

7. УВЕДЕННЯ ДАНИХ БАЗОВИХ ТИПІВ ІЗ ТЕКСТОВОГО ФАЙЛА

Для введення даних з текстових файлів використовують процедури `read` та `readln`. Спочатку розглянемо процедуру `read`. Її виклик має такий найпростіший вигляд.

```
read(f, ім'я_змінної_базового_типу);
```

Уведення символа. При виконанні виклику `read(f, X)` змінній `X` типу `char` присвоюється доступний символ тексту, яким би він не був, а доступним стає наступний за ним. Виняток — якщо доступний `#26`, то `X` отримує значення `#0` і символ `#26` залишається доступним.

Уведення чисової константи. Ціла константа — це послідовність цифр, можливо, зі знаком `+` або `-` на початку, яка задає ціле число відповідного цілого типу; між знаком та першою цифрою в тексті не може бути пропусків. Дійсна константа — це послідовність цифр та інших символів зі структурою констант мови Паскаль, наприклад `1.1, 2., 0.99, 1e-3, -2.73E+02`.

Числові константи в текстах відокремлюються пропусками в довільній кількості. Символи табуляції та кінця рядка також будемо називати пропусками.

Виклик `read(f, X)`, де `X` — ім'я цілої або дійсної змінної, виконується так. З тексту від доступного символа читається пропуски, а потім — символи константи до найближчого пропуску (можливо, до символу `#26` або до кінця файла). Доступним після читання константи буде перший пропуск після неї (відповідно, `#26` або кінець файла). Якщо символи утворюють константу потрібного типу, то за ними обчислюється значення й присвоюється змінній. При дійсній змінній `X` у тексті може знаходитися й ціла константа — за нею обчислюється дійсне значення.

Символи можуть не утворювати константу відповідного типу — тоді виникає помилкова ситуація і виконання програми аварійно

завершується. Наприклад, помилковими є послідовності символів – 2 (пропуск між знаком і цифрою), 12345m, 123– (присутні нецифрові символи там, де їх не може бути) або 13., якщо читається значення цілої змінної.

Якщо доступний кінець файла або від поточної позиції в файлі до його кінця чи найближчого символа #26 записано лише пропуски, то при спробі прочитати число відповідна змінна отримує значення 0.

В одному виклику процедури **read** можна указати кілька змінних.

```
read(f,
```

```
    список_позначень_zmіnnihchislovikh_tipiv);
```

Цей виклик виконується так само, як і відповідна послідовність викликів.

```
read(f, im'я_zmіnnoi_1);
```

```
read(f, im'я_zmіnnoi_2);
```

```
...
```

Читання констант базових типів за процедурою **readln** аналогічно процедурі **read**. Відмінність полягає в тому, що після читання константи всі символи тексту, які залишилися до найближчого кінця рядка в тексті, пропускаються разом із ним. Доступним стає перший символ наступного рядка тексту (якщо до кінця рядка зустрінеться символ #26, доступним стане він). Якщо у виклику указано кілька імен змінних, то «хвіст» рядка пропускається тільки після останньої константи. Якщо список імен порожній, то виклик **readln(f)** пропускає поточний рядок тексту.

Приклад. Нехай у тексті f записано такі символи:

1	2	3	#13
5	5	#26	

Нехай x, y, z, t – імена цілих змінних. Розглянемо приклади викликів процедур і значень, яких набудуть ці змінні.

Виклики	x	y	z	t
read(f,x,y); read(f,z,t)	1	2	3	55
readln(f,x,y); read(f,z,t)	1	2	55	0
readln(f,x); readln(f,y,z,t)	1	55	0	0

- Спосіб виконання процедур уведення дозволяє розглядати текст як послідовність даних, що задають значення різних типів, тобто як *потік різnotипних даних*. Звичайно, дані мають відповідати типам змінних, які у процесі виконання програми отримують значення з цього потоку.