



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



**OP Vzdělávání
pro konkurenceschopnost**

INVESTICE
DO ROZVOJE
VZDĚLÁVÁNÍ

Střední průmyslová škola a Vyšší odborná škola technická Brno, Sokolská 1

Šablona: Inovace a zkvalitnění výuky prostřednictvím ICT

Název: Vytváření aplikací pro systém Windows

Téma: Víceřádkové vstupy – komponenta Memo

Autor: Ing. Hodál Jaroslav, Ph.D.

Číslo: VY_32_INOVACE_27 – 15

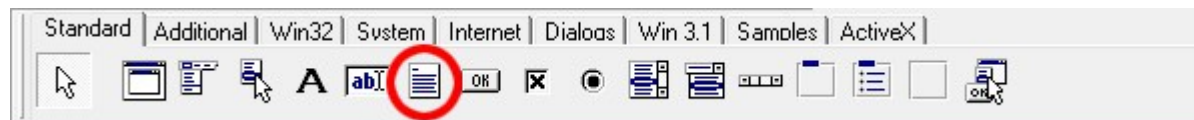
Anotace: *Materiál představuje komponentu Memo IDE Delphi a její použití pro víceřádkové vstupy v aplikacích. Materiál je určen pro 3. a 4. ročník oboru strojírenství a technické lyceum. Vytvořeno listopad 2013.*

15. Víceřádkové vstupy

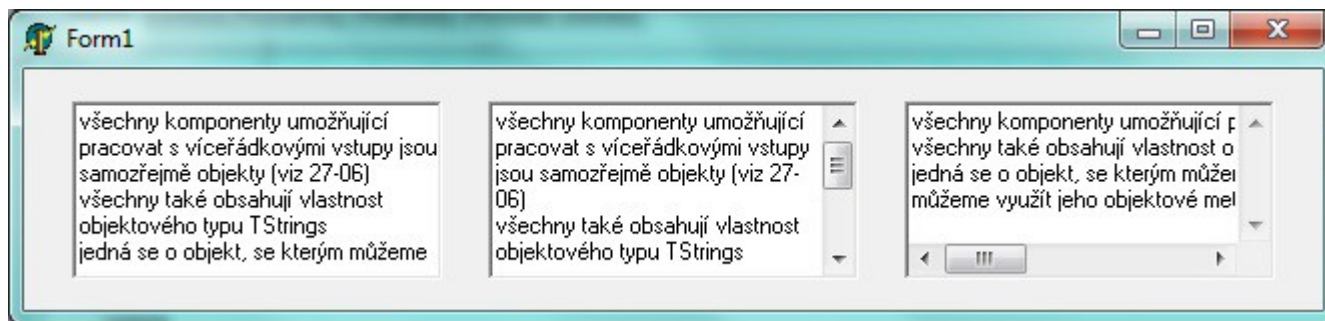
- jednou z výrazných změn oproti konzolovým aplikacím je v moderních aplikacích možnost vizuálně pracovat s **větším objemem informací**
- obvykle mají textovou podobu nebo se jako text dají zobrazovat (např. čísla)
- je tak například zcela běžné, že si necháme zobrazit obsah nějakého souboru, ten upravujeme a nakonec upravený uložíme zpět do souboru
- takovéto základní operace nám umožní některé komponenty, které umí pracovat s více textovými řetězci
- interně jsou organizovány jako dynamický seznam textů

Komponenta Memo

- Memo je komponenta umístěná na záložce standardních komponent



- funkčně nabízí kontejner pro zobrazování a editaci více řádků textu



- textová data jsou dostupná prostřednictvím vlastnosti **Lines** typu **TStrings**

TStrings – jádro víceřádkových vstupů

- všechny komponenty umožňující pracovat s víceřádkovými vstupy jsou samozřejmě **objekty** (viz 27-06)
- všechny také obsahují vlastnost objektového typu **TStrings**
- jedná se o objekt, se kterým můžeme pracovat dvěma způsoby
- můžeme využít jeho **objektové metody a vlastnosti** pro správu textů, které jsou v něm obsaženy
- zároveň se však chová také jako **dynamické pole** textových řetězců, které jsou automaticky indexovány od 0
- dynamickým polem rozumíme pole s proměnlivým počtem prvků (u běžného pole jsme museli předem zadat počet prvků)

TStrings jako objekt

- chceme-li pracovat s textovým obsahem komponenty Memo na úrovni celých řádků, využijeme jeho objektové vlastnosti a metody
- nejčastěji využívaná vlastnost je celočíselná vlastnost **Count**, která umožňuje zjistit aktuální počet řádků textu obsažených v komponentě

```
SpinEdit1.Value := Memo1.Lines.Count;
```

- přistupujeme již ke druhé úrovni objektu, používá se zde ale tečková notace stejně jako u jedné úrovně

Metody objektu TStringList (1)

- nejpoužívanější metody jsou:

```
function Add(const S: string): Integer;
```

- přidání nového řádku s textem S na konec seznamu

```
procedure Clear;
```

- vymazání všech řádků

```
procedure Delete(Index: Integer);
```

- vymazání zvoleného řádku (na pozici Index)

```
procedure Exchange(Index1, Index2:  
Integer);
```

- vymění dva řádky mezi sebou

Metody objektu TString (2)

- `function IndexOf(const S: string): Integer;`
 - zjistí pozici výskytu řádku s textem identickým s S
- `procedure Insert(Index: Integer; const S: string);`
 - vloží nový řádek na pozici Index s textem S
- `procedure Move(CurIndex, NewIndex: Integer);`
 - přesun řádku z pozice CurIndex na pozici NewIndex

Metody objektu TString (3)

- metody, které mají podobu funkce nabízejí výsledek, kterým nám něco sdělují
- metody v podobě procedury jen vykonají činnost a zpracovatelný výsledek nemají
- např. **Memo1.Lines.Clear**; prostě smaže celý obsah Mema, zatímco **p:=Memo1.Lines.Add('Ahoj');** přidá na konec Mema nový řádek obsahující zadaný text a jako výsledek nám sdělí pozici tohoto nového řádku
- pokud bychom tuto výslednou informaci nepotřebovali, můžeme metodu zavolat i bez zpracování výsledku (jako by to byla procedura) takto
Memo1.Lines.Add('Ahoj');

Metody objektu TStringList (4)

- za povšimnutí stojí ještě dvě metody, které zásadním způsobem zjednodušují spolupráci víceřádkových komponent a souborů
- **procedure LoadFromFile (const FileName: string) ;**
- **procedure SaveToFile (const FileName: string) ;**
- tyto metody bez nutnosti běžným způsobem pracovat se soubory (viz 26-07) umožňují načíst resp. zapsat data z/do textového souboru
- k jejich provedení stačí zadat textový řetězec obsahující cestu k zdrojovému resp. cílovému souboru)
Memo1.Lines.LoadFromFile ('C:\Data\data.txt') ;

Další vlastnosti komponenty Memo

- samotná komponenta Memo má ještě několik dalších vlastností, které ovlivňují její vzhled a chování
- výrazně se na fungování Mema podepisuje vlastnost **ScrollBars**, která určuje, jaké posuvníky se v Memu aktivují možnosti – žádné / horizontální / vertikální / oba
- v závislosti na jejich použití lze nejen rolovat větší objem textu, ale ovlivňuje i zalamování řádků
- v určitých kombinacích s některými hodnotami ScrollBars funguje vlastnost **WordWrap**, která právě zapíná a vypíná zalamování řádků
- chceme-li jen zobrazit text bez možnosti zásahů uživatele, využijeme logickou vlastnost **ReadOnly**