

Laboratorium 1 Pomoc. Podstawowe różnice pomiędzy C i C++

Cel laboratorium:

Poznanie podstawowych różnic pomiędzy językiem C i C++.

C++ zawiera: usprawnienia + specyficzne mechanizmy programowania obiektowego

	C	C++
Zależność między językami		Nadzbior C
Paradygmat programowania	Strukturalne Proceduralne Program = Dane + Algorytmy	Obiektowe (nacisk na dane) Klasa = Dane + Metody Obiekt = instancja klasy Program = operacje na obiektach Generyczne (nacisk na algorytmy)
Rozszerzenia plików	*.c *.h	*.cpp *.h, *.hpp, *.hxx
Komentarze	/* komentarz w C */	//komentarz jednoliniowy w C++
Wykorzystywane biblioteki	<pre>#include <stdio.h> #include <conio.h> #include <stdlib.h> #include <math.h> #include <string.h></pre>	Nie pisze się rozszerzeń plików nagłówkowych <pre>#include <cstdio> #include <cstdlib> #include <cmath> #include <iostream> #include <iomanip> #include <string> using namespace std;</pre>
Przestrzenie nazw (korzystanie z tych samych nazw z różnych przestrzeni)	brak	<pre>namespace p1 {char imie[] = "Bolek"; int cena=1000; } namespace p2 {char imie[]= "Lolek"; int cena=2000; } using namespace std; using namespace p1; ... cout<<"Oferta 1: "<<imie<<" "<<cena<<endl; cout<<"Oferta 2: "<<p2::imie<<" "<<p2::cena;</pre>
Deklaracje	Na początku bloku {}	W momencie przypisania pierwszej wartości <pre>for(int i=0;i<n;i++) {... }</pre>
Obsługa WE/WY	<pre>int x; scanf("%d",&x); printf("x: %d\n",x) Formaty: %d, %f, %lf, %c, %s</pre>	<pre>int x; cin>>x; //nie wczytuje spacji cout<<"x="<<x<<endl; Manipulatory formatu</pre>
Stałe typu char	Traktowane jako wartości typu int <code>char ch='A';</code>	Traktowane jako wartości typu char
Typ logiczny	0- fałsz !=0 - true	<code>bool gotowe=true;</code>
Prototypy funkcji	Korzystanie opcjonalne	Korzystanie obowiązkowe

Domyślne argumenty funkcji	brak	<pre>void f(int a, int b=2, int c=3) {...} int main(){ f(1); f(1,1); f(1,1,1); }</pre>
Szablony funkcji, klas	brak	<pre>template<clas T> T kwadrat(T x) {return x*x;} int main(){cout<<kwadrat(2);}</pre>
Określanie rozmiaru tablicy	<pre>#define ROZMIAR 100 double tab[ROZMIAR];</pre>	<pre>const int ROZMIAR=100; double tab[ROZMIAR];</pre>
Struktury i unie	<pre>struct duet{ int a; int b; }; struct duet d;</pre>	<p>Etykieta nazwą typu</p> <pre>struct duet{ int a; int b; }; duet d;</pre> <p>Struktury i unie to klasy</p>
Typ wyliczeniowy	<pre>enum poziom {wysoki, niski}; enum poziom p; p=wysoki; p++;</pre>	<p>Etykieta nazwą typu</p> <pre>enum poziom {wysoki, niski}; poziom p; p=wysoki;</pre> <p>Ograniczenia: nie można inkrementować i przypisywać wartość int bez jawnego rzutowania</p> <pre>p=(poziom) 1;</pre>
Operatory	&& !	<pre>and (&&), or (), not (!) new delete</pre>
Klasa string		<pre>string napis; cin>>napis; cout<<napis <<"ma długość "<<napis.length()<<endl;</pre>
Referencje do zmiennych (wykorzystywana w funkcjach modyfikujących argumenty)	Przekazywanie do funkcji wskaźników do zmiennych	<pre>void zamiana(int&x, int&y) {int temp; temp=x; x=y; y=temp;} int main(){ int a=10; int &b=a; int c=1; b=20;//a=20 a=5;//b=5; zamiana(a,c); cout<<a<<b<<c<<endl;}</pre>

- Manipulatory formatu:

dec - zamienia podstawę systemu liczbowego na 10 (C: %d)

hex- zamienia podstawę systemu liczbowego na 16 (C: %x)

oct - zamienia podstawę systemu liczbowego na 8 (C: %o)

showbase - włącza wyświetlanie podstawy systemu liczbowego

noshowbase - wyłącza wyświetlanie podstawy systemu liczbowego

setw(szerokość) - ustawia szerokość następnego pola

setprecision(precyzja) - ustawia precyzję (liczbę miejsc po przecinku) liczb

setfill(znak) - zmienia znak wypełniający

fixed - włącza stałą liczbę miejsc po przecinku

scientific - włącza reprezentację naukową dla liczb zmiennoprzecinkowych

left - ustawia wyrównanie do lewej

right - ustawia wyrównanie do prawej

- Przykład: `cout<<showbase<<hex<<114<<endl;`