# Лабораторная работа 14

#### 14.1 [#30]

Соберите процедуру RecursiveSort из приведенных ниже разделов проекта. Исправьте ошибки.

**Разделы проекта RecursiveSort**

DP 1

PROCEDURE RecursiveSort(VAR F1: TEXT);

VAR

F2, F3: TEXT;

Ch: CHAR;

{PROCEDURE Split(VAR F1, F2, F3: TEXT)

Разбивает F1 на F2 и F3}

{PROCEDURE Merge(VAR F1, F2, F3: TEXT)

Сливает F2 и F3 в F1}

BEGIN {RecursiveSort}

RESET(F1);

IF NOT (EOLN(F1))

THEN

BEGIN

IF NOT (EOLN(F1))

THEN {Файл имеет как минимум 2 символа}

BEGIN

Split(F1, F2, F3);

RecursiveSort(F2);

RecursiveSort(F3);

Merge(F1, F2, F3);

END

END

END {RecursiveSort}

DP 1.1

PROCEDURE Split(VAR F1, F2, F3: TEXT);

{Разбивает F1 на F2, F3}

VAR

Ch, Switch: CHAR;

BEGIN {Split}

RESET(F1);

REWRITE(F2);

REWRITE(F3);

{Копировать F1 попеременно в F2 и F3}

WRITELN(F2);

WRITELN(F3);

END {Split}

DP 1.1.1

BEGIN {Копировать F1 попеременно в F2 и F3}

Switch := '2';

WHILE NOT (EOLN(F1))

DO

BEGIN

READ(F1, Ch);

IF (Switch = '2')

THEN

BEGIN

WRITE(F2, Ch);

Switch := '3';

END

ELSE

BEGIN

WRITE(F3, Ch);

Switch := '2';

END

END

END

DP 1.2

PROCEDURE Merge(VAR F1, F2, F3: TEXT);

{Сливает F2, F3 в F1 в сортированном порядке}

VAR

Ch2, Ch3: CHAR;

BEGIN {Merge}

RESET(F2);

RESET(F3);

REWRITE(F1);

READ(F2, Ch2);

READ(F3, Ch3);

WHILE (NOT(EOF(F2))) AND (NOT(EOF(F3))))

DO

BEGIN

IF Ch2 < CH3

THEN

BEGIN

WRITE(F1, Ch2);

READ(F2, Ch2);

END

ELSE

BEGIN

WRITE(F1, Ch3);

READ(F3, Ch3);

END

END

{Копировать остаток F2 в F1}

{Копировать остаток F3 в F1}

WRITELN(F1);

END {Merge}

DP 1.2.1

BEGIN {Копировать остаток F2 в F1}

WHILE NOT (EOF(F2))

DO

BEGIN

WRITE(F1, Ch2);

READ(F2, Ch2);

END

END

DP 1.2.2

BEGIN {Копировать остаток F3 в F1}

WHILE NOT (EOF(F3))

DO

BEGIN

WRITE(F1, Ch3);

READ(F3, Ch3);

END

END

**Конец разделов проекта RecursiveSort**