

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ГОЛОВНЕ УПРАВЛІННЯ ОСВІТИ І НАУКИ
ВИКОНАВЧОГО ОРГАНУ КИЇВРАДИ
(КИЇВСЬКОЇ МІСЬКОЇ ДЕРЖАВНОЇ АДМІНІСТРАЦІЇ)
КИЇВСЬКЕ ТЕРИТОРІАЛЬНЕ ВІДДІЛЕННЯ
МАЛОЇ АКАДЕМІЇ НАУК УКРАЇНИ**

Секція: Комп'ютерних навчальних програм

Базова дисципліна: Математика

**Тема роботи:
Навчальна ігрова програма по системам числення**

Автор роботи:

Хоптинець Вадим Андрійович
Дійсний член МАН
Оболонського району
Вихованець гуртка
"Інформатика та обчислювальна техніка"
ЦНТТМ "Сфера"
Учень 10-А класу СШ №214

Домашня адреса:

Дом. тел. _____

Педагогічний керівник:

Кириленко Наталія Сергіївна

Київ – 2009

Навчальна ігрова програма по системам числення

Зміст роботи

Вступ.....	3
1. Теоретичні відомості про системи числення	5
1.1. Представлення інформації.....	5
1.2. Двійкова система числення	5
1.3. Правила переводу чисел із десятикової системи числення в двійкову	6
2. Реалізація механізму переводу на комп'ютері.....	7
2.1 Візуальне представлення механізму переводу числа	7
2.2 Послідовність переводу числа із десятикової системи числення в двійкову	7
3. Графічне представлення механізму переводу числа	12
3.1 Складові частини вікна програми.....	12
3.2. Рішення задачі по розміщенню кнопок на формі	12
3.3 Організація виводу графіки на екран	23
4. Рекомендації щодо використання програми	24
4.1 Вимоги до комп'ютера.....	24
Висновки	25
Переваги програми	25
У наступних версіях	25
Список використаної літератури	26

Вступ

В наш час комп'ютер став невід'ємною частиною будь якої сфери життя сучасної людини. Він супроводжує людину скрізь: на роботі, під час навчання, вдома, на відпочинку, у дорозі. Зараз дуже важко знайти (якщо взагалі можливо) людину, яка б уявляла своє життя без участі комп'ютера. Це стосується не тільки наших домашніх персональних друзів, а й різних терміналів, банкоматів, електронних довідників, які зараз набирають все більшої і більшої популярності. Щоб всі ці комп'ютери могли ефективно допомагати нам у нашому житті, необхідні люди, які б не тільки доглядали їх, але й займалися їх розвитком. Такими людьми є системні адміністратори, програмісти та інженери, які займаються електронно-обчислювальною технікою. Тому зараз питання підготовки кваліфікованих кадрів в ІТ-індустрії надзвичайно актуальне.

Підготовка спеціаліста починається з основ. Такою основою в комп'ютерному світі є двійкова система числення. Жодна людина не може вважати себе серйозним спеціалістом в області інформаційних технологій, якщо не знає суті двійкової системи числення і не вміє нею користуватися. Ці основи треба починати освоювати ще в шкільному віці, бо це дасть змогу швидше зрозуміти специфіку спілкування з комп'ютером. Особливо це стосується майбутніх програмістів.

Один з найважливіших навиків, яким треба володіти при роботі з двійковою системою числення – переведення чисел із десяткової системи в двійкову систему. Найвідоміший спосіб зробити це – ділити число на два і записувати остачі. В час комп'ютеризації освіти цей процес можна зробити інтерактивним, використовуючи наприклад інтерактивну дошку. Саме такі думки підштовхнули мене створити програму, яка б допомогла зробити процес навчання більш цікавим, використовуючи сучасне обладнання.

Враховуючи це, я вирішив використати графічні бібліотеки, щоб зробити роботу в цій програмі більш приємною.

Перед написанням програми я вирішив пошукати аналоги моєї майбутньої програми, але не знайшов таких.

Отже програма уявляє собою ніби поле, в середині якого вимальовується процес переведення чисел з однієї системи в іншу. Навколо цього поля знаходяться кнопки з числами, які необхідно підставляти користувачу у процесі розрахунку. В кінці розрахунку користувачеві показується як саме формується результат. Після цього користувач або завершує роботу програми, або починає «нову гру», отримуючи інше число для переведення.

Отже, зі сказаного вище можна зробити висновки щодо причин написання програми та її мети.

Причини створення програми:

1. Прагнення створити продукт, який би зробив внесок у підготовку кваліфікованих кадрів для майбутнього розвитку національної ІТ-індустрії.
2. Бажання ще більше інтегрувати сучасне комп'ютерне обладнання у процес навчання.
3. Необхідність спонукати молодих програмістів (учнів) до створення програм, які б допомогли їм та їхнім однодумцям.
4. Відсутність аналогів.

1. Теоретичні відомості

про системи числення

1.1. Представлення інформації

Під інформацією у широкому змісті прийнято розуміти різноманітні відомості про події в суспільному житті, явищах природи, про процеси в технічних приладах. На практиці безперервні повідомлення можна подавати в дискретній формі.

Дискретне повідомлення містить набір чисел і символів. Кожне число містить цифри. Спосіб запису чисел цифровими знаками називається *системою числення*. У цифровій техніці використовуються так звані позиційні системи числення. Значення кожної цифри залежить від її положення в записі числа. Кількість різних цифр, що застосовуються у позиційній системі, називають *основою системи*. В залежності від основи, позиційні системи числення можуть бути десятковими – з основою 10, двійковими – з основою 2 тощо.

1.2. Двійкова система числення

Двійкова система числення використовує для запису чисел тільки два символи, зазвичай 0 (нуль) та 1 (одиницю). Детальніше, двійкова система числення є позиційною системою числення, база якої дорівнює двом. Завдяки тому, що таку систему доволі просто використовувати у електричних схемах, двійкова система отримала широке розповсюдження у світі обчислювальних пристроїв.

Приклади:

$$5_{10} = 101_2$$

$$8_{10} = 1000_2$$

1.3. Правила перекладу чисел із десятичної системи числення в двійкову

Щоб перевести ціле число X з основою 10 у систему з основою 2, необхідно послідовно ділити задане число і отримані в процесі ділення частки на 2 до тих пір, поки остання частка не виявиться меншою за 2. Результат перекладення записують в виді послідовності цифр зліва направо, починаючи з останньої частки і закінчуючи першою остачею (при цьому число молодшого розряду є перша остача). Всі дії в процесі ділення числа проводять в десятичній системі числення.

2. Реалізація механізму переводу на комп'ютері

2.1 Візуальне представлення механізму переводу числа із десятькової системи числення в двійкову

Представимо приклад переводу десятикового числа 5 в двійкову систему числення у вигляді матриці 4x3 (4 рядка та 3 стовпчики):

	1	2	3
1	5	2	
2	4	2	2
3	1	2	1
4		0	

Рис.2.1. Вигляд матриці

В цьому випадку можна сказати, що цифра 5 знаходиться в комірці з адресом [1,1], а цифра 4 – в комірці [2,1].

2.2 Послідовність переводу числа із десятикової системи числення в двійкову

Спробуємо записати послідовність дій для переводу числа 5 у двійкову систему числення, так, як це робить учень у зошиті:

Крок 0.

- до комірки [1,1] записується число 5

	1	2	3
1	5		
2			
3			
4			

Рис.2.2. Крок 0

Крок 1.

- рисуються лінії праворуч від числа
- до комірки [1,2] записується основа системи числення
- ставиться відмітка на комірці [2,2] (тут на наступному кроці повинно бути вписано число)

	1	2	3
1	5	2	
2		□	
3			
4			

Рис.2.3. Крок 1

Крок 2.

- до комірки [2,2] записується число; при цьому виконується перевірка - число, що вводиться, не повинно бути більш, ніж число в комірці [1,1]

	1	2	3
1	5	2	
2		2	
3			
4			

Рис.2.4. Крок 2

Крок 3.

- рисується знак віднімання
- ставиться відмітка на комірці [2,1] (тут на наступному кроці повинно бути вписано число)

	1	2	3
1	5	2	
2	□	2	
3			
4			

Рис.2.5. Крок 3

Крок 4.

- до комірки [2,1] записується число; при цьому виконується перевірка - число, що вводиться, повинно рівнятися множенню чисел із комірок [1,2] та [2,2]

	1	2	3
1	5	2	
2	4	2	
3			
4			

Рис.2.6. Крок 4

Крок 5.

- рисується лінія після введеного числа
- ставиться відмітка на комірці [3,1] (тут на наступному кроці повинно бути вписано число)

	1	2	3
1	5	2	
2	4	2	
3	□		
4			

Рис.2.7. Крок 5

Крок 6.

- до комірки [3,1] записується число; при цьому виконується перевірка - число, що вводиться, повинно бути різницею між числами із комірок [1,1] та [2,1]
- число із комірки [3,1] – це перша цифра числа результату

	1	2	3
1	5	2	
2	4	2	
3	1		
4			

Рис.2.8. Крок 6

Крок 7. (дії схожі до кроку 1)

- рисуються лінії праворуч від комірки [2,2]
- до комірки [2,3] записується основа системи числення
- ставиться відмітка на комірці [3,3] (тут на наступному кроці повинно бути вписано число)

	1	2	3
1	5	2	
2	4	2	2
3	1		□
4			

Рис.2.9. Крок 7

Крок 8. (дії схожі до кроку 2)

- до комірки [3,3] записується число; при цьому виконується перевірка - число, що вводиться, не повинно бути більш, ніж число в комірці [2,2]

	1	2	3
1	5	2	
2	4	2	2
3	1		1
4			

Рис.2.10. Крок 8

Крок 9. (дії схожі до кроку 3)

- рисується знак віднімання
- ставиться відмітка на комірці [3,2] (тут на наступному кроці повинно бути вписано число)

	1	2	3
1	5	2	
2	4	2	2
3	1	□	1
4			

Рис.2.11. Крок 9

Крок 10. (дії схожі до кроку 4)

- до комірки [3,2] записується число; при цьому виконується перевірка - число, що вводиться, повинно рівнятися множенню чисел із комірок [2,3] та [3,3]

	1	2	3
1	5	2	
2	4	2	2
3	1	2	1
4			

Рис.2.12. Крок 10

Крок 11. (дії схожі до кроку 5)

- рисується лінія після введеного числа
- ставиться відмітка на комірці [4,2] (тут на наступному кроці повинно бути вписано число)

	1	2	3
1	5	2	
2	4	2	2
3	1	2	1
4		□	

Рис.2.13. Крок 11

Крок 12. (дії схожі до кроку 6)

- до комірки [4,2] записується число; при цьому виконується перевірка - число, що вводиться, повинно бути різницею між числами із комірок [2,2] та [3,2]
- число із комірки [4,2] – це друга цифра числа результату
- виконується перевірка – якщо число в комірці [3,3] менше за основу системи числення – то перевод закінчено, в комірці [3,3] записана остання цифра числа
- таким чином, проаналізувавши комірки з остачами, ми отримуємо кінцевий результат: число із комірки [3,1], [4,2], [3,3]. В даному випадку – 101.

	1	2	3
1	5	2	
2	4	2	2
3	1	2	1
4		0	

Рис.2.14. Крок 12

3. Графічне представлення механізму переводу числа

3.1 Складові частини вікна програми

Після запуску програми перед нами постає вікно на весь екран. В ньому є дві кнопки «Нова гра» та «Вихід». Перша з них розпочинає гру, а друга завершує роботу програми. Якщо ми розпочнемо нову гру, то побачимо число у десятковій системі числення і з правого боку число 2 (основа двійкової системи) з відповідними лініями для ділення у стовпчик та відміткою, куди користувач повинен підставити число. Також побачимо по периметру вікна кнопки з числами від нуля до числа, яке ми переводимо. Щоб підставити число, треба натиснути на кнопку з відповідним числом. Якщо користувач обере вірне число, то кнопка «полетить» до вказаної відмітки і це число замінить відмітку. По такому принципу проходить подальше переведення числа, а в кінці виводиться результат.

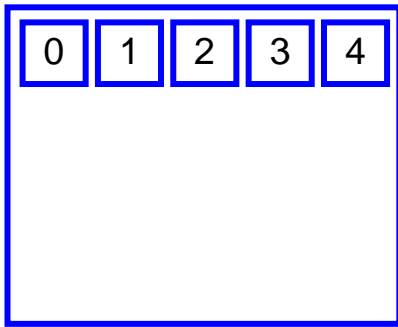
3.2. Рішення задачі по розміщенню кнопок на формі

Завдання

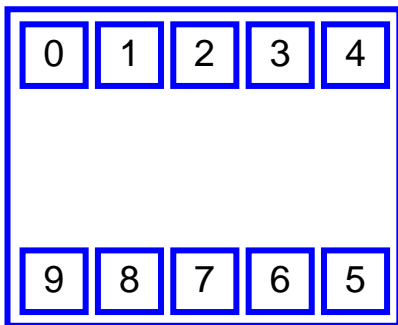
Розмістити рівномірно на формі задану кількість кнопок. В залежності від кількості кнопок, вони розміщуються на формі по горизонталі в один рядок (зверху форми), в два (зверху та знизу форми) або по колу.

Приклади:

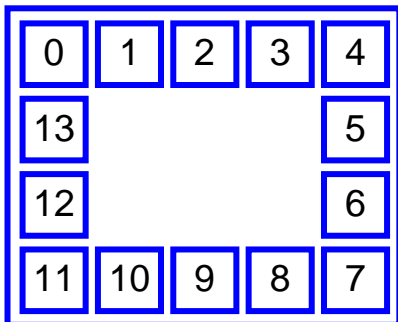
а) кількість кнопок така, що кнопки вміщуються по горизонталі в один рядок:



б) кількість кнопок така, що кнопки вміщуються по горизонталі в два рядки:



в) кількість кнопок більш ніж у варіанті «б», кнопки розміщуються по колу:



Вхідні дані:

CountButtons - кількість кнопок, які необхідно розмістити

WidthForm - ширина форми

HeightForm - висота форми

WidthBut - ширина кнопки

HeightBut - висота кнопки

LeftBut1 - ліва координата першої кнопки

TopBut1 - верхня координата першої кнопки

- MinHSpace - мінімальна горизонтальна відстань між кнопками
 MinVSpace - мінімальна вертикальна відстань між кнопками

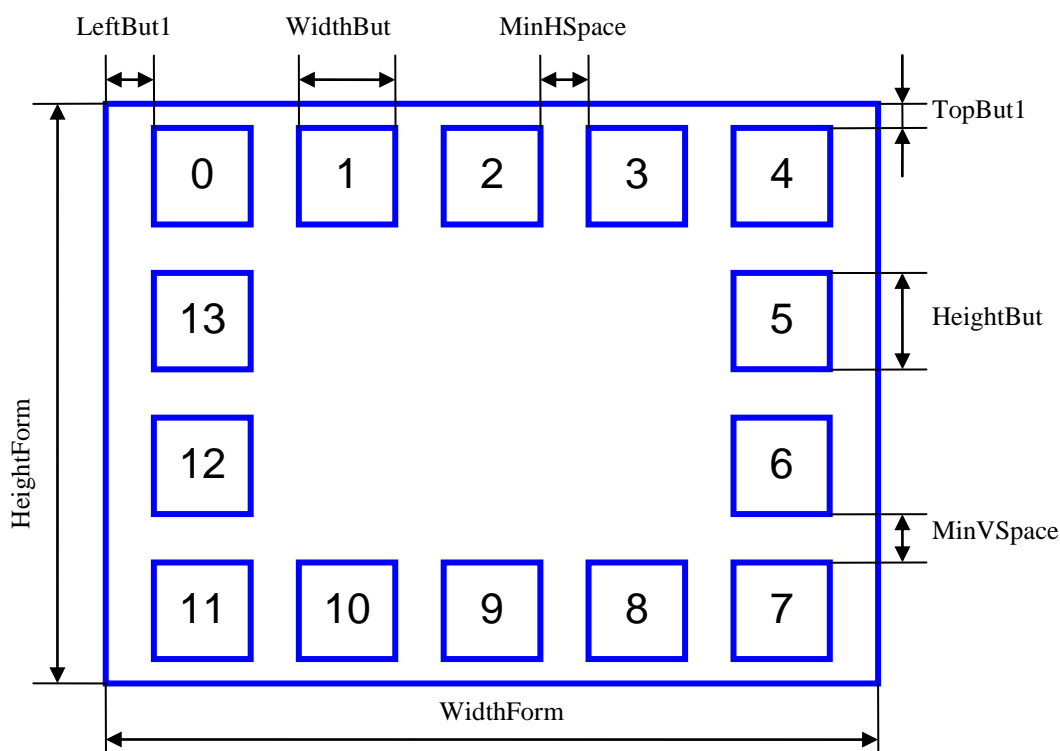


Рис.3.1. Позначення змінних величин

Алгоритм розміщення кнопок на формі

алгоритм Розміщення кнопок на формі

аргументи

- CountButtons - кількість кнопок, які необхідно розмістити
 WidthForm - ширина форми
 HeightForm - висота форми
 WidthBut - ширина кнопки
 HeightBut - висота кнопки
 LeftBut1 - ліва координата першої кнопки
 TopBut1 - верхня координата першої кнопки
 MinHSpace - мінімальна горизонтальна відстань між кнопками
 MinVSpace - мінімальна вертикальна відстань між кнопками

додаткові змінні

MaxHCountButt	- кількість кнопок, які можна розмістити по горизонталі
MaxVCountButt	- кількість кнопок, які можна розмістити по вертикалі
MaxSetCountButt	- кількість кнопок, які можна розмістити по колу
CountSets	- кількість кіл з кнопками
I	- лічильник циклу

початок

розрахунок MaxHCountButt

розрахунок MaxVCountButt

якщо $\text{CountButtons} \leq \text{MaxHCountButt}$

то

розміщуємо кнопки рівномірно по горизонталі в один рядок (зверху форми)

інакше

якщо $\text{CountButtons} \leq (\text{MaxHCountButt} * 2)$

то

розміщуємо кнопки рівномірно по горизонталі в два рядки (зверху та знизу форми)

інакше

$\text{MaxSetCountButt} := ((\text{MaxHCountButt} * 2) + ((\text{MaxVCountButt} - 2) * 2))$

якщо $\text{CountButtons} \leq \text{MaxSetCountButt}$

то

розміщуємо кнопки рівномірно по колу

інакше

$\text{CountSets} := \text{Round}(\text{CountButtons} / \text{MaxSetCountButt} + 0.5)$

початок циклу для I від 1 до CountSets

розміщуємо кнопки (для кола I) рівномірно по колу

якщо $I > 1$

то

розміщуємо кнопку переходу до попереднього кола (◀)

все

якщо $I \leq \text{CountSets}$

то

розміщуємо кнопку переходу до наступного кола (▶)

все

кінець циклу

все

все

все

кінець

Допоміжні алгоритми

алгоритм Розрахунок MaxHCountButt

аргументи

WidthForm - ширина форми

WidthBut - ширина кнопки

LeftBut1 - ліва координата першої кнопки

MinHSpace - мінімальна горизонтальна відстань між кнопками

результат

MaxHCountButt

початок

$A1 := \text{WidthForm} - \text{LeftBut1} * 2 - \text{WidthBut}$

$\text{MaxHCountButt} := A1 \text{ div } (\text{WidthBut} + \text{MinHSpace}) + 1$

кінець

алгоритм Розрахунок MaxVCountButt

аргументи

HeightForm	- висота форми
HeightBut	- висота кнопки
TopBut1	- верхня координата першої кнопки
MinVSpace	- мінімальна вертикальна відстань між кнопками

результат

MaxVCountButt

початок

$A1 := \text{HeightForm} - \text{TopBut1} * 2 - \text{HeightBut}$

$\text{MaxVCountButt} := A1 \text{ div } (\text{HeightBut} + \text{MinVSpace}) + 1$

кінець

Допоміжні процедури

Процедура розміщення кнопок рівномірно по горизонталі в один рядок

```
procedure TForm1.CreateButtons(NumFirst, NumLast, LeftBut1,
TopBut1, WidthBut: integer);
var
  i, n1, n2: integer;
  WidthSpace: integer;
  spbutTemp, Temp: TSpeedButton;
begin
  // space between two buttons
  WidthSpace := 0;
  if NumFirst <> NumLast then
    WidthSpace := ((ClientWidth - LeftBut1*2) -
WidthBut*(NumLast-NumFirst+1)) div (NumLast-NumFirst);
  // variables of loop
  if NumFirst < NumLast then
    begin
```

```

    n1 := NumFirst;
    n2 := NumLast;
end
else
begin
    n1 := NumLast;
    n2 := NumFirst;
end;
// create buttons
for i:=n1 to n2 do
begin
    // check that button already exist
    Temp:=TSpeedButton(FindComponent('But' + IntToStr(i)));
    if Temp is TSpeedButton then exit;
    // create a new button
    spbutTemp:=TSpeedButton.Create(Form1);
    with spbutTemp do
    begin
        Parent:=Form1;
        Name:='But'+IntToStr(i);
        Caption:=IntToStr(i);
        Width:=WidthBut;
        Height:=WidthBut;
        Left := LeftBut1 + (i-NumFirst)*(WidthBut + WidthSpace);
        Top:=TopBut1;
    end;
end;
end;

```

Приклади виклику процедури CreateButtons:

Приклад 1 – виклик процедури CreateButtons для розміщення зверху форми кнопок від 0 до 4:

```
CreateButtons(0, 4, 20, 10, 41);
```

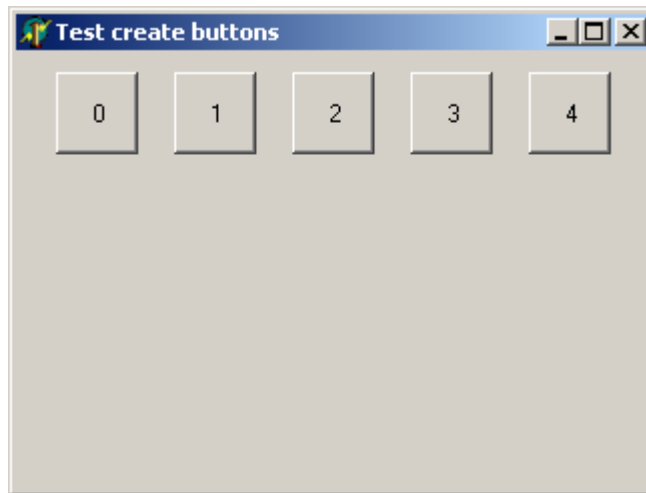


Рис. 3.2. Результат роботи прикладу 1

Приклад 2 – виклик процедури `CreateButtons` для розміщення зверху форми кнопок від 0 до 4, а знизу форми – кнопок від 5 до 9:

```
CreateButtons(0, 4, 20, 10, 41);  
CreateButtons(9, 5, 20, 170, 41);
```

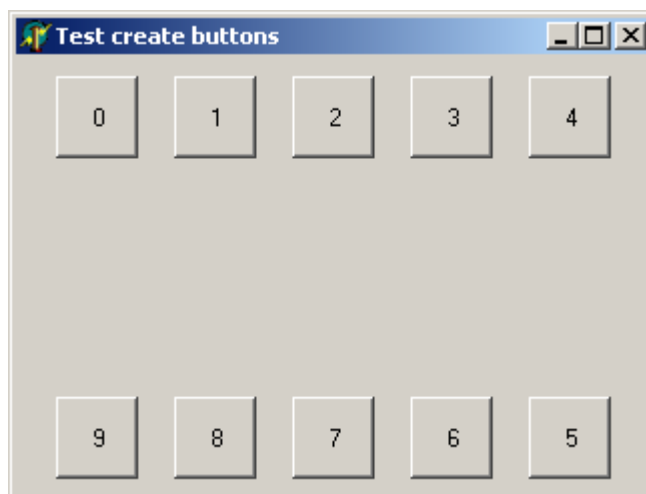


Рис. 3.3. Результат роботи прикладу 2

Приклад 3 – виклик процедури `CreateButtons` для розміщення зверху форми кнопок від 0 до 4, а знизу форми – кнопок від 5 до 7:

```
CreateButtons(0, 4, 20, 10, 41);  
CreateButtons(7, 5, 20, 170, 41);
```

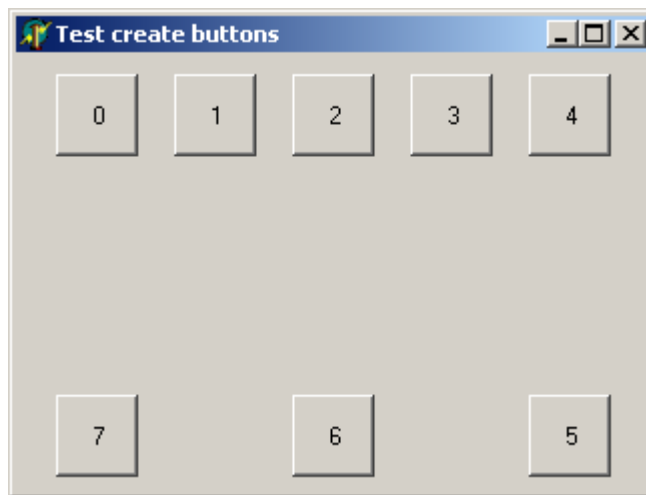


Рис. 3.4. Результат роботи прикладу 3

Приклад 4 – виклик процедури `CreateButtons` для розміщення зверху форми кнопок від 0 до 4, а знизу форми – кнопки 5:

```
CreateButtons(0, 4, 20, 10, 41);  
CreateButtons(5, 5, 20, 170, 41);
```

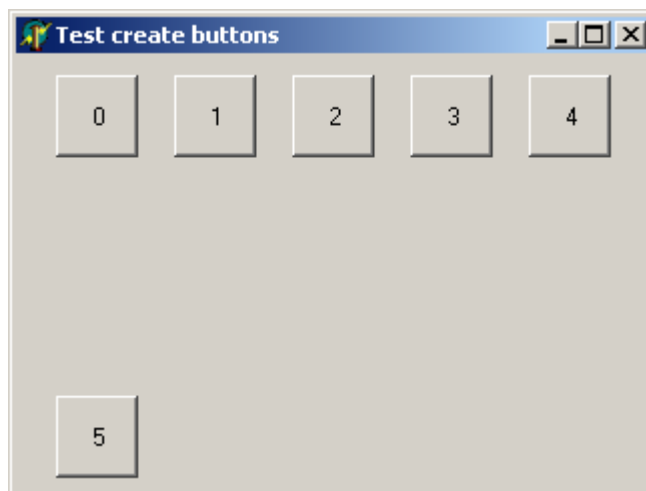


Рис. 3.5. Результат роботи прикладу 4

Процедура розміщення кнопок рівномірно по вертикалі:

```
procedure TForm1.CreateVButtons(NumFirst, NumLast, LeftBut1,  
TopBut1, WidthBut: integer);
```

```
var
```

```
    i, n1, n2, m: integer;
```

```

HeightSpace: integer;
spbutTemp, Temp: TSpeedButton;
begin
  // variables of loop
  if NumFirst < NumLast then
    begin
      n1 := NumFirst;
      n2 := NumLast;
      m := 1;
    end
  else
    begin
      n1 := NumLast;
      n2 := NumFirst;
      m := -1;
    end;
  // space between two buttons
  HeightSpace := 0;
  if NumFirst <> NumLast then
    HeightSpace := ((ClientHeight - TopBut1*2 - WidthBut*2) -
WidthBut*(n2-n1+1)) div (n2-n1+2);
  // create buttons
  for i:=n1 to n2 do
    begin
      // check that button already exist
      Temp:=TSpeedButton(FindComponent('But' + IntToStr(i)));
      if Temp is TSpeedButton then exit;
      // create a new button
      spbutTemp:=TSpeedButton.Create(Form1);
      with spbutTemp do
        begin
          Parent:=Form1;
          Name:='But'+IntToStr(i);
          Caption:=IntToStr(i);
          Width:=WidthBut;
          Height:=WidthBut;

```

```

    Left := LeftBut1;
    Top:=TopBut1 + ((i-NumFirst)*m+1)*(WidthBut +
HeightSpace);
    end;
    end;
end;

```

Приклад 5 – виклик процедур CreateButtons та CreateVButtons для розміщення кнопок рівномірно по колу:

```

CreateButtons(0, 4, 20, 10, 41);
CreateVButtons(5, 6, 256, 10, 41);
CreateButtons(11, 7, 20, 170, 41);
CreateVButtons(13, 12, 20, 10, 41);

```

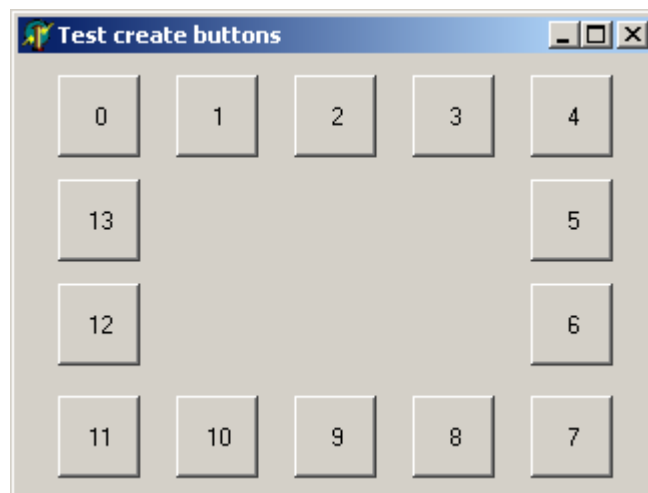


Рис. 3.6. Результат роботи прикладу 5

3.3. Організація виводу графіки на екран

Для виводу графічних зображень програма використовує бібліотеку DirectX. Після запуску програми вікно переводиться в графічний режим з роздільною здатністю екрана 800x600 пікселів.

Щоб в результаті згорання/відновлення вікна елементи не зникали, їх треба перемальовувати. Всі елементи знаходяться у відповідних масивах і обробляються в необхідний момент. Цю задачу виконує процедура TMainForm.Paint.

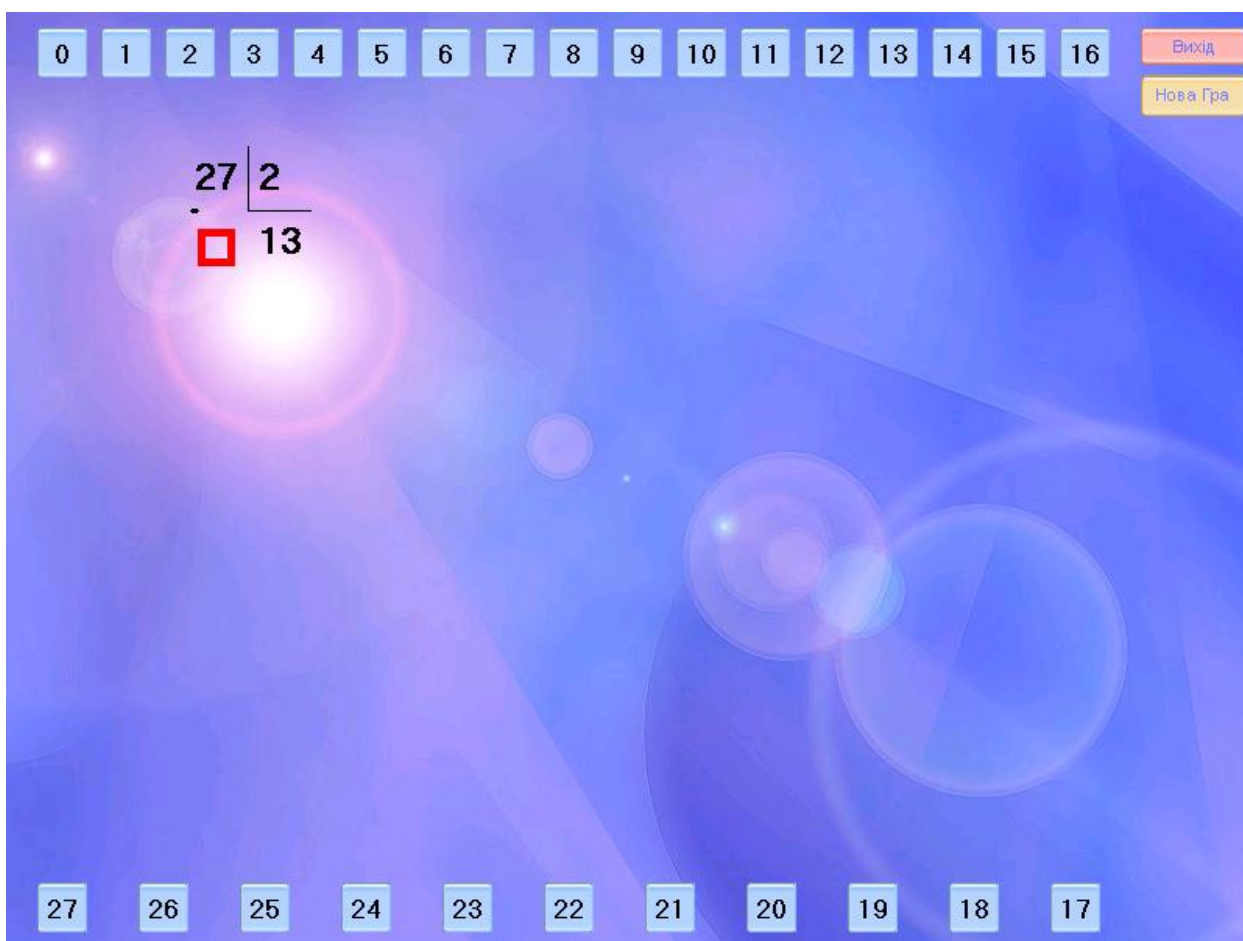


Рис. 3.7. Результат роботи програми

4. Рекомендації

щодо використання програми

4.1 Вимоги до комп'ютера

Програма призначена для використання на комп'ютері з операційною системою Windows 98/XP.

Мінімальна роздільна здатність екрана: 800 x 600 пікс.

Встановлені графічні драйвери: DirectX 7

Висновки

Переваги програми

Особливостями та перевагами даної програми є:

1. Відсутність аналогів.
2. Простота та зручність в користуванні.
3. Можливість застосування і в якості гри, і як методичний посібник для пояснення матеріалу у школах на уроках інформатики та математики або в гуртках позашкільних навчальних закладів.
4. Можливість зручної демонстрації роботи програми на інтерактивній дошці.

У наступних версіях

У наступних версіях планується зробити можливість вибору основи системи числення, вибору фонового зображення.

Список використаної літератури

1. Культин Н.Б. Delphi в задачах и примерах. – СПб.: БХВ-Петербург, 2003.
2. Зеленьак О.П. Практикум програмування на Turbo Pascal. Задачі, алгоритми и рішення. - 2-е изд., испр. и доп. - СПб.: ООО "ДиаСофтЮП", 2002.-320с.
3. Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачі по програмуванню. – М.: Наука, 1988.
4. Сван Т. Основы програмування в Delphi для Windows 95. – К.: «Диалектика», 1996.-480 с.
5. Хомоненко А.Д. и др. Delphi 7. - СПб.: БХВ-Петербург, 2003. – 1216 с.
6. Кетков Ю.Л., Кетков А.Ю. Практика програмування: Бейсик, Си, Паскаль. Самоучитель.-СПб.: БХВ-Петербург, 2002.-480 с.
7. Флєнов М.Є., Delphi глазами хакера – СПб.: БХВ-Петербург, 2004. 350с.
8. Фаронов В. Система програмування Delphi – СПб: БХВ-Петербург, 2004. 912с.
9. Ресурси інтернету