////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

bool \_\_fastcall c\_CreateTiersByEdges( char\* FileName )

{ // создаёт (на основе ИГА из файла FileName первоначальную (в "верхней" форме)

// - ярусно-параллельную форму (ЯПФ) графа в массиве Tiers[][]

// - и вычисляет общее количество операторов в информационном графе алгоритма

INT iOp, fromOp, toOp, i;

char str[\_2048], w[\_256];

//

INT old\_nMoves = nMoves; // запомнили значение счётчика перемещений операторов с яруса на ярус ЯПФ

//

if( !c\_ReadEdges( FileName ) ) // читаем ИГА-файл в массив Mem\_Edges[][]

{

flagExistsEdges =

flagExistsTiers = false;

//

DisplayMessage( "E", \_\_FUNC\_\_, messNotEdges, ERR\_NOT\_MASSIVE\_EDGES ); // выдать сообщение

return ERR\_NOT\_MASSIVE\_EDGES ;

}

//

////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

//

\_maxTiers = \_maxTiersPrim; // текущие значения - без Prim

\_maxOpsOnTier = \_maxOpsOnTierPrim;

//

pTiers = (INT\*) realloc( pTiers, (\_maxTiersPrim+1) \* (\_maxOpsOnTierPrim+1) \* sizeof(INT) ); // реаллокировали память под Tiers[][]...

//

if( pTiers == NULL ) // совсем плохо - память в "куче" закончилась

{

snprintf(str,sizeof(str), "нехватка памяти для размещения массива Tiers[][] (Prim). Затребовано %d x %d = %d элементов...",

\_maxTiersPrim+1, \_maxOpsOnTierPrim+1, (\_maxTiersPrim+1) \* (\_maxOpsOnTierPrim+1));

flagExistsTiers = false ; // массив Tiers[][] не создан...

DisplayMessage( "E", \_\_FUNC\_\_, str, ERR\_NOT\_MASSIVE\_TIERS ); // выдать сообщение

MessageBeep( MB\_ICONEXCLAMATION ); // звуковое предупреждение...

MessageBox(0, str, "Предупреждение", MB\_OK | MB\_ICONWARNING | MB\_TOPMOST);

return ERR\_NOT\_MASSIVE\_TIERS ;

}

//

snprintf(str,sizeof(str), "массив Tiers[][] (Prim) перераспределён на %d x %d = %d элементов",

\_maxTiersPrim+1, \_maxOpsOnTierPrim+1, (\_maxTiersPrim+1) \* (\_maxOpsOnTierPrim+1));

DisplayMessage( "I", \_\_FUNC\_\_, str, 0 ); // выдать сообщение

//

////////////////////////////////////////////////////////////////////////////////

//==============================================================================

////////////////////////////////////////////////////////////////////////////////

//

// --- в Tiers[0][\*] помещаем операторы ВХОДНОГО (нулевого) яруса --------------

// --- это операторы, НЕ ИМЕЮЩИЕ ВХОДОВ (только выходы) ------------------------

Tiers(0,0) = 0; // число операторов ВХОДНОГО (нулевого) яруса

//

for(INT iEdges=1; iEdges<=nEdges; iEdges++) // по всем дугам

for(i=0; i<=1; i++) // по ВХОДЯЩИМ и ВЫХОДЯЩИМ операторам данной дуги

{

iOp = Edges(i,iEdges); // взяли номер оператора

if( !c\_GetCountInEdgesByOp( iOp ) && // этот оператор не имеет ВХОДЯЩИХ дуг "И"

!c\_IsOpExistOnTier( iOp, 0 ) ) // iOp НЕ ПРИСУТСТВУЕТ на ярусе iTier

{

//

////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

//

if( ( Tiers(0,0) + 3 ) >= \_maxOpsOnTier ) // то же, но с БоЛЬШИМ запасом

if( IncreaseOpsOnTier( 0, \_maxOpsOnTier \* stockMem, 1 ) == false ) // неудача перераспределения памяти

{

flagExistsTiers = false ; // массив Tiers[][] не создан...

DisplayMessage( "E", \_\_FUNC\_\_, messNotTiers, ERR\_NOT\_MASSIVE\_TIERS ); // выдать сообщение

return ERR\_NOT\_MASSIVE\_TIERS ;

}

//

////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

//

Tiers(0,0) ++ ; // увеличили на 1 число операторов на ВХОДНОМ (нулевом) уровне

Tiers(0, Tiers(0,0)) = Edges(i,iEdges); // запомнили его в столбце Tiers(0,j)

//

} // конец if

} // конец цикла по i

// конец цикла по iEdges -------------------------------------------------------

//

////////////////////////////////////////////////////////////////////////////////

// --- начинаем выбирать операторы по ярусам начиная с iTier=1 до ??? ----------

for(INT iTier=1; iTier<=\_maxTiers; iTier++) // пока не знаем, сколько будет ярусов...

{

Tiers(iTier,0) = 0; // (начальное) число операторов на ярусе iTier

// на ярусе iTier располагаем операторы, завИсимые по дугам от операторов на

// ярусе iTier-1; этих операторов ровно Tiers[iTier-1][0] штук

for(INT j=1; j<=Tiers(iTier-1,0); j++) // по from-операторам на ярусе iTier-1

{

fromOp = Tiers(iTier-1,j); // номер fromOp ("от которого" дуга)

// --- ищем в списке дуг to-оператор, по дуге соответствующий from-оператору ---

for(INT i=1; i<=nEdges; i++) // по всем дугам (связям операторов)

{

if( Edges(0,i) == fromOp ) // найдена дуга c from-оператором, соответствующим данному

{

// --- зависит ли toOp=Mem\_Edges[i] ещё от каких либо операторов, НЕ НАХОДЯЩИХСЯ

// --- НА ЯРУСАХ 1 - iTier ? Если зависит - ТАКОЙ НАМ НЕ ПОДХОДИТ !!!

toOp = Edges(1,i); // вот от этого зависит fromOp !..

// --- при каких условиях оператор toOp может быть добавлен на ярус iTier ??????

if( **c\_IsOpDependOnlyAboveTiers( toOp, iTier )** && /// если toOp зависит ТОЛЬКО

// от операторов на ВЫШЕЛЕЖАЩИХ ярусах (искл. данный iTier) "И"

**!c\_IsOpExistOnTier( toOp, iTier )** ) // Op НЕ ПРИСУТСТВУЕТ на ярусе iTier

{

//

////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

//

if( ( Tiers(iTier,0)+1 ) >= \_maxOpsOnTier ) // не помещаемся в \_maxOpsOnTier...

if( IncreaseOpsOnTier( iTier, \_maxOpsOnTier \* stockMem, 11 ) == false ) // неудача перераспределения памяти

{

flagExistsTiers = false ; // массив Tiers[][] не создан...

DisplayMessage( "E", \_\_FUNC\_\_, messNotTiers, ERR\_NOT\_MASSIVE\_TIERS ); // выдать сообщение

return ERR\_NOT\_MASSIVE\_TIERS ;

}

//

////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

//

// --- всё скопировано - можно продолжать работать уже с новым pTiers ----------

Tiers(iTier,0) ++ ; // всё хорошо, можно увеличивать на 1 число операторов на ярусе iTier...

Tiers(iTier, Tiers(iTier,0) ) = toOp; // в Tiers[iTier][\*] соответствующий toOp

} // конец if ( Is... / Is... )

} // конец if( Mem\_Edges[0][i] == fromOp )

} // конец цикла по i (списку дуг)

// -----------------------------------------------------------------------------

} // конец цикла по j (числу операторов на ярусе iTier-1)

//

////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

//

// --- анализируем, закончились ли ярусы ЯПФ -----------------------------------

if( ( Tiers(iTier,0) != 0 ) && // если на ярусе iTier есть операторы...

( (iTier+1) >= \_maxTiers ) ) // а строки массива Tiers[][] заканчиваются...

{

// --- увеличиваем \_maxTiers, а \_maxOpsInTier оставляем тем же самым -----------

// --- в этом случае можно использовать realloc, обеспечивающий сохранность данных

\_maxTiers \*= stockMem; // новое число строк с запасом stockMem

//

pTiers = (INT\*) realloc( pTiers, (\_maxTiers+1) \* (\_maxOpsOnTier+1) \* sizeof(INT) ); // захват памяти под новый массив

//

if( pTiers == NULL ) // совсем плохо - память в "куче" закончилась

{

snprintf(str,sizeof(str), "Нехватка памяти для размещения массива Tiers[][] (2). Затребовано %d x %d = %d элементов...",

\_maxTiers+1, \_maxOpsOnTier+1, (\_maxTiers+1) \* (\_maxOpsOnTier+1));

flagExistsTiers = false ; // массив Tiers[][] не создан...

DisplayMessage( "E", \_\_FUNC\_\_, str, ERR\_NOT\_MASSIVE\_TIERS ); // выдать сообщение

MessageBeep( MB\_ICONEXCLAMATION ); // звуковое предупреждение...

MessageBox(0, str, "Предупреждение", MB\_OK | MB\_ICONWARNING | MB\_TOPMOST);

return ERR\_NOT\_MASSIVE\_TIERS ;

} // конец if ( npTiers == NULL )

//

snprintf(str,sizeof(str), "массив Tiers[][] (2) перераспределён на %d x %d = %d элементов",

\_maxTiers+1, \_maxOpsOnTier+1, (\_maxTiers+1) \* (\_maxOpsOnTier+1));

DisplayMessage( "E", \_\_FUNC\_\_, str, ERR\_NOT\_MASSIVE\_TIERS ); // выдать сообщение

} // конец if ( Tiers(iTier,0) != 0 && ...

//

// --- всё скопировано - можно продолжать работать уже с новым pTiers ----------

//

////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

//

if( Tiers(iTier,0) == 0 ) // не найдено ни одного to-оператора на ярусе iTier

{

nTiers = iTier - 1 ; // запомнили число ярусов

break; // на этом ярусе нет to-операторов... ярусы кончились... выбрасываемся из цикла!

} // заполнено iTier-1

////////////////////////////////////////////////////////////////////////////////

//

} // конец цикла по iTier (ярусам ЯПФ) ========================================

//

////////////////////////////////////////////////////////////////////////////////

nOps = 0; // общее число операторов БЕЗ ВХОДНЫХ (глобальное)

// --- вычисляем cуммарное количество операторов (суммируем по ярусам ЯПФ) -----

for(INT iTier=1; iTier<=nTiers; iTier++) // по числу ярусов ЯПФ кроме ВХОДНОГО (нулевого)

nOps += Tiers(iTier,0);

//

////////////////////////////////////////////////////////////////////////////////

//

flagExistsTiers = true ; // признак того, что ЯПФ построена

//

TestExistInOutAtAllOps(); // тестирование входов и выходов

//

nOpsInput = Tiers( 0, 0 ); // число ВХОДНЫХ вершин (операторов)

nOpsOutput = Tiers( nTiers, 0 ); // число ВЫХОДНЫХ вершин (операторов)

//

t\_printf( "\n-I- %s(): ЯПФ графа в \"верхней\" форме по файлу [%s] успешно построена -I-", \_\_FUNC\_\_, FileNameEdges );

//

flagExistsTiers = true ; // массив Tiers[][] сформирован..!

//

flagCalcTLD = false ; // paramsTLD не соответствует Tiers[][]

//

if( F1->N5->Checked ) // если включён режим "приземления" всех всех выходных операторов...

OutOpsMoveLowerTier(); // все выходные операторы перенести на самый нижний ярус

//

nMoves = old\_nMoves; // восстановили значение счётчика перемещений операторов с яруса на ярус ЯПФ

//

return true ; // всё нормально - массив Tiers[][] в "верхней" форме построен

//

} // --- конец c\_CreateTiersByEdges---------------------------------------------