Object Oriented Programming

Lecture No. 8

Ing. Lukáš Slánský

Univerzita Pardubice

15. 12. 2008

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Obsah

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Základní pojmy OOP

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Základní pojmy OOP

- Objekt – Object

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Základní pojmy OOP

- Objekt – Object
- Třída – Class

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Základní pojmy OOP

- Objekt – Object
- Třída – Class
- Zpráva – Message

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Základní pojmy OOP

- Objekt – Object
- Třída – Class
- Zpráva – Message
- Atribut – Attribute

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Základní pojmy OOP

- Objekt – Object
- Třída – Class
- Zpráva – Message
- Atribut – Attribute
- Metoda – Method

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Základní pojmy OOP

- Zapouzdření – Encapsulation

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Základní pojmy OOP

- Zapouzdření – Encapsulation
- Dědičnost/Zobecnění (generalizace) – Inheritance/Generalisation

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Základní pojmy OOP

- Zapouzdření – Encapsulation
- Dědičnost/Zobecnění (generalizace) – Inheritance/Generalisation
- Předek – Ancestor

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Základní pojmy OOP

- Zapouzdření – Encapsulation
- Dědičnost/Zobecnění (generalizace) – Inheritance/Generalisation
- Předek – Ancestor
- Potomek – Descendant

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Základní pojmy OOP

- Zapouzdření – Encapsulation
- Dědičnost/Zobecnění (generalizace) – Inheritance/Generalisation
- Předek – Ancestor
- Potomek – Descendant
- Abstraktní metoda – Abstract Method

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Základní pojmy OOP

- Zapouzdření – Encapsulation
- Dědičnost/Zobecnění (generalizace) – Inheritance/Generalisation
- Předek – Ancestor
- Potomek – Descendant
- Abstraktní metoda – Abstract Method
- Kompatibilita tříd – Class Compatibility

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Vztahy mezi objekty

- Dědičnost

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Vztahy mezi objekty

- Dědičnost
- Asociace – Association

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Vztahy mezi objekty

- Dědičnost
- Asociace – Association
- Agregace – Aggregation

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Vztahy mezi objekty

- Dědičnost
- Asociace – Association
- Agregace – Aggregation
- Kompozice – Composition

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Polymorfizmus

- Včasná vazba – Early Binding

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Polymorfizmus

- Včasná vazba – Early Binding
- Pozdní vazba – Late Binding

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Polymorfizmus

- Včasná vazba – Early Binding
- Pozdní vazba – Late Binding
- Polymorfizmus – Polymorphism

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Polymorfizmus

- Včasná vazba – Early Binding
- Pozdní vazba – Late Binding
- Polymorfizmus – Polymorphism
  - = mnohotvárnost – jeden příkaz se chová různě dle dat (instance objektu)

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Polymorfizmus

- Včasná vazba – Early Binding
- Pozdní vazba – Late Binding
- Polymorfizmus – Polymorphism
  - = mnohotvárnost – jeden příkaz se chová různě dle dat (instance objektu)
- Virtuální metoda – Virtual Method

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Polymorfizmus

- Včasná vazba – Early Binding
- Pozdní vazba – Late Binding
- Polymorfizmus – Polymorphism
  - = mnohotvárnost – jeden příkaz se chová různě dle dat (instance objektu)
- Virtuální metoda – Virtual Method
- Tabulka virtuálních metod (VMT) – Virtual Methods Table

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Polymorfizmus

- Včasná vazba – Early Binding
- Pozdní vazba – Late Binding
- Polymorfizmus – Polymorphism
  - = mnohotvárnost – jeden příkaz se chová různě dle dat (instance objektu)
- Virtuální metoda – Virtual Method
- Tabulka virtuálních metod (VMT) – Virtual Methods Table
- Konstruktor – Constructor

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Polymorfizmus

- Včasná vazba – Early Binding
- Pozdní vazba – Late Binding
- Polymorfizmus – Polymorphism
  - = mnohotvárnost – jeden příkaz se chová různě dle dat (instance objektu)
- Virtuální metoda – Virtual Method
- Tabulka virtuálních metod (VMT) – Virtual Methods Table
- Konstruktor – Constructor
- Destruktor – Destructor

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Základy UML

- Unified Modeling Language – UML

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Základy UML

- Unified Modeling Language – UML
- Diagram tříd – Class Diagram

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Základy UML

- Unified Modeling Language – UML
- Diagram tříd – Class Diagram
  - Zápis třídy

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Základy UML

- Unified Modeling Language – UML
- Diagram tříd – Class Diagram
  - Zápis třídy
  - Zápis prvků (atributů, metod), jejich přístupnost

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Základy UML

- Unified Modeling Language – UML
- Diagram tříd – Class Diagram
  - Zápis třídy
  - Zápis prvků (atributů, metod), jejich přístupnost
  - Generalizace

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Základy UML

- Unified Modeling Language – UML
- Diagram tříd – Class Diagram
  - Zápis třídy
  - Zápis prvků (atributů, metod), jejich přístupnost
  - Generalizace
  - Agregace

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Základy UML

- Unified Modeling Language – UML
- Diagram tříd – Class Diagram
  - Zápis třídy
  - Zápis prvků (atributů, metod), jejich přístupnost
  - Generalizace
  - Agregace
  - Kompozice

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Základy UML

- Unified Modeling Language – UML
- Diagram tříd – Class Diagram
  - Zápis třídy
  - Zápis prvků (atributů, metod), jejich přístupnost
  - Generalizace
  - Agregace
  - Kompozice
  - Asociace

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Vícevrstvá architektura

- Trojvrstvý návrh systému – Three-tier System Design

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Vícevrstvá architektura

- Trojvrstvý návrh systému – Three-tier System Design
- Prezentační vrstva – Presentation Tier

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Vícevrstvá architektura

- Trojvrstvý návrh systému – Three-tier System Design
- Prezentační vrstva – Presentation Tier
- Datová vrstva – Data Tier

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Vícevrstvá architektura

- Trojvrstvý návrh systému – Three-tier System Design
- Prezentační vrstva – Presentation Tier
- Datová vrstva – Data Tier
- Výkonná (bussines) vrstva – Bussines Tier

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Best Practices v OO programech

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Best Practices v OO programech

- Používejte třídy jako třídy (ne „zásobníky na procedury/funkce“)
  - Včetně správného použití atributů

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Best Practices v OO programech

- Používejte třídy jako třídy (ne „zásobníky na procedury/funkce“)
  - Včetně správného použití atributů
- Nepoužívejte globální proměnné – veškerá data lze předávat pomocí parametrů

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Best Practices v OO programech

- Používejte třídy jako třídy (ne „zásobníky na procedury/funkce“)
  - Včetně správného použití atributů
- Nepoužívejte globální proměnné – veškerá data lze předávat pomocí parametrů
- Používejte čitelné identifikátory (tříd, atributů, metod, parametrů)

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Best Practices v OO programech

- Používejte třídy jako třídy (ne „zásobníky na procedury/funkce“)
  - Včetně správného použití atributů
- Nepoužívejte globální proměnné – veškerá data lze předávat pomocí parametrů
- Používejte čitelné identifikátory (tříd, atributů, metod, parametrů)
- Využívejte vztahů mezi objekty – většinu lze modelovat pomocí *ukazatelů* na objekty

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Best Practices v OO programech

- Používejte třídy jako třídy (ne „zásobníky na procedury/funkce“)
  - Včetně správného použití atributů
- Nepoužívejte globální proměnné – veškerá data lze předávat pomocí parametrů
- Používejte čitelné identifikátory (tříd, atributů, metod, parametrů)
- Využívejte vztahů mezi objekty – většinu lze modelovat pomocí *ukazatelů* na objekty
- Nenadužívejte dědičnost

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Best Practices v OO programech

- Používejte třídy jako třídy (ne „zásobníky na procedury/funkce“)
  - Včetně správného použití atributů
- Nepoužívejte globální proměnné – veškerá data lze předávat pomocí parametrů
- Používejte čitelné identifikátory (tříd, atributů, metod, parametrů)
- Využívejte vztahů mezi objekty – většinu lze modelovat pomocí *ukazatelů* na objekty
- Nenadužívejte dědičnost
- Myslete na případné jiné použití objektu

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Best Practices v OO programech

- Používejte třídy jako třídy (ne „zásobníky na procedury/funkce“)
  - Včetně správného použití atributů
- Nepoužívejte globální proměnné – veškerá data lze předávat pomocí parametrů
- Používejte čitelné identifikátory (tříd, atributů, metod, parametrů)
- Využívejte vztahů mezi objekty – většinu lze modelovat pomocí *ukazatelů* na objekty
- Nenadužívejte dědičnost
- Myslete na případné jiné použití objektu
  - Zejména vstup/výstup jen v objektech (metodách), které mají tyto akce „v popisu práce“

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Best Practices v OO programech

- Používejte třídy jako třídy (ne „zásobníky na procedury/funkce“)
  - Včetně správného použití atributů
- Nepoužívejte globální proměnné – veškerá data lze předávat pomocí parametrů
- Používejte čitelné identifikátory (tříd, atributů, metod, parametrů)
- Využívejte vztahů mezi objekty – většinu lze modelovat pomocí *ukazatelů* na objekty
- Nenadužívejte dědičnost
- Myslete na případné jiné použití objektu
  - Zejména vstup/výstup jen v objektech (metodách), které mají tyto akce „v popisu práce“
  - Zápis (čtení) dat za pomoci oddělené datové vrstvy

Univerzita Pardubice
Fakulta elektrotechniky a informatiky