

Podstawowe polecenia w języku C++

Słowa kluczowe/ instrukcje/funkcje	Opis	Uwagi
{...}	blok programu	Nawiasy klamrowe { } wyznaczają blok programu.
<code>opis_typu lista_zmiennych;</code>	deklarowanie zmiennych	Dla każdej zadeklarowanej zmiennej rezerwowany jest fragment pamięci o określonym adresie i wielkości odpowiadającej danemu typowi danych.
<code>cin >> zmienna;</code>	<code>cin</code> jest obiektem reprezentującym standardowe wejście programu	W połączeniu z obiektem <code>cin</code> operator <code>>></code> oznacza wprowadzenie danych do odpowiedniej zmiennej, podanej po prawej stronie. Jeżeli chcemy wprowadzić dane do kilku zmiennych, należy operatora <code>>></code> użyć wielokrotnie.
<code>cout << wartość;</code>	<code>cout</code> jest obiektem reprezentującym standardowe wyjście programu	W połączeniu z obiektem <code>cout</code> operator <code><<</code> oznacza wyprowadzenie wartości podanej po prawej stronie. Jeżeli chcemy wyprowadzić kilka wartości, należy operatora <code><<</code> użyć wielokrotnie.
<code>Nazwa_zmiennej = wyrażenie;</code>	instrukcja przypisania	Zmiennej podanej po lewej stronie instrukcji zostanie przypisana obliczona przez komputer wartość wyrażenia znajdującego się po prawej stronie instrukcji. Operacji przypisania można użyć wielokrotnie w ramach jednej instrukcji, stosując zapis: <code>a = b = c = 150;</code>
<code>if (wyrażenie) instrukcja1; else instrukcja2; lub if (wyrażenie) instrukcja1;</code>	instrukcja warunkowa	Wyrażenie, które występuje po słowie kluczowym <code>if</code> , może być warunkiem logicznym prostym lub złożonym. Jako <i>instrukcja1</i> i <i>instrukcja2</i> może wystąpić również blok instrukcji {...}.

<pre>switch(wyrażenie) { case etykieta1: instrukcja1; break; case etykieta2: instrukcja2; break; ... default: instrukcja; break; }</pre>	instrukcja wyboru	<p>Wyrażenie musi być typu całkowitego lub znakowego, a wszystkie etykiety muszą być stałymi: liczbami całkowitymi lub znakami. Najpierw obliczana jest wartość wyrażenia, po czym, zależnie od tego, której etykietie będzie równa, zostanie wykonana dana instrukcja. Jeśli wynikiem wyrażenia nie będzie żadna z wartości podanych w etykietach, zostanie wykonana instrukcja znajdująca się po opcjonalnym słowie default (C++). Jeśli nie ma tego słowa, program przejdzie do następnej instrukcji po switch (C++). Konieczne jest zwykle zastosowanie instrukcji break, w przeciwnym przypadku program przejdzie do wykonania instrukcji znajdującej się po następnym słowie case.</p>
<pre>for (wyrażenie_początkowe; warunek; wyrażenie_pętli) instrukcja;</pre>	instrukcja iteracyjna (pętli) for	<p>Najpierw wykonywane jest <i>wyrażenie_początkowe</i>, zwykle inicjalizujące wartość zmiennej sterującej. Następnie, dopóki spełniony jest warunek, wykonywana jest instrukcja, a potem obliczana jest wartość <i>wyrażenia_pętli</i>, służącego zazwyczaj do zwiększenia lub zmniejszenia wartości zmiennej sterującej. Jako instrukcja może wystąpić również instrukcja pętli lub blok {...}.</p>
<pre>while (warunek) instrukcja;</pre>	instrukcja iteracyjna (pętli) while	<p>Najpierw sprawdzany jest <i>warunek</i>; jeśli jest on spełniony, to wykonywana jest instrukcja. Wewnątrz bloku instrukcji powinna być zawsze umieszczona instrukcja, która zmienia wartość <i>warunku</i> – w przeciwnym wypadku pętla nigdy się nie zakończy. W szczególnej sytuacji, gdy <i>warunek</i> od razu jest niespełniony, instrukcja może w ogóle nie zostać wykonana. Instrukcja może być pojedyncza lub złożona.</p>
<pre>do instrukcja; while (warunek);</pre>	instrukcja iteracyjna (pętli) do..while	<p>Warunek jest sprawdzany po wykonaniu instrukcji, więc instrukcja (instrukcje) wewnątrz pętli zostanie wykonana przynajmniej raz, niezależnie od wartości początkowej warunku. Pętla do wykonuje się, dopóki <i>warunek</i> jest spełniony. Gdy <i>warunek</i> przestanie być spełniany, nastąpi wyjście z pętli. Instrukcja może być pojedyncza lub złożona.</p>