

Pseudocod -> C#

Pseudocod	C#
Start	static void Main(string[] args)
Stop	{ }

Tipuri de date:

Pseudocod	C#
Întreg	int
Caracter	char
Real	double
Sir	string
Boolean	bool

Tablouri:

Pseudocod	C#
tip nume_tablou [dimensiunea]	tip [] nume_tablou = new tip [dimensiunea]
tip nume_tablou [dimensiunea1][dimensiunea2]	tip [] nume_tablou = new tip [dimensiunea1][dimensiunea2]

Instrucțiuni:

Instrucțiune	Pseudocod	C#
Instrucțiunea de atribuire	variabila = valoare variabila -= valoare variabila += valoare variabila *= valoare variabila /= valoare variabila %/=valoare	variabila = valoare variabila -= valoare variabila += valoare variabila *= valoare variabila /= valoare variabila %/=valoare
Instrucțiunea IF	daca (condiție) atunci { instrucțiuni; } altfel { instrucțiuni; } daca (condiție) atunci { instrucțiuni; }	if (condiție) {instrucțiuni;} else {instrucțiuni;} * { și } pot să lipsească dacă pe fiecare ramură există doar o singură instrucțiune. ** ramura else a instrucțiunii IF poate

		<i>deasemenea să lipsească</i>
Instrucțiunea FOR	pentru (contor = expresie; condiție; pas) executa { instrucțiuni; }	for (contor = expresie; condiție; pas) {instrucțiuni;} * oricare dintre cele 3 „componente” ale instrucțiunii FOR pot să lipsească, dar în paranteză trebuie să apară delimitatorul „;” ** acoladele pot sa lipsească dacă în instrucțiunea FOR se execută o singură instrucțiune
Instrucțiunea SWITCH	comuta (expresie) { cazul expresie-constanta executa: instrucțiuni; break; . . . implicit executa: instrucțiuni; break; }	switch (expresie) { case expresie- constanta : instrucțiuni break; . . . default: instrucțiuni; }
Instrucțiunea WHILE	cattimp (condiție) executa { instrucțiuni; }	while (condiție) {instrucțiuni;}
Instrucțiunea DO...WHILE	executa { instrucțiuni; } cattimp (condiție);	do {instrucțiuni} while (condiție);

Citire/scriere

Pseudocod	C#
Citește	Console.ReadLine();
Scrie	Console.WriteLine();

Operatori logici

Pseudocod	C#
NOT	!
AND	&&
OR	

Operatori matematici

Pseudocod	C#
+	+
-	-
*	*
/	/
%	%

Operatori relationali: < , > , <= ,>=, ==, !=

Pseudocod	C#
<	<
>	>
<=	<=
>=	>=
==	==
!=	!=

Operatori matematici de incrementare, decrementare: ++, --

Pseudocod	C#
++	++
--	--

Functii

Pseudocod	C#
Radical{exp}	Math.Sqrt(exp)
Absolut{exp}	Math.Abs(exp)
Cos(exp)	Math.Cos(exp)
Sin(exp)	Math.Sin(exp)
Tan(exp)	Math.Tan(exp)

Program de test:

Start

/*comentariile se scriu pe o singura linie */

/*comentariile se scriu doar in zona delimitata de Start si Stop*/

//se pot scrie comentarii in ambele moduri

//si se pot lasa randuri libere intre instructiuni

Intreg a, b=2, c, d=-24;

Real a=2.3, d, e=2, _e =-3.1;

Sir Sir1 = "Primul sir", sir2 = "Alt sir", sir3, _SsS;

Boolean gasit = Adevarat, ok;

//variabilele se pot declara doar global

//nu se pot declara variabile locale functiilor

b=(a+c)*2-d/(a-c);

Citeste a;

Daca (a<2) Atunci

Scrie a;

Altfel

{

Citeste a,b,c;

Scrie a;

}

**//!!!ACOLADELE NU SUNT OBLIGATORII CAND SE DORESTE EXECUTIA
UNEI SINGURE INSTRUCIUNI**

**//!!!ACOLADELE SE PUN SINGURE PE RAND, NU EXISTA NIMIC INAINTE SI
DUPA ACOLADE!!!!**

Pentru (a=(e*2); a<=30; a++) Executa

{

Pentru (c=3; (b<c) AND (b >= -23.3); 7) Executa

Citeste a;

b=a*5;

Scrie "A are valoarea: ",a ," iar B are valoarea: ",b;

}

a=Sin(2*3-b);

CatTimp (Adevarat) Executa

{

Scrie "Mesaj la infinit";

Citeste b;

}

Executa

Citeste a;

CatTimp ((a<1) OR (a>10));

Comuta (b)

{

Caz a Executa:

Scrie "Ai luat nota: ", a;

Break;

Caz C Executa:

Scrie "Esti la limita!";

Break;

Implicit Executa: Scrie "Ai picat examenul!";

Break;

}

Stop