**C++ forráskód készítése PLanG programokból**

Az alábbi táblázat a PLanG nyelvben használható elemek C++ nyelvre való átírását mutatja be. A bal oldalon egy-egy programelem, a jobb oldalon a C++ megfelelője van. Ahol a C++ utasítás használatához szükséges egy fejlécfájl, ott az átírt alak melletti megjegyzés tartalmazza a fejlécfájl nevét.

|  |  |
| --- | --- |
| **A program váza** | |
| A főprogram | |
| PROGRAM  *Változók*  *Utasítások*  PROGRAM-VÉGE | *Fejlécek*  **using namespace** std;  **int** main()  {  *Változók*  *Utasítások*  **return** 0;  } |
| Változódeklarációk | |
| VÁLTOZÓK:  i: EGÉSZ,  v: VALÓS,  c: KARAKTER,  s: SZÖVEG,  bf: BEFÁJL,  kf: KIFÁJL,  tomb: *Típus*[*Méret*] | **int** i=0;  **double** v=0;  **char** c=' ';  string s; //<string>  ifstream bf; //<fstream>  ofstream kf; //<fstream>  *Típus* tomb[*Méret*]; |
| Fejlécek | |
| Ha szükséges egy *fejléc*: | #include <*fejléc*> |

|  |  |
| --- | --- |
| **Vezérlő szerkezetek** | |
| Elágazás | |
| HA *FeltKif*  AKKOR  *Utasítások*  HA\_VÉGE | **if** (*FeltKif*)  {  *Utasítások*  } |
| HA *FeltKif*  AKKOR  *Utasítások-Akkor*  KÜLÖNBEN  *Utasítások-Különben*  HA\_VÉGE | **if** (*FeltKif*)  {  *Utasítások-Akkor*  }  **else**  {  *Utasítások-Különben*  } |
| Elöltesztelő cilus | |
| CIKLUS AMÍG *FeltKif*  *Utasítások*  CIKLUS\_VÉGE | **while** (*FeltKif*)  {  *Utasítások*  } | |
| Hátultesztelő cilus | |
| CIKLUS  *Utasítások*  AMÍG *FeltKif* | **do**  {  *Utasítások*  } **while** (*FeltKif*); | |
| **Általános kifejezések** | |
| Értékadás | |
| Val := *Kifejezés* | Val = *Kifejezés*; | |
| Beolvasás | |
| BE: Val1, Val2  BE bf: Val1, Val2  BE: SzovValt  BE bf: SzovValt | cin >> Val1 >> Val2;  //<iostream>  bf >> Val1 >> Val2;  getline(cin, SzovValt); //<iostream>  getline(bf, SzovValt); | |
| Kiírás | |
| KI: *Kif1*, *Kif2*  KI: SV  KI kf: *Kif1*, *Kif2* | cout << *Kif1* << *Kif2*; //<iostream>  cout << endl; //<iostream>  kf << *Kif1* << *Kif2*; | |
| Összehasonlító műveletek | |
| *Kif1* = *Kif2*  *Kif1* /= *Kif2*  *Kif1* > *Kif2*  *Kif1* <= *Kif2* | *Kif1* == *Kif2*  *Kif1* != *Kif2*  *Kif1* > *Kif2*  *Kif1* <= *Kif2* | |
| **Logikai kifejezések** | |
| Logikai értékek | |
| IGAZ  HAMIS | **true**  **false** | |
| Logikai műveletek | |
| NEM *Kif*  *Kif1* ÉS *Kif2*  *Kif1* VAGY *Kif2* | ! *Kif*  *Kif1* && *Kif2*  *Kif1* || *Kif2* | |
| **Egész kifejezések** | |
| Egész értékek | |
| 0  15  -7 | 0  15  -7 | |
| Egész műveletek | |
| - *Kif*  |*Kif*|  *Kif1* + *Kif2*  *Kif1* - *Kif2*  *Kif1* \* *Kif2*  *Kif1* DIV *Kif2*  *Kif1* MOD *Kif2*  KEREK(*Kif1*) | - *Kif*  abs(*Kif*) //<cstdlib>  *Kif1* + *Kif2*  *Kif1* - *Kif2*  *Kif1* \* *Kif2*  *Kif1* / *Kif2*  *Kif1* % *Kif2*  round(*Kif1*) //<cmath> | |
| **Valós kifejezések** | |
| Valós értékek | |
| 3.141  -6.2  7.0 | 3.141  -6.2  7.0 | |
| Valós műveletek | |
| - *Kif*  |*Kif*|  *Kif1* + *Kif2*  *Kif1* - *Kif2*  *Kif1* \* *Kif2*  *Kif1* / *Kif2*  *Kif1* ^ *Kif2*  *Kif* ^ 0.5  SIN *Kif* | - *Kif*  abs(*Kif*)         //<cmath>  *Kif1* + *Kif2*  *Kif1* - *Kif2*  *Kif1* \* *Kif2*  *Kif1* / *Kif2*  pow(*Kif1*, *Kif2*)  //<cmath>  sqrt(*Kif1*) //<cmath>  sin(*Kif*)         //<cmath> | |
| **Karakter kifejezések** | |
| Karakter értékek | |
| 'a'  ' '  SV | 'a'  ' '  '\n' | |
| Karakter műveletek | |
| KIS *Kif*  NAGY *Kif*  BETŰ *Kif*  SZÁM *Kif* | tolower(*Kif*)   //<cctype>  toupper(*Kif*)   //<cctype>  isalpha(*Kif*)   //<cctype>  isdigit(*Kif*)   //<cctype> | |
| **Szöveges kifejezések** | |
| Szöveg értékek | |
| "alma"  "" | "alma"  "" | |
| Szöveg műveletek | |
| *Szov*[*Index*]  *Szov1* + *Szov2*  |*Szov*|  *Szov*[*Kezd* : *Veg*]  *Szov1* @ *Szov2*  *Szov* @ *Kar* | *Szov*[*Index*]  *Szov1* + *Szov2*  *Szov*.length()  *Szov*.substr(*Kezd*, *Veg*-*Kezd*)  *Szov1*.find(*Szov2*)  *Szov*.find(*Kar*) | |
| **Fájl típusok műveletei** | |
| Bemeneti műveletek | |
| MEGNYIT *bf*: *Szov*  LEZÁR *bf*  VÉGE *bf* | *bf*.open(*Szov*);  *bf*.close();  !*bf*.good() | |
| Kimeneti műveletek | |
| MEGNYIT *kf*: *Szov*  LEZÁR *kf* | *kf*.open(*Szov*);  *kf*.close(); | |