSpelniony trzeci warunek ";
   if ( a != b )
      cout << "\nSpelniony czwarty warunek ";
}
```

ZAGNIEŻDŻANIE INSTRUKCJI WARUNKOWYCH IF

```
#include <iostream>
using namespace std;

int main( void)
{
    double waga, godz;
    cout << "WYSYLAMY LIST\n\n";
    cout << "Podaj godzine dostarczenia:";      cin >> godz;
    cout << "Podaj wage listu w gramach:";      cin >> waga;
    cout << endl;
    if (godz>10 && godz<18)
    {
        if (waga > 100)
        {
            cout << "  Waga przekracza 100 gram." << endl;
            cout << " List może być przesłany tylko jako paczka." << endl;
        }
        else if (waga >0)
             cout << " Cena znaczka na list wynosi 1,60 zł" << endl;
    }
    else
        cout << " POCZTA ZAMKNIETA " << endl;
}
```

CZĘSTE BŁĘDY POCZATKUJĄCYCH

```
#include <iostream>
using namespace std;

int main( void)
{
  int waga = 12;
  if( waga > 100);
  {
    cout << " Waga przekracza 100 gram." << endl;
    cout << " List może być przesłany tylko jako paczka." << endl;
  }
}
```

```
// Tak rozumie to kompilator
if( waga > 100)
{
  ;
}
{
   cout<< " Waga przekracza 100 gram." << endl;
   cout<< " List może być przesłany tylko jako paczka. " << endl;
}
```

```
// Poprawne rozwiązanie
if( waga > 100)
{
   cout << " Waga przekracza 100 gram." << endl;
   cout << " List może być przesłany tylko jako paczka." << endl;
}
```

CZĘSTE BŁĘDY POCZATKUJĄCYCH

```
#include <iostream>
using namespace std;

int main( void)
{
  int wiekOli = 12;

  if( wiekOli = 10)       cout << "  Ola ma dziesiec lat " << endl;
  if( wiekOli = 11)       cout << "  Ola ma jedenascie lat " << endl;
  if( wiekOli = 12)       cout << "  Ola ma dwanascie lat " << endl;

}
```

```
// Poprawne rozwiązanie
int wiekOli = 12;

 if( wiekOli == 10)          cout << "  Ola ma dziesiec lat " << endl;
  if( wiekOli == 11)         cout << "  Ola ma jedenascie lat " << endl;
  if( wiekOli == 12)         cout << "  Ola ma dwanascie lat " << endl;
```

CZĘSTE BŁĘDY POCZATKUJĄCYCH

```
#include <iostream>
using namespace std;

int main( void)
{
  int wiekOli = 12;
  int wiekUli = 12;
  int wiekAni = 12;

  if( wiekOli == wiekUli == wiekAni)
    cout<<"  Ola, Ula i Ania maja tyle samo lat "<<endl;
  else
    cout <<"  Ola, Ula i Ania maja rozna ilosc lat "<<endl;
}
```

```
// Tak rozumie to kompilator
 int wiekOli = 12;
  int wiekUli = 12;
  int wiekAni = 12;
  if( wiekOli == wiekUli == wiekAni)
// if( (wiekOli == wiekUli )== wiekAni)
// if( (12 == 12) == 12)
// if( true == 12)
// if( 1 == 12)
      cout<< "  Ola, Ula i Ania mają tyle samo lat " << endl;
```

```
// Poprawne rozwiązanie
if( wiekOli == wiekUli && wiekUli == wiekAni)
  cout<< "  Ola, Ula i Ania mają tyle samo lat " << endl;
```

CZĘSTE BŁĘDY POCZATKUJĄCYCH

```
#include <iostream>
using namespace std;

int main( void)
{
  double godz = 12, waga = -12;
  if( godz>10 && godz<18 )
      if( waga > 100)
      {
        cout<< " Waga przekracza 100 gram." << endl;
        cout<< " List moze byc przeslany tylko jako paczka." << endl;
       }
       else if( waga >0)
         cout<< " Cena znaczka na list wynosi 1,80 zl" << endl;
  else
    cout<< " POCZTA ZAMKNIETA " << endl;
}
```

```
// Poprawne rozwiązanie
 if( godz>10 && godz<18 )
  {
      if( waga > 100)
      {
        cout<< " Waga przekracza 100 gram." << endl;
        cout<< " List moze byc przeslany tylko jako paczka." << endl;
       }
       else if( waga >0)
         cout<< " Cena znaczka na list wynosi 1,80 zl" << endl;
  }
  else
    cout<< " POCZTA ZAMKNIETA " << endl;
}
```