

## Тема № 13.

### ВИКОРИСТАННЯ МОДУЛЯ CRT

Pascal має підпрограми розширених можливостей текстового вводу/виводу. Ці підпрограми зосереджені в бібліотеці (модулі) *CRT*, що входить у комплект постачання Borland Pascal. У модуль включені також процедури роботи зі звуковим генератором комп'ютера.

Функція *KeyPressed* повертає значення типу *boolean*, що вказує стан буфера клавіатури: *FALSE* означає, що буфер порожній, а *TRUE* — що в буфері є хоча б один символ, ще не прочитаний програмою.

Функція *ReadKey* повертає значення типу *char*. При звертанні до цієї функції аналізується буфер клавіатури: якщо в ньому є хоча б один не прочитаний символ, код цього символу береться з буфера і повертається як значення функції, в іншому випадку функція буде очікувати натискання на будь-яку клавішу. Введення символу за допомогою цієї функції не супроводжується ехо-повтором і вміст екрану не змінюється.

Процедура *TextMode (Mode: Word)*; використовується для зазначення одного з можливих текстових режимів роботи адаптеру монітора. Параметр *Mode* — код текстового режиму. Як значення цього виразу можуть використовуватися наступні константи, визначені в модулі *CRT*:

*BW40=0*; {Чорно-білий режим 40\*25}

*Co40=1*; {Кольоровий режим 40\*25}

*BW80=2*; {Чорно-білий режим 80\*25}

*Co80=3*; {Кольоровий режим 80\*25}

*Mono=7*; {Використовується з MDA}

*Font8\*8=256*; {Використовується для шрифту, що завантажується, у режимі 80\*43 або 80\*50 з адаптерами EGA або VGA}

При виборі відповідного режиму стають доступними можливості, надані підпрограмами з модуля *CRT*: установка кольору тексту, фону і т.д.

Процедура *Sound(F: Word)* змушує динамік звучати з потрібною частотою. Параметр *F* — вираз типу *Word*, що визначає частоту звуку в герцах. Після звертання до процедури включається динамік і керування негайно повертається в основну програму, динамік звучатиме надалі аж до виклику процедури *NoSound*.

Процедура *NoSound* виключає динамік. Якщо він до цього моменту не був включений, виклик процедури ігнорується.

Процедура *Delay(T: Word)* забезпечує затримку роботи програми на заданий інтервал часу. Параметр *T* — вираз типу *Word*, що визначає інтервал часу (у мілісекундах), протягом якого затримується виконання наступного оператора програми.

Для генерації звукового сигналу, як правило, використовується виклик описаних процедур за схемою *Sound-Delay-NoSound*.

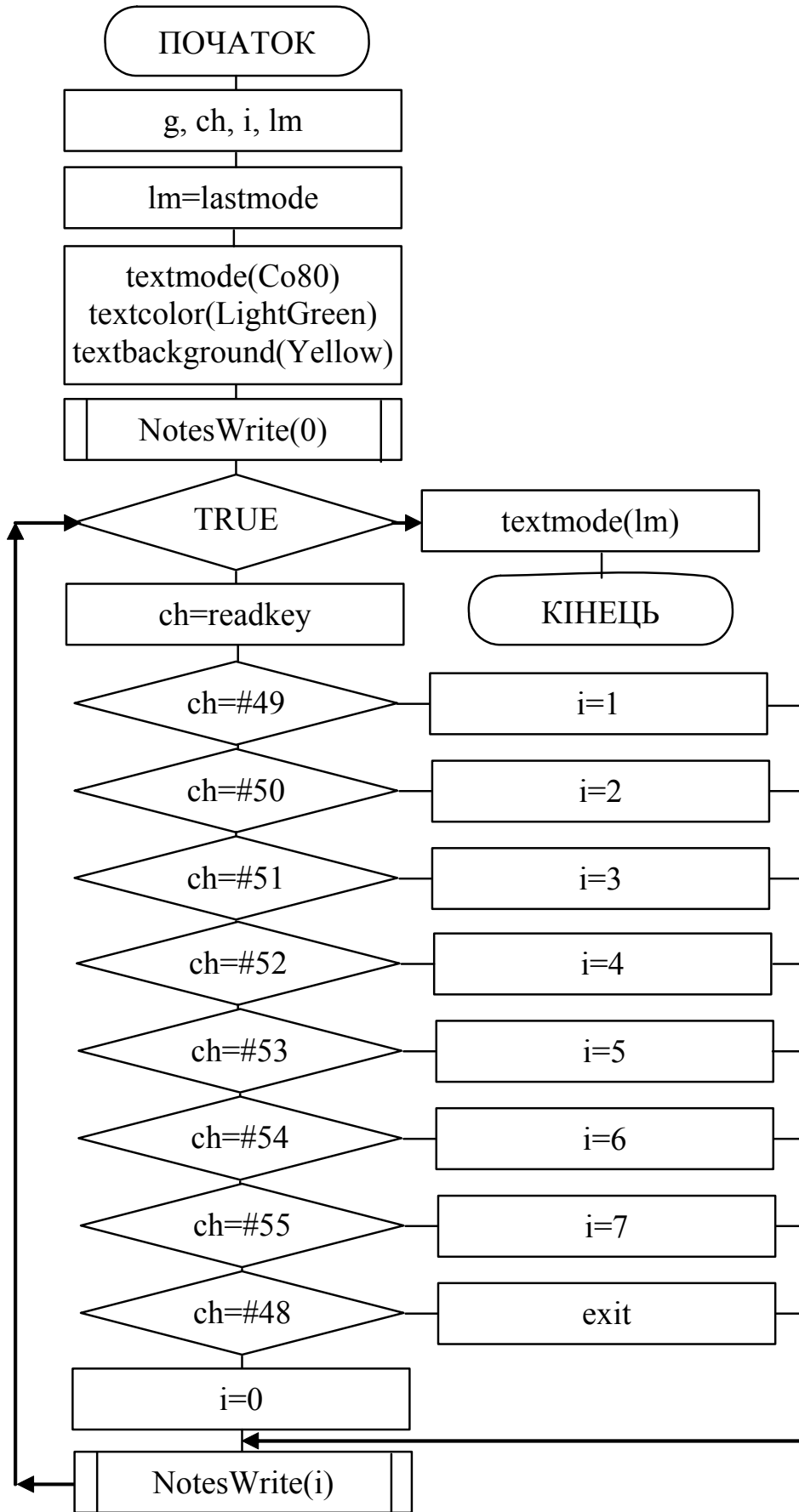
### **Завдання**

Скласти блок-схему алгоритму та написати програму використання клавіш клавіатури 1, 2, ..., 7 для програвання нот музичної гами: до, ре, ..., сі. Звучання кожної ноти супроводити виведенням її назви на екрані іншим кольором серед всього переліку нот. Вихід з програми – клавіша 0.

### **Методичні рекомендації**

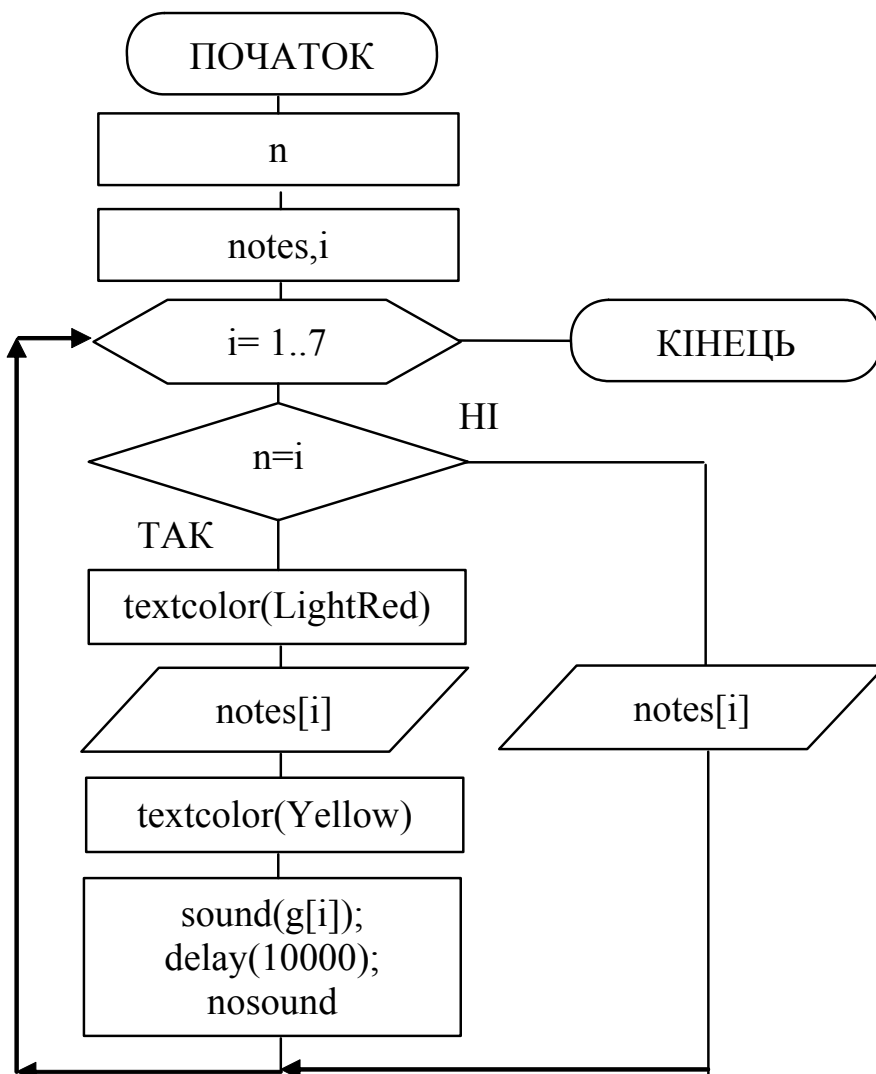
1. Проаналізувати технічну постановку задачі та скласти блок-схему.

За допомогою процедур модуля *CRT* потрібно встановити режим виводу, кольори фону та тексту, після чого зчитати символ з клавіатури і проаналізувати його. Виведення назви нот та звукового сигналу будемо здійснювати в окремій процедурі. Отже, блок-схема програми:



Блок-схема процедури *NotesWrite*:

Процедура *NotesWrite* виводу на екран переліку нот та виводу звукового сигналу відповідної ноти (параметр)



2. Записати код програми за складеною блок-схемою.

```

{*****}
* MNotes - програма програвання музичних нот      *
* Copyright (c) Шищук В.В. гр.ІС-04-1, 11/12/04  *
{*****}
PROGRAM MNotes;
USES crt;           {підключення модуля}
CONST
  g:array[1..7] of integer = (262, 294, 330, 349,
                             392, 440, 494);
                             {частоти октави}
  
```

```
VAR
  ch: char;
  i: integer;
  lm: word;

PROCEDURE NotesWrite(n:byte);
CONST
  notes:array[1..7] of string[5] = ('ДО', 'РЕ',
    'МИ', 'ФА', 'СОЛЬ', 'ЛЯ', 'СИ');
VAR
  i: byte;
BEGIN
  gotoxy(1,12);      {перехід на середній рядок}
  for i:=1 to 7 do
    begin
      if n=i then
        begin
          textcolor(LightRed);
          write(notes[i]:10);
          textcolor(Yellow);
          sound(g[i]);
          delay(10000);
          nosound;
        end
      else write(notes[i]:10);
    end;
END;

BEGIN
  lm:=lastmode;      {запам'ятовуємо режим}
  textmode(Co80);    {змінюємо режим на кольоровий}
  textcolor(LightGreen);      {колір тексту}
  textbackground(Blue);      {колір фону}
  clrscr;                {очистити екран}
  gotoxy(35,1);
  writeln('Програма генерації звуків музичної октави');
  textcolor(Yellow);
  NotesWrite(0);
  while true do
    begin
      ch:=readkey;
      case ch of
```

```
#49: i:=1;
#50: i:=2;
#51: i:=3;
#52: i:=4;
#53: i:=5;
#54: i:=6;
#55: i:=7;
#48: exit           {вихід з програми}
else i:=0;
end;
NotesWrite(i);
end;
textmode(lm);      {повернути попередній режим}
writeln
END.
```

3. Виконати компіляцію та запустити програму на виконання.
4. Протестувати програму, натискаючи різні клавіші клавіатури.
5. Зберегти написану програму в персональну папку.
6. Оформити звіт про виконану роботу.

### **Індивідуальні завдання:**

Використовуючи можливості модуля *CRT*, виконати оформлення програми роботи з типізованими файлами попередньої теми №12.

**рівень А**

Фон екрану та текст, що виводиться програмою повинні відображатись в кольорах. При вводі/виводі даних використати розміщення на екрані. Кожний запит на введення даних супроводжувати звуковим сигналом.

**рівень Б**

Фон екрану та текст, що виводиться програмою повинні відображатись в кольорах. Меню вибору режиму робо-

ти розмістити в рамочці. Кожну дію користувача супроводжувати звуковими сигналами різної тональності. Запуск та вихід з програми супроводжувати програванням музичної октави.

## рівень **B**

Фон екрану та текст, що виводиться програмою повинні відображатись в кольорах. Меню вибору режиму роботи здійснювати у виділеному вікні. Організувати виведення даних та результати обробки в виділених вікнах. Кожну дію користувача супроводжувати звуковими сигналами різної тональності. Запуск та вихід з програми супроводжувати програванням музичної октави.

### **Контрольні запитання:**

- 1) Що таке модуль і для чого його використовують?
- 2) Якого роду задачі дозволяє виконувати модуль CRT?
- 3) В чому суть програмування клавіатури?
- 4) Які можливості текстового виводу надає модуль CRT?
- 5) Наведіть приклади застосування звукового генератора у практичних задачах? Як записується код програми у таких випадках?
- 6) Як можна проаналізувати клавішу, натиснуту на клавіатурі?
- 7) Які дії можна виконувати з кольорами при застосуванні модуля CRT?
- 8) Які методи розташування на екрані надає модуль CRT?
- 9) Де в інтегрованому середовищі Borland Pascal вказується шлях до модулів?
- 10) Як зазначити необхідний режим виводу? Чим відрізняються режими один від одного?