Objektově orientované programování

Přednáška č. 11

Ing. Lukáš Slánský

Ústav elektrotechniky a informatiky, Univerzita Pardubice

4. 1. 2007

Obsah

Klíčové slovo Inherited

- Inherited znamená

Klíčové slovo Inherited

- Inherited znamená zděděný

Klíčové slovo Inherited

- Inherited znamená zděděný
- Klíčovým slovem *Inherited* vyvoláte metodu předka

Klíčové slovo Inherited

- Inherited znamená zděděný
- Klíčovým slovem *Inherited* vyvoláte metodu předka
- Použití u zděděných metod, konstruktoru i destruktoru

Klíčové slovo Inherited

- Inherited znamená zděděný
- Klíčovým slovem *Inherited* vyvoláte metodu předka
- Použití u zděděných metod, konstruktoru i destruktoru

```
function Predek.Metoda(prom1:Integer):string;
begin
  ...
end;
```

Klíčové slovo Inherited

- Inherited znamená zděděný
- Klíčovým slovem *Inherited* vyvoláte metodu předka
- Použití u zděděných metod, konstruktoru i destruktoru

```
function Predek.Metoda(prom1:Integer):string;
begin
  ...
end;
function Potomek.Metoda(prom1:Integer):string;
var temp:string;
begin
  temp:=Inherited Metoda(prom1);
  ...
end;
```

Klíčové slovo inherited - příklady použití

```
constructor Potomek.Init;
```

Klíčové slovo inherited - příklady použití

```
constructor Potomek.Init;
begin
  inherited Init;
  ... další kód konstruktoru
end;
```

Klíčové slovo inherited - příklady použití

```
constructor Potomek.Init;
begin
  inherited Init;
  ... další kód konstruktoru
end;


destructor Potomek.Done;
```

Klíčové slovo inherited - příklady použití

```
constructor Potomek.Init;
begin
  inherited Init;
  ... další kód konstruktoru
end;


destructor Potomek.Done;
begin
  ... kód destruktoru
  inherited Done;
end;
```

Funkce Dispose

- Rozšířená syntaxe pro použití s objekty:

Funkce Dispose

- Rozšířená syntaxe pro použití s objekty:
- `procedure Dispose(p:Pointer, Done:Destructor);`

Funkce Dispose

- Rozšířená syntaxe pro použití s objekty:
- `procedure Dispose(p:Pointer, Done:Destructor);`
  - Vyvolá destruktor a uvolní objekt z haldy

Funkce Dispose

- Rozšířená syntaxe pro použití s objekty:
- `procedure Dispose(p:Pointer, Done:Destructor);`
  - Vyvolá destruktor a uvolní objekt z haldy

```
Dispose(Predek, Done);
Dispose(Predek, Done(10));
```

Funkce Dispose

- Rozšířená syntaxe pro použití s objekty:
- `procedure Dispose(p:Pointer, Done:Destructor);`
  - Vyvolá destruktor a uvolní objekt z haldy

```
Dispose(Predek, Done);
Dispose(Predek, Done(10));
  {U destruktoru nemá parametr význam.}
```

Funkce Fail

- Lze ji použít pouze v konstruktoru.

Funkce Fail

- Lze ji použít pouze v konstruktoru.
- Značí, že v průběhu vykonávání konstruktoru došlo k chybě a konstruktor se má přerušit.

Funkce Fail

- Lze ji použít pouze v konstruktoru.
- Značí, že v průběhu vykonávání konstruktoru došlo k chybě a konstruktor se má přerušit.
- Způsobí, že konstruktor dealokuje dynamickou instanci objektu na haldě.

Funkce Fail

- Lze ji použít pouze v konstruktoru.
- Značí, že v průběhu vykonávání konstruktoru došlo k chybě a konstruktor se má přerušit.
- Způsobí, že konstruktor dealokuje dynamickou instanci objektu na haldě.
- Do ukazatele na objekt se přiřadí hodnota *nil* a objekt je nepoužitelný.

Studentská přednáška

- Příští týden (11.1.2007).

Studentská přednáška

- Příští týden (11.1.2007).
- Návrhové vzory (Design Patterns) – Marek Pohl
- Agenty a multiagentní systémy – Lukáš Černý
- CORBA a jiné obdobné technologie – Lukáš Hlaváč