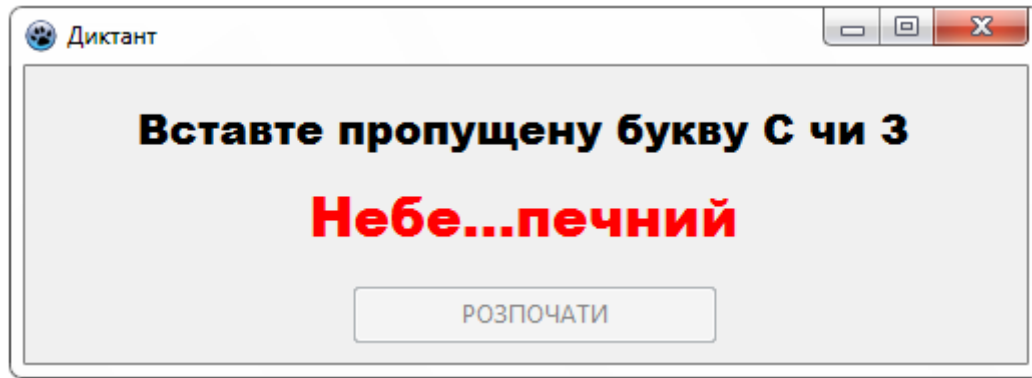


Завдання: змінити раніше створену програму «Диктант» таким чином, щоб вона перемішувала завдання, які виводяться на екран (видавала існуючі запитання щоразу в іншому порядку).



Реалізація

Відкриємо раніше створений проект з програмою «Диктант». Внесемо у нього зміни. На цей раз створимо масив запитань **Zavd**, в якому кожне запитання буде окремим елементом масиву (наприклад, **Zavd[2]** – друге запитання). При натисканні на кнопку «РОЗПОЧАТИ» будемо перемішувати завдання (для цього «загадаємо» випадкові номери завдань **N1** та **N2**, а потім поміняємо місцями запитання з цими номерами, використовуючи при обміні додаткову змінну **buff** – буфер для тимчасового зберігання даних).

■1. Змінимо розділ описів. Опишемо змінні **N**, **i**, **N1**, **N2** цілого типу (змінна **i** буде використовуватись як лічильник при перемішуванні і рахуватиме кількість обмінів), змінну **buff** рядкового типу (нагадаємо, у ній будуть тимчасово розміщуватись тексти запитань у процесі перемішування) та масив **Zavd** із 5 елементів-запитань (звісно, запитань може бути більше – стільки, скільки вам потрібно).

```
var
  Form1: TForm1;
  N, i, N1, N2: integer;
  buff: string;
  Zavd: array[1..5] of string;
```

■2. Створимо обробник події **OnCreate**: для цього клацніть де—небудь на формі і в *Інспекторі об'єктів* на вкладці *Події* викличте подвійним клацанням обробник події. В процедурі **FormCreate** використаємо оператор **randomize** для ініціалізації генератора випадкових чисел, а також назначимо значення елементам масиву **Zavd**:

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  Zavd[1] := 'Небе...печний';
  Zavd[2] := '...шити';
  Zavd[3] := '...творити';
  Zavd[4] := '...купатися';
  Zavd[5] := '...турбований';
  randomize
end;
```

■3. Внесемо суттєві зміни до обробника події клацання кнопки (процедури **Button1Click**): додамо цикл з лічильником **i**, який 10 разів (число 10 взято довільно, при бажанні його можна змінити) перемішає запитання. Щоразу при перемішуванні, як сказано вище, загадуються випадкові цілі числа **N1** та **N2** в межах від 1 до 5 (саме такі номери мають наші запитання), а потім відбувається обмін запитань місцями за класичним алгоритмом з використанням проміжної змінної **buff**. Оскільки замість однієї дії у циклі виконуються аж 5 дій, то ці дії помістимо в операторні дужки **begin...end**.

```

procedure TForm1.Button1Click(Sender: TObject);
begin
    Timer1.Enabled:=true;
    Label1.Visible:=true;
    Button1.Enabled:=false;

    For i:=1 to 10 do
    begin
        N1:=1+random(5);
        N2:=1+random(5);
        buff:=Zavd[N1];
        Zavd[N1]:=Zavd[N2];
        Zavd[N2]:=buff;
    end;
end;

```

■4. Змінимо і тіло таймера – процедуру **Timer1Timer**. На відміну від попередньої програми, тепер при значеннях змінної **N** від 1 до 5 ми лише вкажемо, що в **Label1** потрібно виводити запитання з номером **N**. Якщо ж значення **N** перевищить кількість наших запитань, то, як і в попередній програмі, виведемо напис «Робота закінчена» і на наступному «клацанні» таймера закриємо форму.

```

procedure TForm1.Timer1Timer(Sender: TObject);
begin
    N:=N+1;
    case N of
        1..5: Label2.Caption:=Zavd[N];
        6: Begin Label1.Caption:='Робота закінчена'; Label2.Caption:='' End;
        7: Form1.Close;
    end;
end;

```

Примітка 1: закривати програму після проходження всіх запитань зовсім не обов'язково. Якщо ви захочете не закривати програму, а просто зупинити таймер після виведення всіх запитань, то замість **Form1.Close** запишіть **Timer1.Enabled:=false**; крім того, вам доведеться знову зробити доступною кнопку «РОЗПОЧАТИ», відновити початковий напис **Label1** та назначити змінній **N** початкове значення 0. Після цього програма готова до нового диктанту з наново перемішаними запитаннями.

```

7: Begin
    Timer1.Enabled:=false;
    Button1.Enabled:=true;
    Label1.Caption:='Вставте пропущену букву З чи С';
    N:=0;
End;

```

Примітка 2: якщо вам здається, що час на відповідь занадто великий чи замалий, змініть властивість **Interval** таймера на якесь інше число (наприклад, 5000 – для 5-секундного інтервалу).

Примітка 3: якщо кількість запитань (у нашому випадку – 5) вас не влаштовує, змініть її (наприклад, до 12). Для цього додайте нові елементи масиву (у процедурі **FormCreate**) та замініть число 5 числом 12 всюди, де це потрібно (при описі масиву **Zavd**, при загадуванні випадкових номерів **N1** та **N2**, в операторі **Case**).

■5. Як завжди, після редагування збережіть зміни та скомпілюйте програму наново.