



آموزش
نکات
برنامه‌نویسی
سیمبین

series60.blogfa.com

مقاله شماره ۲: نحوه نمایش پنجره پیغام در سیمبین

از سری مقاله‌های «آموزش نکات برنامه‌نویسی سیمبین»

ویرایش ۱ - ۵ خرداد ۱۳۸۶

سیستم عامل سیمبین

موسی مرادی

mousamk@gmail.com

نحوه نمایش MessageBox در سیمبین:

برای استفاده کردن از Message Box ها باید فایل aknnotewrappers.h را به سورس فایلمان ضمیمه کنیم و همچنین باید کتابخانه avkon.lib را اگر اضافه نشده است، به فایل mmp اضافه کنیم.

حال فرض کنید می‌خواهیم عبارت Hello World را در یک Message Box به کاربر نشان دهیم. ابتدا باید رشته مورد نظر را تعریف کنیم:

```
_LIT(message, "Hello World!");
```

همانطور که می‌دانید، می‌توانیم توسط ماکروی _LIT یک واصف (رشته‌ها در سیمبین) تعریف کنیم. سپس از کلاس CAknInformationNote یک آبجکت می‌گیریم:

```
CAknInformationNote* infonote = new (ELeave) CAknInformationNote();
```

حالا کافی است که رشته مورد نظر را به این آبجکت بدهیم و متد اجرا کردن آن را صدا بزنیم:

```
infonote->ExecuteLD(message);
```

بعد از اجرا شدن این خط، یک Message Box با یک آیکن ا نمایش داده خواهد شد که با زدن هر کلیدی توسط کاربر، این پنجره محو می‌شود و اجرای برنامه ادامه پیدا می‌کند.



البته می‌توانید به جای اینکه متن را در اینجا بنویسید، ابتدا آن را در فایل های Resource تعریف کنید و سپس در اینجا از آن استفاده کنید. برای انجام این کار کافی است به جای کدهای بالا از کد زیر استفاده کنید:

```
TBuf<256> buf;  
iCoeEnv->ReadResource(buf, RS_R_MESSAGE);  
...  
Infonote->ExecuteLD(buf);
```

RS_R_MESSAGE را هم در فایل Resource باید به صورت زیر تعریف کرده باشید:

```
RESOURCE TBUF rs_r_message
{
    buf="Hello World!";
}
```

حتی می‌توانیم متن مورد نظر را در فایل‌های .lxx بنویسیم و در اینجا هم دوباره نام رشته مورد نظر را بنویسیم. مثلاً به جای کد بالا می‌نویسیم:

```
RESOURCE TBUF rs_r_message
{
    buf=r_message;
}
```

در فایل‌های .lxx هم متن مورد نظر را به صورت زیر می‌نویسیم:

```
#define r_message "Hello World!"
```

استفاده کردن از حالت آخری (یعنی متن‌ها را در فایل‌های .lxx بنویسیم) موقع طراحی برنامه‌های چندزبانه، کارمان را بسیار راحت‌تر می‌کند و باعث می‌شود که به راحتی بتوانیم رشته‌های مربوط به زبان‌های مختلف را، هر کدام را در یک فایل .lxx جداگانه بنویسیم و در انبوه کدهای انتخاب متن و ... گم نشویم! (ضمناً در lxx منظور از xx همان کد زبان مربوطه است).

همچنین می‌توانیم به جای پنجره پیام، یک Error Box نمایش دهیم که تنها تفاوتی که با حالت قبلی دارد، در آیکن برنامه و همچنین در صدایی است که هنگام ظاهر شدن پنجره پخش می‌شود. برای استفاده از این حالت تنها کافی است که به جای استفاده کردن از کلاس CAknInformationNote از کلاس CAknErrorNote استفاده کنیم.

همچنین برای استفاده کردن از پنجره تأیید هم کافی است که از کلاس CAknConfirmationNote استفاده کنید. تصاویر این دو تا را در زیر می‌بینید: (بجز این‌ها، کلاس‌های دیگری هم برای انواع دیگر حالت‌ها وجود دارد).



البته این کلاس‌ها دارای تعدادی تنظیمات هم هستند که می‌توانید به آنها اعمال کنید. آنها را در زیر توضیح می‌دهیم: می‌توانید صدایی که موقع نمایش پنجره پخش می‌شود را مشخص کنید. برای این کار باید از تابع زیر استفاده کنید:

```
Infonote->SetTone(/*the sound tone*/);
```

برای پارامتر این تابع، می‌توانید مقادیر زیر را استفاده کنید:

- CAknNoteDialog::ENoTone : یعنی هیچ صدایی پخش نشود.
- CAknInformationNote::EConfirmationTone : صدای پیشفرض CAknConfirmationNote است.
- CAknInformationNote::EWarningTone : صدای پیشفرض CAknWarningNote است.
- CAknInformationNote::EErrorTone : صدای پیشفرض

برای کلاس دیگر نمایش پنجره پیام هم، همین تابع و همین پارامترها وجود دارند. یک خاصیت دیگری که می‌توانید برای هم پنجره پیام، اعمال کنید، Timeout است. این خاصیت مشخص می‌کند که پنجره تا چه مقدار زمانی در معرض دید باشد. برای تعیین این خاصیت می‌توانید از تابع زیر استفاده کنید:

```
infonote->SetTimeout(/*the timeout*/);
```

برای پارامتر تابع هم می‌توانید از مقادیر زیر استفاده کنید:

- CAknNoteDialog::ENoTimeout : یعنی پنجره برای همیشه ظاهر بماند و در صورتی که کاربر کلیدی زد، ناپدید شود.
- CAknNoteDialog::EShortDTimeout : یعنی پنجره برای مدت کوتاهی ظاهر شود و سپس به طور اتوماتیک ناپدید شود.
- CAknNoteDialog::ELongTimeout : یعنی پنجره برای مدت طولانی تری بماند و در صورتی که کاربر هیچ کلیدی را نزد، بعد از مدتی خودبخود ناپدید شود.

همچنین می‌توانید مشخص کنید که آیا Text Wrapping برای این پنجره پیام فعال باشد یا نه؟ (Text Wrapping یعنی این که در صورتی که مقدار متنی که قرار است در یک سطر نمایش داده شود، از ظرفیت آن بیشتر شد، ادامه‌اش را در سطر بعدی نمایش دهد؛ برای مثال در نرم‌افزار notepad ویندوز در منوی format این گزینه وجود دارد که می‌توانید آن را بررسی کنید).

برای مشخص کردن این ویژگی هم می‌توانید از تابع زیر استفاده کنید:

```
Infonote->SetTextWrapping(/*ETrue or EFalse*/);
```

برای پارامتر این تابع هم کافی است که مقدار ETrue یا EFalse بدهید.

درباره این مقاله:

این مقاله دومین مقاله از سری مقاله‌های آموزش نکات برنامه‌نویسی سیمبین می‌باشد که در زمینه «نحوه نمایش پنجره پیغام در سیمبین» نوشته شده است.

این مقاله تألیفی است و در نوشتن این مقاله، از منبع خاصی استفاده نشده است.

برای دریافت مقالات دیگر این سری می‌توانید به [وبلاگ بنده](#) و یا [فروم مخصوص برنامه نویسی سیمبین](#) مراجعه کنید.

نویسنده: موسی مرادی کندلجی (mousamk@gmail.com)

در صورتی که هرگونه سؤال، پیشنهاد، انتقاد و یا مطلبی داشتید، می‌توانید از طریق ایمیل تماس بگیرید. ضمناً می‌توانید از طریق وبلاگ و یا فروم هم با من ارتباط داشته باشید.