

Лабораторна робота № 8 (4 год)

Файли

Файл – це іменована область зовнішньої пам'яті, в якій розмішена послідовність даних. Файл має три характерні особливості:

- у файла є ім'я;
- компоненти файла мають однаковий тип, це може бути будь-який тип, крім файлового;
- довжина файла обмежується тільки ємністю пристрою, на якому він розміщений.

За способом організації і доступом до компонентів файли можуть бути з послідовним методом доступу і прямим методом доступу. Для доступу до деякого компонента у послідовному файлі необхідно переглянути всі компоненти, розміщені перед ним. У файлі з прямим методом доступу будь-який компонент доступний безпосередньо.

Object Pascal підтримує три види файлів: текстові, типізовані і нетипізовані. Програма працює з логічним файлом, який на етапі її виконання зв'язується з реальним фізичним файлом. Логічний файл або файлова змінна описуються у програмі одним із трьох способів:

<ім'я>: TextFile;

<ім'я>: File of <тип>;

<ім'я>: File;

де TextFile – стандартний ідентифікатор типу для текстових файлів; File, of – зарезервовані слова і <тип> – тип компонентів для типізованого файла; нетипізований файл описується зарезервованим словом File. Наприклад,

```
Type Tanketa = record
    Pib: string{25};
    Rn: word;
    Pos: string{15};
    Zp; currency;
End;
```

```
Text80 = File of string{80};  
Var F1: File of Char;  
    F2: TextFile;  
    F3: File;  
    F4: Text80;  
    F5: File of Tanketa;
```

тут F1, F4, F5 – типізовані файли; F2 – текстовий файл; F3 – нетипізований файл.

В Object Pascal немає засобів контролю виду створених файлів, тому програміст сам повинен слідкувати за відповідністю виду файла і описаною файловою змінною. Однак будь-який файл можна відкрити як нетипізований і працювати з ним.

Файл стає доступним програмі після виконання процедури його відкриття, яка зв'язує файлову змінну з фізичним файлом і відкриває його для читання або запису.

Процедури доступу до файлів:

- Procedure **AssignFile** (Var F; FileName: String); – зв'язує файлову змінну F з іменем файла FileName. FileName – символьний вираз, що містить ім'я файла або шлях до файла;
- Procedure **Reset** (Var F: File [; RecSize: Word]); – відкриває існуючий файл. RecSize має зміст тільки для нетипізованих файлів і вказує розмір блоку даних. Відкритий текстовий файл можна тільки читати, а типізований і нетипізований як читати, так і записувати. При відкритті неіснуючого файла фіксується помилка. З відкритим файлом зв'язується покажчик, який вказує на компонент з номером нуля. У процесі роботи з файлом покажчик завжди вказує на компонент, який буде наступним читатися або записуватися. Переміщення покажчика здійснюється автоматично;
- Procedure **Rewrite** (Var F: File [; RecSize: Word]); – створює новий файл. RecSize має зміст тільки для нетипізованих файлів і вказує розмір блоку даних. Відкритий текстовий файл можна тільки записувати, а типізований і нетипізований як читати, так і записувати. При відкритті

існуючого файлу всі дані, які зберігалися у файлі, знищуються і записуються нові. Повідомлень про це у програму не надходить.

- Procedure **CloseFile** (Var F); – закриває файл, а зв'язок файлової змінної з іменем файлу, встановлений раніше процедурою AssignFile, зберігається. При створенні нового або розширенні старого файлу процедура забезпечує збереження у файлі всіх нових записів і реєстрацію файлу в каталозі. Процедура CloseFile виконується автоматично стосовно до всіх відкритих файлів при нормальному завершенні програми. Оскільки зв'язок файлової змінної з іменем файлу зберігається, то файл можна повторно відкрити без додаткового використання процедури AssignFile.

Тепер розглянемо особливості текстових, типізованих і не типізованих файлів.

Текстовий файл – це послідовність символьних рядків різної довжини. В кінці рядка записується ознака кінця рядка EOLN (послідовність кодів #13 і #10). У кінці файлу записується ознака кінця файлу EOF (код #26). Текстові файли дозволяють працювати з окремими символами, символьними рядками і числами (якщо символи є послідовністю цифр).

Текстові файли допускають тільки послідовний метод доступу, тому файл, відкритий процедурою Reset, можна тільки читати, а відкритий процедурою Rewrite – тільки записувати.

Ознаки кінця рядка і кінця файлу тестуються функціями Eoln, Eof, SeekEoln, SeekEof.

Обмін даними з текстовим файлом здійснюється процедурами Read, Readln, Write, Writeln.

Для того, щоб додати записи у текстовий файл, його потрібно відкрити процедурою Append.

Типізований файл – це послідовність компонентів чітко визначеного типу (символ, число, масив, множина, запис та інше). Типізований файл – це файл прямого методу доступу, тобто доступ до компонента здійснюється за його порядковим номером

Обмін даними з типізованим файлом здійснюється

процедурами Read і Write.

Нетипізований файл – це послідовність компонентів невизначеного типу. Відсутність типу робить ці файли, з одного боку, сумісними з будь-якими іншими файлами, а з другого – дозволяє організувати високошвидкісний обмін даними між диском і пам'яттю. Нетипізовані файли є файлами з прямим методом доступу.

Обмін даними з нетипізованими файлами здійснюється процедурами BlockRead і BlockWrite.

При роботі з типізованими і нетипізованими файлами використовується процедура Seek, яка встановлює покажчик файла F на N-ий компонент (перший компонент файла має номер 0) і функції: FileSize – повертає кількість компонентів файла; FilePos — повертає біжучу позицію покажчика у файлі, тобто номер компонента, який буде оброблятися наступною операцією введення-виведення.

Типізовані і нетипізовані файли є файлами з прямим методом доступу і при їх відкритті покажчик вказує на компонент з нульовим номером. Після виконання кожної операції читання або запису покажчик буде вказувати на наступний запис. Переміщення покажчика здійснюється автоматично.

Докладно процедури і функції для роботи з файлами описані у додатку.

Приклад. Розробити програму, яка: а) створює текстовий файл TF_1 із символьних рядків різної довжини; б) читає вміст файла TF_1, формує рядки по 10 символів, вставляє перед кожним рядком п'ятизначний номер ($1 \div 99999$), відділяючи його пробілом і записує у файл TF_2; в) читає вміст файла TF_2 і друкує його по рядках.

Для розв'язку задачі командою File!New Application створимо новий проєкт. Присвоїмо формі заголовок Caption = Текстовий файл і програмне ім'я Name = FT. Командою File!Save All запишемо програмний модуль під іменем

ULABR12_1.pas, а проект – LAB12_1.dpr.

На формі розмістимо два компоненти Мемо (багаторядковий редактор). Один для введення початкових даних і другий для виведення результатів. Присвоїмо їм програмні імена Memo1, Memo2 (властивість Name) і значення властивості ScrolBars=ssBoth (горизонтальне і вертикальне прокручування). В полі значення властивості Lines, цих компонентів, двічі клацнувши мишкою, ввійдемо в редактор і очистимо значення цієї властивості.

Крім цього, розмістимо на формі дві управляючих кнопки (компонент Button) і встановимо їм такі значення властивостей, першій: Caption = Виконати, Name = Button1, а другій: Caption = Вихід, Name = Button2 (Рис. 12.1).

Тепер напишемо обробники кнопок Виконати і Вихід.

В обробнику кнопки Виконати процедурою AssignFile файлові змінні f1, f2 зв'язуються з файлами TF_1 і TF_2. Процедурою Rewrite(f1) файл TF_1 відкривається для запису. Введені у вхідний набір Memo1.Lines символні рядки переписуються у файл TF_1 і процедурою CloseFile (f1) він закривається. Потім файл TF_1 відкривається для читання, а файл TF_2 для запису. Читається вміст файла TF_1, формується рядок із номера і десяти чергових символів і записується у файл TF_2. Потім вміст файл TF_2 виводиться у вихідний набір Memo2.Lines.

В обробнику кнопки Вихід виконується метод Close.



Рис. 12.1. Форма Текстовий файл.

{Обробник кнопки Виконати}

```
procedure TFT.Button1Click(Sender: TObject);
var   f1, f2: TextFile;
      s: string;
      c: char;
      n, i, j: integer;
begin
  {Зв'язування файлових змінних з файлами на диску}
  AssignFile (f1, 'TF_1');
  AssignFile (f2, 'TF_2');
  {Відкриття файла TF_1 для запису}
  Rewrite(f1);
  {Створення файла TF_1 }
  n:=Memo1.Lines.Count;
  for i:=0 to n-1 do
  begin
```

```

        s:=Memo1.Lines[i];
        writeln(f1, s);
    end;
    {Закриття файла TF_1 }
    CloseFile (f1);
    {Відкриття файла TF_1 для читання}
    Reset (f1);
    {Відкриття файла TF_2 для запису}
    Rewrite (f2);
    {Створення файла TF_2 }
    i:=0;
    while not eof(f1) do
    begin
        i:=i+1;
        write(f2, i:5, ' ');
        j:=1;
        while (j<=10) and (not eof(f1)) do
            if eoln(f1) then readln(f1)
            else begin j:=j+1; read(f1, c); write(f2, c); end;
        writeln(f2);
    end;
    {Закриття файлів }
    CloseFile (f1);
    CloseFile (f2);
    {Відкриття файла TF_2 для читання}
    Reset (f2);
    {Виведення вмісту файла TF_2}
    while not eof(f2) do
    begin
        readln(f2, s);
        Memo2.Lines.Add(s);
    end;
    CloseFile (f2);
end;
{Обробник кнопки Вихід}
procedure TFT.Button2Click(Sender: TObject);
begin
    Close;

```

end;

Розробка проекту завершена і його можна запустити на виконання.

Завдання:

1. Розробити програму, яка: а) створює текстовий файл TF1_1 із символьних рядків різної довжини, слова в яких розділені пробілами і розділовими знаками; б) читає вміст файла TF1_1 і записує кожне слово в окремий рядок файла TF1_2 (розділові знаки опускаються); в) читає вміст файла TF1_2 і друкує його по рядках.
2. Розробити програму, яка: а) створює текстовий файл TF2_1 із символьних рядків різної довжини; б) читає вміст файла TF2_1, знаходить всі послідовності цифр і записує кожну із них в окремий рядок файла TF2_2; в) читає вміст файла TF2_2 і друкує його по рядках.
3. Задано n ключових слів. Розробити програму, яка: а) створює текстовий файл TF3_1 із символьних рядків різної довжини, слова в яких розділені пробілами і розділовими знаками; б) читає вміст файла TF3_1 і записує у файл TF3_2 тільки ті рядки, які містять хоча б одне із ключових слів (якщо таких рядків немає, то повідомлення про це поміщається у файл TF3_2); в) читає вміст файла TF3_2 і друкує його по рядках.
4. Розробити програму, яка: а) створює текстовий файл TF4_1 із символьних рядків різної довжини, слова в яких розділені пробілами і розділовими знаками (слова не перевищують довжину 16 символів); б) читає вміст файла TF4_1, визначає скільки в ньому є слів із одного, двох, трьох, і т. д. символів, результат записує у файл TF4_2: у перший рядок кількість слів із одного символу, у другий рядок кількість слів із двох символів і т. д.; в) читає вміст файла TF4_2 і друкує його по рядках.
5. Розробити програму, яка: а) створює текстовий файл TF5_1 із символьних рядків різної довжини; б) читає вміст файла TF5_1,

знаходить всі послідовності цифр, які відповідають синтаксису цілого числа, і записує їх, розділяючи пробілами, у файл TF5_2; в) читає вміст файла TF5_2 і обчислює суму цих чисел.

6. Розробити програму, яка: а) створює текстові файли TF6_1 і TF6_2, елементи в яких упорядковані за неспаданням; б) зливає файли TF6_1 і TF6_2 в один, також упорядкований за неспаданням, файл TF6_3 по 10 символів у рядку (останній рядок може бути не повним); в) читає вміст файла TF6_3 і друкує його по рядках.

7. Розробити програму, яка: а) створює текстовий файл TF7_1 із символьних рядків різної довжини, слова в яких розділені пробілами і розділовими знаками; б) читає вміст файла TF7_1, знаходить слова з подвоєнням букв і записує їх у файл TF7_2 по одному в рядок (якщо таких слів немає, то повідомлення про це поміщається у файл TF7_2); в) читає вміст файла TF7_2 і друкує його по рядках.

8. Розробити програму, яка: а) створює текстовий файл TF8_1 із символьних рядків різної довжини; б) читає вміст файла TF8_1, пропускає цифри і формує рядки по 10 символів (останній рядок може бути неповним), вставляє перед кожним порядковий номер (номер займає п'ять позицій і відділяється від тексту пробілом) і записує їх у файл TF8_2; в) читає вміст файла TF8_2 і друкує його по рядках.

9. Розробити програму, яка: а) створює текстовий файл TF9_1 із символьних рядків різної довжини; б) читає вміст файла TF9_1, формує рядки за правилом: якщо рядок файла TF1 коротший ніж 20 символів, то він доповнюється пробілами, а якщо довший, то зрізається і записує їх у файл TF9_2; в) читає вміст файла TF9_2 і друкує його по рядках.

10. Розробити програму, яка: а) створює текстовий файл TF10_1 із символьних рядків різної довжини; б) читає вміст файла TF10_1, вилучає всі цифри і записує у файл TF10_2 по 10 символів у рядок (останній рядок може бути не повним); в) читає вміст файла TF10_2 і друкує його по рядках.

11. Розробити програму, яка: а) створює текстовий файл TF11_1 із символьних рядків однакової довжини; б) читає вміст файла TF11_1, вилучає у кожному рядку всі символи крім цифр, доповнює його до заданої довжини пробілами і записує у файл

TF11_2; в) читає вміст файла TF11_2 і друкує його по рядках.

12. Розробити програму, яка: а) створює текстовий файл TF12_1 із символьних рядків різної довжини; б) читає вміст файла TF12_1 і записує його у файл TF12_2 по рядках: у першому – один символ, у другому – два символи, ... , у десятому – десять символів, у одинадцятому – один символ і т. д. (останній рядок може бути не повним); в) читає вміст файла TF12_2 і друкує його по рядках.

13. Розробити програму, яка: а) створює текстовий файл TF13_1 із символьних рядків різної довжини, слова в яких розділені пробілами і розділовими знаками; б) читає вміст файла TF13_1, знаходить слова, які розпочинаються голосною літерою і записує кожне в окремий рядок файла TF13_2; в) читає вміст файла TF13_2 і друкує його по рядках.

14. Розробити програму, яка: а) створює текстовий файл TF14_1 із символьних рядків різної довжини; б) читає вміст файла TF14_1, знаходить всі послідовності цифр, які відповідають синтаксису дійсного числа, і записує їх, розділяючи пробілами, у файл TF14_2; в) читає вміст файла TF14_2 і знаходить найбільше значення із записаних чисел.

15. Розробити програму, яка: а) створює текстовий файл TF15_1 із символьних рядків різної довжини, слова в яких розділені пробілами і розділовими знаками; б) читає вміст файла TF15_1, знаходить симетричні слова (симетричними будемо називати слова виду “абввба”) і записує їх, розділяючи пробілами, у файл TF15_2; в) читає вміст файла TF15_2 і друкує кожне слово в окремому рядку.

16. Розробити програму, яка: а) створює текстовий файл TF16_1 із символьних рядків різної довжини, слова в яких розділені пробілами і розділовими знаками; б) читає вміст файла TF16_1, знаходить слова, які розпочинаються голосною літерою, і записує кожне в окремий рядок файла TF16_2; в) читає вміст файла TF16_2 і друкує його по рядках.

17. Розробити програму, яка: а) створює текстовий файл TF17_1 із символьних рядків різної довжини; б) переписує вміст файла TF17_1 у файл TF17_2 (використовуючи при цьому допоміжний файл TF17_3) так, щоб спочатку розмішувалися всі цифри, а потім всі інші символи по 10 у рядку (останній рядок може бути

не повним); в) читає вміст файла TF17_2 і друкує його по рядках.

18. Розробити програму, яка: а) створює текстові файли TF18_1 і TF18_2 із символьних рядків різної довжини; б) переписує вміст файла TF18_1 у файл TF18_2 і вміст файла TF18_2 у файл TF18_1 (використовуючи при цьому допоміжний файл TF18_3) по 20 символів у рядку (останній рядок може бути не повним); в) читає вміст файлів TF18_1, TF18_2 і друкує його по рядках.

19. Розробити програму, яка: а) створює текстовий файл TF19_1 із символьних рядків різної довжини, слова в яких розділені пробілами (одним або декількома); б) читає вміст файла TF19_1, вилучає в кожному рядку всі слова з однієї букви та зайві пробіли і записує їх у файл TF19_2; в) читає вміст файла TF19_2 і друкує його по рядках.

20. Розробити програму, яка: а) створює текстовий файл TF20_1 із символьних рядків різної довжини, слова в яких розділені пробілами і розділовими знаками; б) читає вміст файла TF20_1, знаходить всі слова найбільшої довжини і записує їх, розділяючи пробілами, у файл TF20_2; в) читає вміст файла TF20_2 і друкує його по п'ять слів у рядок.

21. Задано n ключових слів. Розробити програму, яка: а) створює текстовий файл TF21_1 із символьних рядків різної довжини, слова в яких розділені пробілами і розділовими знаками; б) читає вміст файла TF21_1, вилучає в кожному рядку ключові слова і записує їх у файл TF21_2; в) читає вміст файла TF21_2 і друкує його по рядках.

22. Розробити програму, яка: а) створює текстовий файл TF22_1 із символьних рядків різної довжини, слова в яких розділені пробілами і розділовими знаками; б) читає вміст файла TF22_1, знаходить всі слова найбільшої довжини, які містять символ "а" і записує кожне з них в окремий рядок файла TF22_2 (якщо таких слів немає, то повідомлення про це поміщається у файл TF22_2); в) читає вміст файла TF22_2 і друкує його по рядках.

23. Розробити програму, яка: а) створює текстовий файл TF23_1 із символьних рядків різної довжини; б) читає вміст файла TF23_1, замінює символ "1" на "0" та навпаки і записує у файл TF23_2 по 15 символів у рядок (останній рядок може бути неповним); в) читає вміст файла TF23_2 і друкує його по

рядках.

24. Розробити програму, яка: а) створює текстовий файл TF24_1 із символьних рядків різної довжини; б) читає вміст файла TF24_1 і записує його у файл TF24_2 так, щоб символи у кожному рядку розміщувалися в оберненому порядку; в) читає вміст файла TF24_2 і друкує його по рядках.

25. Розробити програму, яка: а) створює текстовий файл TF25_1 із символьних рядків різної довжини; б) читає вміст файла TF25_1 і записує його у файл TF25_2 так, щоб у кожному рядку спочатку розміщувалися цифри, а потім всі останні символи із збереженням порядку їх розміщення; в) читає вміст файла TF25_2 і друкує його по рядках.

26. Розробити програму, яка: а) створює текстовий файл TF26_1 із рядків латинських букв різної довжини; б) читає вміст файла TF26_1, замінює великі літери на одноіменні малі і записує його у файл TF26_2 по 15 символів у рядок (останній рядок може бути не повним); в) читає вміст файла TF26_2 і друкує його по рядках.

27. Задано два ключових слова. Розробити програму, яка: а) створює текстовий файл TF27_1 із символьних рядків різної довжини, слова в яких розділені пробілами і розділовими знаками; б) читає вміст файла TF27_1, замінює в кожному рядку входження першого ключового слова на друге ключове слово і записує їх у файл TF27_2 (якщо ключового слова у рядку немає, то він пропускається; якщо пропускаються усі рядки, то повідомлення про це поміщається у файл TF27_2); в) читає вміст файла TF27_2 і друкує його по рядках.

28. Розробити програму, яка: а) створює текстовий файл TF28_1 із символьних рядків різної довжини, слова в яких розділені пробілами і розділовими знаками; б) читає вміст файла TF28_1, знаходить всі слова, які починаються і закінчуються однаковим символом, і записує кожне в окремий рядок файла TF28_2 (якщо таких слів немає, то повідомлення про це поміщається у файл TF28_2); в) читає вміст файла TF28_2 і друкує його по рядках.

29. Розробити програму, яка: а) створює текстовий файл TF29_1 із символьних рядків різної довжини; б) читає вміст файла TF29_1, пропускає всі крапки і записує його у файл TF29_2 по

25 символів у рядку (останній рядок може бути не повним); в) читає вміст файла TF29_2 і друкує його по рядках.

30. Задано слово. Розробити програму, яка: а) створює текстовий файл TF30_1 із символьних рядків різної довжини, слова в яких розділені пробілами і розділовими знаками; б) читає вміст файла TF30_1, вилучає в кожному рядку всі входження заданого слова і записує їх у файл TF30_2; в) читає вміст файла TF30_2 і друкує його по рядках.