

Objektově orientované programování

Přednáška č. 3

Ing. Lukáš Slánský

Ústav elektrotechniky a informatiky, Univerzita Pardubice

19. 10. 2006



- 1 Další vlastnosti dědičnosti
 - Chování metod při dědičnosti

- 2 Abstraktní metody
 - Abstraktní metody v Pascalu



Chování tříd při dědičnosti

Atributy:



Chování tříd při dědičnosti

Atributy:

- potomek dědí po předkovi všechny atributy,



Chování tříd při dědičnosti

Atributy:

- potomek dědí po předkovi všechny atributy,
- může některé atributy přidávat, potřebuje-li je.



Chování tříd při dědičnosti

Atributy:

- potomek dědí po předkovi všechny atributy,
- může některé atributy přidávat, potřebuje-li je.

Metody:



Chování tříd při dědičnosti

Atributy:

- potomek dědí po předkovi všechny atributy,
- může některé atributy přidávat, potřebuje-li je.

Metody:

- potomek dědí všechny metody po předkovi,



Chování tříd při dědičnosti

Atributy:

- potomek dědí po předkovi všechny atributy,
- může některé atributy přidávat, potřebuje-li je.

Metody:

- potomek dědí všechny metody po předkovi,
- může některé metody přidávat, potřebuje-li je.



Chování metod při dědičnosti

Potomek dědí všechny metody po předkovi



Chování metod při dědičnosti

Potomek dědí všechny metody po předkovi včetně implementace metod.

- Ne všichni potomci potřebují stejnou implementaci metody.



Chování metod při dědičnosti

Potomek dědí všechny metody po předkovi včetně implementace metod.

- Ne všichni potomci potřebují stejnou implementaci metody.
- Ne vždy je možné přesně určit, jak metodu implementovat.



Chování metod při dědičnosti

Potomek dědí všechny metody po předkovi včetně implementace metod.

- Ne všichni potomci potřebují stejnou implementaci metody.
- Ne vždy je možné přesně určit, jak metodu implementovat. Víme jen, že existuje a potřebujeme ji používat.



Abstraktní metody

- Každý živý organismus se umí reprodukovat:

Abstraktní metody

- Každý živý organismus se umí reprodukovat:
 - ▶ snáší vejce,



Abstraktní metody

- Každý živý organismus se umí reprodukovat:
 - ▶ snáší vejce, produkují semena,



Abstraktní metody

- Každý živý organismus se umí reprodukovat:
 - ▶ snáší vejce, produkují semena, rodí,



Abstraktní metody

- Každý živý organismus se umí reprodukovat:
 - ▶ snáší vejce, produkují semena, rodí, dělí se,



Abstraktní metody

- Každý živý organismus se umí reprodukovat:
 - ▶ snáší vejce, produkují semena, rodí, dělí se, ...



Abstraktní metody

- Každý živý organismus se umí reprodukovat:
 - ▶ snáší vejce, produkuje semena, rodí, dělí se, ...
- Každé zvíře umí hledat a přijímat potravu, ...



Abstraktní metody

- Každý živý organismus se umí reprodukovat:
 - ▶ snáší vejce, produkují semena, rodí, dělí se, ...
- Každé zvíře umí hledat a přijímat potravu, ...

```
procedure CZivocich.zij;
```



Abstraktní metody

- Každý živý organismus se umí reprodukovat:
 - ▶ snáší vejce, produkují semena, rodí, dělí se, ...
- Každé zvíře umí hledat a přijímat potravu, ...

```
procedure CZivocich.zij;  
begin  
  While not mrtvy do
```



Abstraktní metody

- Každý živý organismus se umí reprodukovat:
 - ▶ snáší vejce, produkují semena, rodí, dělí se, ...
- Každé zvíře umí hledat a přijímat potravu, ...

```
procedure CZivocich.zij;  
begin  
  While not mrtvy do  
  begin  
    hledejPotravu;
```



Abstraktní metody

- Každý živý organismus se umí reprodukovat:
 - ▶ snáší vejce, produkují semena, rodí, dělí se, ...
- Každé zvíře umí hledat a přijímat potravu, ...

```
procedure CZivocich.zij;  
begin  
  While not mrtvy do  
  begin  
    hledejPotravu;  
    prijimejPotravu;
```



Abstraktní metody

- Každý živý organismus se umí reprodukovat:
 - ▶ snáší vejce, produkují semena, rodí, dělí se, ...
- Každé zvíře umí hledat a přijímat potravu, ...

```
procedure CZivocich.zij;  
begin  
  While not mrtvy do  
  begin  
    hledejPotravu;  
    prijimejPotravu;  
    if muzesSeRozmnozovat then  
      rozmnozujSe;
```



Abstraktní metody

- Každý živý organismus se umí reprodukovat:
 - ▶ snáší vejce, produkují semena, rodí, dělí se, ...
- Každé zvíře umí hledat a přijímat potravu, ...

```
procedure CZivocich.zij;  
begin  
  While not mrtvy do  
  begin  
    hledejPotravu;  
    prijimejPotravu;  
    if muzesSeRozmnozovat then  
      rozmnozujSe;  
    if potrebujesSpat then  
      spi;  
  end;  
end;
```



Abstraktní metody II.

- "Obecné zvíře" (objekt třídy CZvire) neví, jak hledat potravu, jak ji přijímat atd.



Abstraktní metody II.

- "Obecné zvíře" (objekt třídy CZvire) neví, jak hledat potravu, jak ji přijímat atd. Přesto může mít "definovanou metodu", která tyto akce používá.
- O "obecném zvířeti" můžeme říct, jak žije – viz. program ne předchozím slajdu.



Abstraktní metody II.

- "Obecné zvíře" (objekt třídy CZvire) neví, jak hledat potravu, jak ji přijímat atd. Přesto může mít "definovanou metodu", která tyto akce používá.
- O "obecném zvířeti" můžeme říct, jak žije – viz. program ne předchozím slajdu.
- Po "obecném zvířeti" nemůžeme chtít, aby žilo (vykonávalo metodu zij).



Abstraktní metody II.

- "Obecné zvíře" (objekt třídy CZvire) neví, jak hledat potravu, jak ji přijímat atd. Přesto může mít "definovanou metodu", která tyto akce používá.
- O "obecném zvířeti" můžeme říct, jak žije – viz. program ne předchozím slajdu.
- Po "obecném zvířeti" nemůžeme chtít, aby žilo (vykonávalo metodu zij).
 - ▶ Neví, jak hledat potravu. Lovit? Hrabat?



Abstraktní metody II.

- "Obecné zvíře" (objekt třídy CZvire) neví, jak hledat potravu, jak ji přijímat atd. Přesto může mít "definovanou metodu", která tyto akce používá.
- O "obecném zvířeti" můžeme říct, jak žije – viz. program ne předchozím slajdu.
- Po "obecném zvířeti" nemůžeme chtít, aby žilo (vykonávalo metodu zij).
 - ▶ Neví, jak hledat potravu. Lovit? Hrabat?
 - ▶ Neví, jak potravu přijímat.



Abstraktní metody II.

- "Obecné zvíře" (objekt třídy CZvire) neví, jak hledat potravu, jak ji přijímat atd. Přesto může mít "definovanou metodu", která tyto akce používá.
- O "obecném zvířeti" můžeme říct, jak žije – viz. program ne předchozím slajdu.
- Po "obecném zvířeti" nemůžeme chtít, aby žilo (vykonávalo metodu zij).
 - ▶ Neví, jak hledat potravu. Lovit? Hrabat?
 - ▶ Neví, jak potravu přijímat.
- Některé metody třídy nemusí mít implementaci



Abstraktní metody II.

- "Obecné zvíře" (objekt třídy CZvire) neví, jak hledat potravu, jak ji přijímat atd. Přesto může mít "definovanou metodu", která tyto akce používá.
- O "obecném zvířeti" můžeme říct, jak žije – viz. program ne předchozím slajdu.
- Po "obecném zvířeti" nemůžeme chtít, aby žilo (vykonávalo metodu zij).
 - ▶ Neví, jak hledat potravu. Lovit? Hrabat?
 - ▶ Neví, jak potravu přijímat.
- Některé metody třídy nemusí mít implementaci (nemusíme znát jejich tělo),



Abstraktní metody II.

- "Obecné zvíře" (objekt třídy CZvire) neví, jak hledat potravu, jak ji přijímat atd. Přesto může mít "definovanou metodu", která tyto akce používá.
- O "obecném zvířeti" můžeme říct, jak žije – viz. program ne předchozím slajdu.
- Po "obecném zvířeti" nemůžeme chtít, aby žilo (vykonávalo metodu zij).
- ▶ Neví, jak hledat potravu. Lovit? Hrabat?
- ▶ Neví, jak potravu přijímat.
- Některé metody třídy nemusí mít implementaci (nemusíme znát jejich tělo), přesto si můžeme být jisti tím, že existují.
- Metoda, která nemá žádnou implementaci, se označuje jako



Abstraktní metody II.

- "Obecné zvíře" (objekt třídy CZvire) neví, jak hledat potravu, jak ji přijímat atd. Přesto může mít "definovanou metodu", která tyto akce používá.
- O "obecném zvířeti" můžeme říct, jak žije – viz. program ne předchozím slajdu.
- Po "obecném zvířeti" nemůžeme chtít, aby žilo (vykonávalo metodu zij).
- ▶ Neví, jak hledat potravu. Lovit? Hrabat?
- ▶ Neví, jak potravu přijímat.
- Některé metody třídy nemusí mít implementaci (nemusíme znát jejich tělo), přesto si můžeme být jisti tím, že existují.
- Metoda, která nemá žádnou implementaci, se označuje jako *abstraktní*.



Abstraktní metody v Pascalu

- V Borland Pascalu nejsou abstraktní metody podporovány :-(



Abstraktní metody v Pascalu

- V Borland Pascalu nejsou abstraktní metody podporovány :-(
- Řešením je použít metodu pouze s voláním chyby.



Abstraktní metody v Pascalu

- V Borland Pascalu nejsou abstraktní metody podporovány :-(
- Řešením je použít metodu pouze s voláním chyby.

```
type CAbstractClass=object
    procedure abstractMethod;
end;
```



Abstraktní metody v Pascalu

- V Borland Pascalu nejsou abstraktní metody podporovány :-(
- Řešením je použít metodu pouze s voláním chyby.

```
type CAbstractClass=object
    procedure abstractMethod;
end;

procedure CAbstractClass.abstractMethod;
begin
    WriteLn('Volání abstraktní metody');
    halt(1);
end;
```



Abstraktní metody v Pascalu II.

Nebo lépe s podporou Borlandských knihoven:

```
uses Objects;  
type CAbstractClass=object  
    procedure abstractMethod;  
end;
```



Abstraktní metody v Pascalu II.

Nebo lépe s podporou Borlandských knihoven:

```
uses Objects;  
type CAbstractClass=object  
    procedure abstractMethod;  
end;  
procedure CAbstractClass.abstractMethod;  
begin  
    abstract;  
end;
```

