

Буфер обмена

Кнопки «Копировать», «Вставить», «Вырезать» есть практически в любом полноценном приложении. Эти же возможности можно использовать в своих программах. Если в программе установлены такие компоненты как Edit, Мемо и ряд других, то они уже умеют работать с буфером обмена, для чего достаточно выделить текст и щелкнуть правой кнопкой мыши, откроется меню, в котором предлагается использовать необходимые пункты. Но не все компоненты позволяют так просто произвести нужные действия и поэтому в C++ Builder есть возможность для большинства компонентов использовать буфер обмена.

Для работы с буфером обмена в C++ Builder есть класс TClipboard. Этот объект можно использовать без предварительной инициализацией, достаточно подключить модуль Clipbrd.hpp. У этого объекта имеется ряд свойств и методов, которые можно использовать в своих приложениях. Рассмотрим примеры использования этих методов и свойств.

Установим на форме компоненты Мемо и ListBox, а также 6 кнопок для реализации событий «Копировать», «Вырезать», «Вставить». Четвертую кнопку будем использовать для вставки скопированного или вырезанного фрагмента текста из Мемо в ListBox, а пятую – для очистки буфера от накопленной информации.

```
// вставим необходимый модуль
```

```
#include <Clipbrd.hpp>
```

```
// кнопка копировать
```

```
void __fastcall TForm1::CopyButtonClick(TObject *Sender)
{ Memo1->CopyToClipboard(); }
```

```
// кнопка вырезать
```

```
void __fastcall TForm1::CutButtonClick(TObject *Sender)
```

```
{ String s;
```

```
Memo1->CutToClipboard();
```

```
// после вырезания строки текста пустые строки рекомендуется удалить
```

```
int k=Memo1->Lines->Count;
```

```
for (int i=0;i<=k-1;i++)
```

```
{ s=Memo1->Lines->Strings[i];
```

```
if (s=="")
```

```
Memo1->Lines->Delete(i);
```

```
}
```

```
}
```

```
//кнопка вставить
```

```
void __fastcall TForm1::PasteButtonClick(TObject *Sender)
```

```
{ Memo1->PasteFromClipboard(); }
```

```
//кнопка вставки скопированного текста в ListBox
```

```
void __fastcall TForm1::Button1Click(TObject *Sender)
```

```
{ String s,d;
```

```
s=Clipboard()->AsText;
```

```
if (s.Pos("\r\n")==0)
```

```
ListBox1->Items->Add(s) ;
```

```
else
```

```
{ while (s.Pos("\r\n")>0) //строка в Мемо заканчивается символами \r\n – конец строки и переход на
```

```
    // новую строку, поэтому их необходимо удалить
```

```
    // если копирование осуществляется в Мемо, то этого делать не обязательно
```

```
{   d=s.SubString(1,s.Pos("\r\n")-1);
```

```
    ListBox1->Items->Add(d) ;
```

```
    s.Delete(1,s.Pos("\r\n")+1);
```

```
    if (s.Pos("\r\n")==0)
```

```
        ListBox1->Items->Add(s) ;
```

```
}
```

```
}
```

```
}
```

```
//для удаления строки в ListBox, строку необходимо выделить, а затем удалить следующим образом  
void __fastcall TForm1::DelListBoxButtonClick(TObject *Sender)  
{ ListBox1->DeleteSelected(); }
```

```
// кнопка очистить буфер  
void __fastcall TForm1::ClearButtonClick(TObject *Sender)  
{ Clipboard()->Clear(); }
```

Для копирования и вставки изображения с помощью буфера обмена, будем использовать компонент Image. Для наглядности создадим новую форму, на которой установим два компонента Image и три кнопки – «Копировать», «Вставить» и «Очистить буфер».

В первую очередь добавим библиотеку «Clipbrd.hpp» и загрузим Image1 любым изображением. Соответствующие отклики кнопок имеют следующие коды.

```
//кнопка копирования  
void __fastcall TForm1::CopyImageButtonClick(TObject *Sender)  
{ Clipboard()->Assign(Image1->Picture); }
```

```
//кнопка вставить  
void __fastcall TForm1::PasteImageButtonClick(TObject *Sender)  
{ Image2->Picture->Assign(Clipboard()); }
```

```
// кнопка очистить буфер  
void __fastcall TForm1::ClearButtonClick(TObject *Sender)  
{ Clipboard()->Clear(); }
```

Таким образом, из приведенных примеров можно перечислить основные методы и свойства объекта Clipboard:

- CopyToClipboard() – копировать;
- CutToClipboard() – вырезать;
- PasteFromClipboard() – вставить;
- Clear() - очистить буфер;
- Assign(Image1->Picture) – копировать изображение;
- Picture->Assign(Clipboard()) – вставить изображение;
- HasFormat – проверяет, какого типа данные хранятся в буфере обмена. Возможны следующие типы данных: CF_Text – буфер содержит текст, CF_BITMAP – содержит картинку, CF_PICTURE – содержит объект типа Tpicture, CF_KOMPONENT – содержит компонент.
- AsText – запись и отображение текстовой информации.

Отметим, что перечисленные методы и свойства можно использовать и обрабатывать и из других приложений. Так например, скопированные данные в Word, можно перенести в свою свое приложение, например в Мемо, или наоборот. Тоже самое можно выполнить и с изображением. Например, построив график по некоторым данным в Excel, этот график можно отобразить в объекте TImage.

Рассмотрим практический пример переноса данных, полученных в Excel и переносе этих данных с помощью буфера в приложение, отобразив их, например, в StringGrid. А после корректировки этих данных опять с помощью буфера обмена отобразить новые данные либо в Excel, либо Word. Записанные данные в Word можно преобразовать в таблицу. Затем, построив по скорректированным данным график, перенести этот график в свое приложение.

Примерный вид рабочей формы имеет вид (рис. 1):

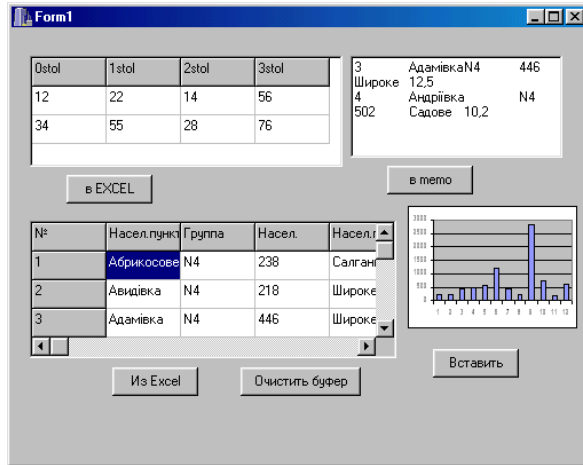
Из рисунка видно, что на форме установлены два компонента StringGrid (свойства Name - SG1 и SG2), компонент Мемо и 4 кнопки – «В Excel», «Из Excel», «В Мемо», «Очистка буфера». Соответствующие реализации кода откликов для этих кнопок имеет вид:

```
//копирование из Excel  
TForm1::FromExcelButtonClick(TObject *Sender)  
{ String s; int i,j;
```

```

String d,x;
s=Clipboard()->AsText;
if (s=="")
{ ShowMessage("Необходимо запустить EXCEL\nи скопировать данные!");
return; }
SG1->ColCount=2; SG1->RowCount=2;
// i=-1; //если вводить в фикс строки и столбцы
i=0; // если не вводить в фиксированные строки и столбцы
while (s.Length(>0)
{ i++;
if (SG1->RowCount<i+1)
SG1->RowCount=i+1;
d=s.SubString(1,s.Pos("\r\n")-1);
s.Delete(1,s.Pos("\r\n")+1);
d=d+"\t";
j=0;
// j=-1;
while (d.Length(>0)
{ j++;
if (SG1->ColCount<j+1)
SG1->ColCount=j+1;
x=d.SubString(1,d.Pos(char(9)));
x.Delete(x.Length(),1);
SG1->Cells[j][i]=x;
d.Delete(1,d.Pos(char(9)));
}
}
}
}

```



```

//копирование в Excel
TForm1::InExcelButtonClick(TObject *Sender)
{ AnsiString s;
s="";
for(int i=0;i<SG2->RowCount;i++)
{
for(int j=0;j<SG2->ColCount;j++)
s=s+SG2->Cells[j][i)+"\t";
if (s[s.Length()]=='\t')
s.Delete(s.Length(),1);
s=s+"\r\n";
}
Clipboard()->AsText=s;
}

//копирование в Мемо
TForm1::InMemoButtonClick(TObject *Sender)
{ AnsiString s;
s=Clipboard()->AsText;
Memo1->Lines->Add(s);
}

// Вставить диаграмму из Excel
void __fastcall TForm1::Button6Click(TObject *Sender)
{ Image2->Picture->Assign(Clipboard()); }

// очистка буфера
void __fastcall TForm1::ClearButtonClick(TObject *Sender)
{ Clipboard()->Clear(); }

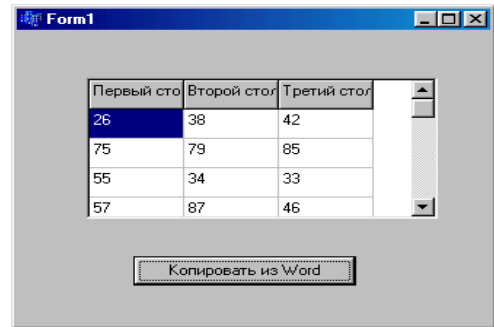
```

Примечание: При копировании данных из Excel необходимо выделить ту часть таблицы, которую следует скопировать, и выполнить команду «Копировать», либо с помощью быстрых команд, либо используя команды из списка всплывающего меню по нажатию правой кнопки мыши. При отображении данных из StringGrid в Excel, необходимо отметить ячейку в Excel, куда будут вставляться данные, и выполнить команду «Вставить». После построения диаграммы в Excel, его также необходимо выделить, скопировать, после чего вставить в Image.

Точно также можно копировать таблицы из Word.

Пример рабочей формы приведен ниже, а код копирования не приводится, так как он повторяется без изменений.

С помощью буфера обмена можно также копировать компоненты. Отметим, что при копировании компонентов необходимо создать метакласс. Пример копирования двух компонентов приведен ниже. Причем событие, созданное для кнопки, переносится в скопированную кнопку.



Копирование компонента в буфер

```
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner) //конструктор формы
{
    TMetaClass *MetaClass[1];
    MetaClass[0] = __classid(TMemo);
    RegisterClasses(&MetaClass[0], 0);
    MetaClass[1] = __classid(TButton);
    RegisterClasses(&MetaClass[1], 0);
}
```

Копирование в буфер Мемо

```
void __fastcall
TForm1::Button1Click(TObject *Sender)
{ Clipboard()->SetComponent(Memo1); }
```

Копирование в буфер Button2

```
void __fastcall
TForm1::Button7Click(TObject *Sender)
{ Clipboard()->SetComponent(Button2); }
```

Показать Мемо из буфера

```
void __fastcall TForm1::Button3Click(TObject *Sender)
{ Clipboard()->GetComponent(Owner, Form1);
Memo1->Parent=Form1;
Memo1->Top=10;
Memo1->Left=10; }
```

Показать Button из буфера

```
void __fastcall TForm1::Button6Click(TObject *Sender)
{ Clipboard()->GetComponent(Owner, Form1);
Button2->Parent=Form1;
Button2->Top=70;
Button2->Left=20; }
```

Очистка буфера

```
void __fastcall TForm1::Button4Click(TObject *Sender)
{ Clipboard()->Clear(); }
```

При нажатии на кнопке “Это кнопка” показать сообщение

```
void __fastcall TForm1::Button2Click(TObject *Sender)
{ ShowMessage("Меня скопировали"); }
```

