

## ЛАБОРАТОРНА РОБОТА №21

### «Обробка двовимірних масивів»

**Мета роботи:** вивчити основні функціональні можливості C++ Builder для обробки двовимірних масивів.

Для введення і виведення двовимірних масивів можна використовувати ті ж компоненти, що і для одновимірних масивів. Крім того, для цих потреб зручно використовувати компонент **StringGrid**, який входить до складу сторінки компонентів Additional і являє собою таблицю рядків. Кількість стовпців і рядків таблиці визначається властивостями **Cols[n]** і **Rows[n]**, а звернення до комірки таблиці здійснюється з використанням властивостей **Cells[x][y]**.

**Приклад 1.** Елементи прямокутної матриці  $mt(4,5)$  ввести за допомогою компонента **StringGrid**, а вивести з використанням компонента **Label**.

Розмістимо на формі компоненти:

- **Label1** — для заголовка «Початкова матриця»;
- **Label2** — для заголовка «Отримана матриця»;
- **Label3** — для виведення самої матриці;
- **StringGrid** — таблиця рядків;
- **Button1** — для створення події «Обчислити»;
- **Button2** — для створення події «Кінець роботи».

Таблиця **StringGrid** складається з рядків і стовпців, кожен з яких має свій індекс. Елементом таблиці є елемент масиву **Cells [j][i]**, де  $j$  — номер стовпця, а  $i$  — номер рядка. Як рядки, так і стовпці можуть бути фіксованими і робочими. У фіксованих комірках звичайно записується заголовна інформація, а в робочих — поточна. При розміщенні компонента **StringGrid** на формі за допомогою **Object Inspector** слід встановити такі параметри:

- ім'я таблиці — **Table**, (замість **StringGrid**);
- **RowCount = 4** — кількість рядків у таблиці;
- **ColCount = 5** — кількість стовпців;
- **FixedCols = 0** — заголовних рядків не буде;
- **Fixed Rows = 0** — заголовних стовпців не буде;
- **DefaultColWidth=30** — ширина стовпців;
- **DefaultRowHeight = 15** — висота рядків;
- **GridLineWidth = 1** — товщина ліній у таблиці.

Крім того, для забезпечення введення даних з клавіатури необхідно в нижче наведених властивостях установити значення «**true**»:

- **Option->goEditing = true;**
- **Option->Tabs = true;**

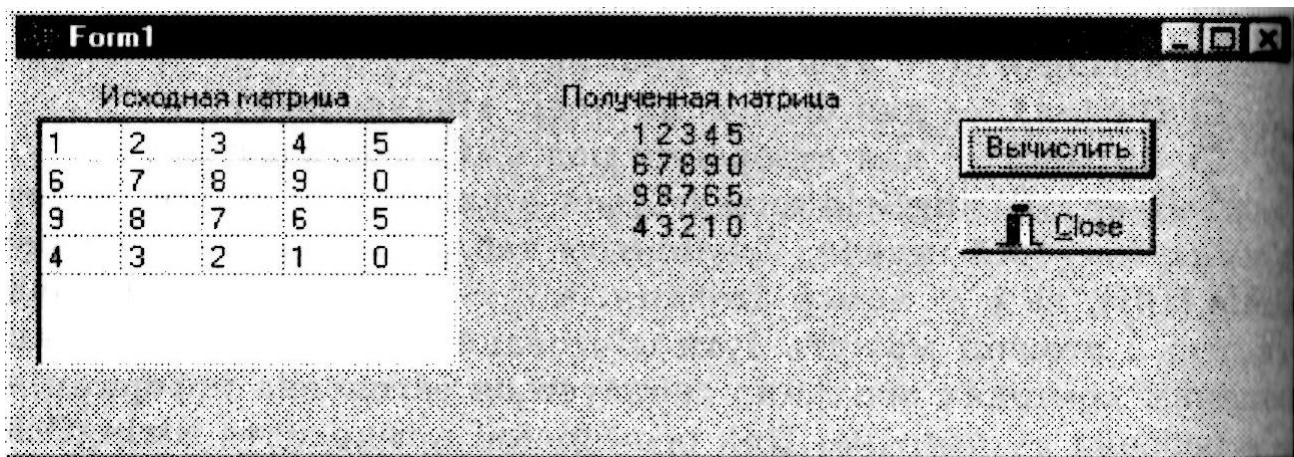
- **Option->goAlwaysShowEditor = true.**

Працюючи з таблицею, рекомендується виконати таку послідовність операцій:

- після розміщення таблиці на формі, установки її параметрів, а також розміщення інших компонентів запустити додаток на виконання і у відповідних комірках таблиці записати необхідні значення;
- перехід від одної комірки до іншої робити за допомогою клавіші **Tab** або клавішами:  $\uparrow \downarrow \rightarrow \leftarrow$ ;
- коли будуть заповнені всі комірки, натиснути кнопку «**Обчислити**».

При написанні програмного коду треба враховувати, що у властивості **StringGrid->Cells[i][j]** першим ідентифікатором є номер стовпця, а другим — номер рядка.

Форма і програмний код реалізації даного прикладу мають вигляд:



```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    const n=4, m=5; int i, j; int mt[n][m];
    //----- введення матриці
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
        {mt[i][j] = StrToInt(Table->Cells[j][i]);}
    }
    //----- виведення матриці
    Label3->Caption=" ";
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
        {
            Label3->Caption=Label3->Caption+StrToInt(mt[i][j]) + " ";
            Label3->Caption=Label3->Caption + '\n'; }
    }
}
```

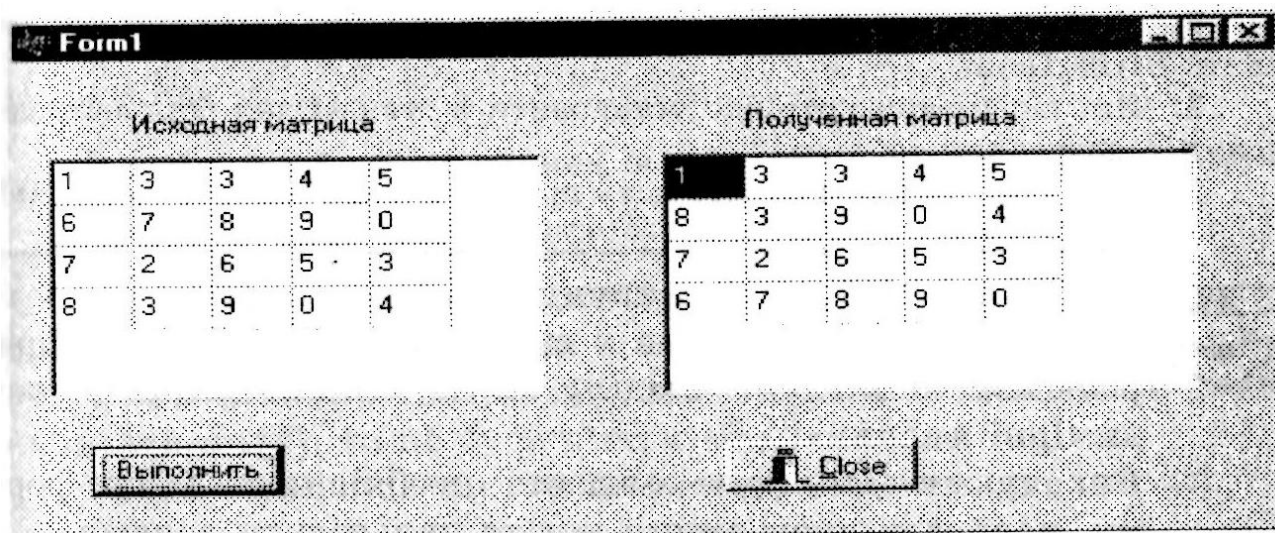
```
}  
}  
}
```

У даному прикладі для виведення матриці використаний компонент Label3 лише для того, щоб показати, як можна виконувати цю операцію. Її можна здійснити із застосуванням тільки таблиці, навіть не записуючи матрицю в пам'ять комп'ютера, тому що з елементами таблиці можна здійснювати різні операції, у тому числі і перестановку рядків.

**Приклад 2.** Ввести прямокутну матрицю  $mt(4,5)$  і здійснити в ній перестановку 2-гої 4-го рядків. Задачу розв'язати за допомогою компонента StringGrid.

Для реалізації задачі слід використовувати таблицю з ім'ям **Tb1**, в яку вводяться елементи матриці, а потім занесені значення переписати з таблиці **Tb1** у **Tb2**. В таблиці **Tb2** здійснюється перестановка місцями другого і четвертого рядків.

Розмістимо на формі дві таблиці із заголовками «**Початкова матриця**» і «**Отримана матриця**». Властивості **Name** обох таблиць **StringGrid1** і **StringGrid2** замінимо на більш зручні **Tb1** та **Tb2**. Іншим властивостям таблиць, зазначеним раніше, привласнимо ті ж значення, що й у попередньому прикладі. Для запису заголовків будемо використовувати відповідно компоненти **Label1** і **Label2**. Крім того, на формі розташуємо кнопки **Button1** з назвою «**Выполнить**» і **BitBtn** з ім'ям «**Close**». Вигляд форми і текст програмного коду наведені нижче:



```

void __fastcall TForm1::Button1Click(TObject *Sender)
{ int i, j;
  AnsiString p;

  //-----перезапись Tb1 в Tb2
  for (i = 0; i < 4; i++)
    for (j = 0; j < 5; j++)
      Tb2->Cells[j][i] = Tb1->Cells[j][i];

  //-----перестановка строк в Tb2
  for (j = 0; j < 5; j++)
  {
    p = Tb2->Cells[j][1];
    Tb2->Cells[j][1] = Tb2->Cells[j][3];
    Tb2->Cells[j][3] = p;
  }
}

```

З цього прикладу видно, по-перше, що таблицю можна заповнювати не тільки вручну, але і програмними засобами, і, по-друге, з елементами таблиці можна робити різні операції.