

Usage of modular scissors in the implementation of FEM

Dalibor Frydrych

PANM 2010
6.-11. června 2010
Dolní Maxov

8. června 2010

- 1 Úvod
 - Zúžený pohled na OOP
- 2 Základy objektově orientovaného přístupu
 - Objektové paradigma
 - Úrovně abstrakce
 - Použití UML
 - Návrhové vzory
- 3 Užití OOP a modulárních nůžek v FEM
 - OOP pohled na FEM
 - Testování
- 4 Závěr

Mýty v OOP

- Zapouzdření
- Polymorfismus
- Dědičnost

Programování \neq Přístup

Objektově orientovaný přístup je jazykově nezávislý a netýká se pouze tvorby zdrojového kódu.

Mýty v OOP

- Zapouzdření
- Polymorfismus
- Dědičnost

Programování \neq Přístup

Objektově orientovaný přístup je jazykově nezávislý a netýká se pouze tvorby zdrojového kódu.

5. základních vlastností OO jazyka^a

^aAlan Kay - Smalltalk

- **Všechno je objekt.**
- Každý objekt má vlastní paměť, vytvořenou na základě jiných objektů.
- Program je tvořen skupinou objektů, které si posíláním zpráv navzájem říkají, co je třeba udělat.
- Každý objekt má určitý typ.
- Všechny objekty určitého typu mohou přijímat tytéž zprávy.

5. základních vlastností OO jazyka^a

^aAlan Kay - Smalltalk

- Všechno je objekt.
- Každý objekt má vlastní paměť, vytvořenou na základě jiných objektů.
- Program je tvořen skupinou objektů, které si posíláním zpráv navzájem říkají, co je třeba udělat.
- Každý objekt má určitý typ.
- Všechny objekty určitého typu mohou přijímat tytéž zprávy.

5. základních vlastností OO jazyka^a

^aAlan Kay - Smalltalk

- Všechno je objekt.
- Každý objekt má vlastní paměť, vytvořenou na základě jiných objektů.
- Program je tvořen skupinou objektů, které si posíláním zpráv navzájem říkají, co je třeba udělat.
- Každý objekt má určitý typ.
- Všechny objekty určitého typu mohou přijímat tytéž zprávy.

5. základních vlastností OO jazyka^a

^aAlan Kay - Smalltalk

- Všechno je objekt.
- Každý objekt má vlastní paměť, vytvořenou na základě jiných objektů.
- Program je tvořen skupinou objektů, které si posíláním zpráv navzájem říkají, co je třeba udělat.
- Každý objekt má určitý typ.
- Všechny objekty určitého typu mohou přijímat tytéž zprávy.

5. základních vlastností OO jazyka^a

^aAlan Kay - Smalltalk

- Všechno je objekt.
- Každý objekt má vlastní paměť, vytvořenou na základě jiných objektů.
- Program je tvořen skupinou objektů, které si posíláním zpráv navzájem říkají, co je třeba udělat.
- Každý objekt má určitý typ.
- Všechny objekty určitého typu mohou přijímat tytéž zprávy.

Pojmový mišmaš

- Třída vs. Objekt (instance)
- Attribute, prvky, proměnné (instance, třídy)
- Metody, zprávy, funkce, operace, události (statické, virtuální, abstraktní)
- Přetěžování, Překrývání

Pojmový mišmaš

- Třída vs. Objekt (instance)
- Attribute, prvky, proměnné (instance, třídy)
- Metody, zprávy, funkce, operace, události (statické, virtuální, abstraktní)
- Přetěžování, Překrývání

Pojmový mišmaš

- Třída vs. Objekt (instance)
- Atributy, prvky, proměnné (instance, třídy)
- Metody, zprávy, funkce, operace, události (statické, virtuální, abstraktní)
- Přetěžování, Překrývání

Pojmový mišmaš

- Třída vs. Objekt (instance)
- Atributy, prvky, proměnné (instance, třídy)
- Metody, zprávy, funkce, operace, události (statické, virtuální, abstraktní)
- Přetěžování, Překrývání

4. pilíře OOP

- Objektové paradigma
- Úrovně abstrakce
- Použití UML
- Návrhové vzory

4. pilíře OOP

- Objektové paradigma
- Úrovně abstrakce
- Použití UML
- Návrhové vzory

4. pilíře OOP

- Objektové paradigma
- Úrovně abstrakce
- Použití UML
- Návrhové vzory

4. pilíře OOP

- Objektové paradigma
- Úrovně abstrakce
- Použití UML
- Návrhové vzory

Objektové paradigma

Anonymita klienta

Relativita pohledu vnější/vnitřní

- Klient vidí služby - nevidí (vnitřní) implementaci
- Implementaci nezáleží na tom, kdo (který klient) ji volá^a

^aje BLBUVZDORNÁ

Objektové paradigma

Anonymita klienta

Relativita pohledu vnější/vnitřní

- Klient vidí služby - nevidí (vnitřní) implementaci
- Implementaci nezáleží na tom, kdo (který klient) ji volá^a

^aje BLBUVZDORNÁ

Objektové paradigma

Anonymita klienta

Relativita pohledu vnější/vnitřní

- Klient vidí služby - nevidí (vnitřní) implementaci
- Implementaci nezáleží na tom, kdo (který klient) ji volá^a

^aje BLBUVZDORNÁ

Dichotomie typu

prvek

META

typ prvku

prvek

META

třída

META

meta-třída

Dichotomie - řecky:

- dicha - na dvakrát
- tome - řez

Vícenásobná abstrakce

prvek

M

fyzika

M

matematika

M

třída

M

meta-třída

Dichotomie typu

prvek

META

typ prvku

prvek

META

třída

META

meta-třída

Dichotomie - řecky:

- dicha - na dvakrát
- tome - řez

Vícenásobná abstrakce

prvek

M

fyzika

M

matematika

M

třída

M

meta-třída

Dichotomie typu

prvek

META

typ prvku

prvek

META

třída

META

meta-třída

Dichotomie - řecky:

- dicha - na dvakrát
- tome - řez

Vícenásobná abstrakce

prvek

M

fyzika

M

matematika

M

třída

M

meta-třída

Úrovně abstrakce

- Analytické modelování
- Design
- Programování

Realizace

- Analytik
- Designer
- Programátor

Úrovně abstrakce

- Analytické modelování
- Design
- Programování

Realizace

- Analytik
- Designer
- Programátor

The Object Management Group

<http://www.omg.org>

Unified Modeling Language

<http://www.uml.org>

Diagramy

The Object Management Group

<http://www.omg.org>

Unified Modeling Language

<http://www.uml.org>

Diagramy

- Simulční diagramy

The Object Management Group

<http://www.omg.org>

Unified Modeling Language

<http://www.uml.org>

Diagramy

- Strukturní diagramy
- Diagramy chování
- Diagramy interakce

The Object Management Group

<http://www.omg.org>

Unified Modeling Language

<http://www.uml.org>

Diagramy

- Strukturní diagramy
- Diagramy chování
- Diagramy interakce

The Object Management Group

<http://www.omg.org>

Unified Modeling Language

<http://www.uml.org>

Diagramy

- Strukturní diagramy
- Diagramy chování
- Diagramy interakce

The Object Management Group

<http://www.omg.org>

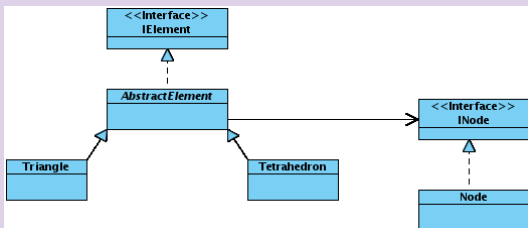
Unified Modeling Language

<http://www.uml.org>

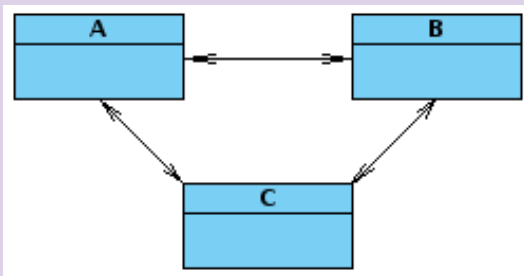
Diagramy

- Strukturní diagramy
- Diagramy chování
- Diagramy interakce

Strukturní diagram - diagram tříd



Cirkulární vazba



Problém 1 → Řešení 1

Problém 2 → Řešení 2

Návrhový vzor

Problém → Řešení

Účastníci vzoru

Model vzoru s účastníky(parametry)

Návrhový vzor

- Proč vymýšlet vymyšlené
- Proč popisovat publikované

Problém 1 → Řešení 1

Problém 2 → Řešení 2

Návrhový vzor

Problém → Řešení

Účastníci vzoru

Model vzoru s účastníky(parametry)

Návrhový vzor

- Proč vymýšlet vymyšlené
- Proč popisovat publikované

Problém 1 → Řešení 1

Problém 2 → Řešení 2

Návrhový vzor

Problém → Řešení

Účastníci vzoru

Model vzoru s účastníky(parametry)

Návrhový vzor

- Proč vymýšlet vymyšlené
- Proč popisovat publikované

Problém 1 → Řešení 1

Problém 2 → Řešení 2

Návrhový vzor

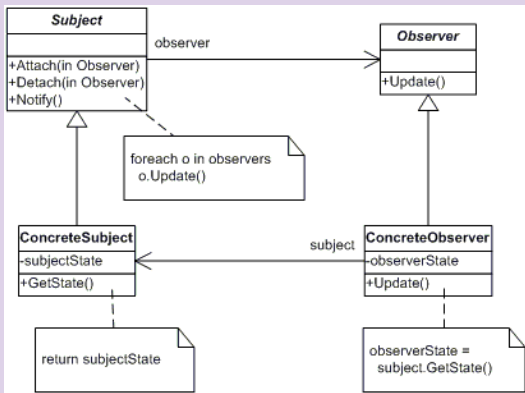
Problém → Řešení

Účastníci vzoru

Model vzoru s účastníky(parametry)

Návrhový vzor

- Proč vymýšlet vymyšlené
- Proč popisovat publikované

Návrhový vzor *Observer*

Literatura

- E.Gamma, R.Helm, R.Johnson, J.Vlissides: *Design Patterns: Elements of Reusable Object-Oriented Software* (23 vzorů)
- M.Fowler: *Analysis patterns*
- M.Fowler: *Pattern of Enterprise Application Architecture*
Enterprise Architecture Patterns - (OOP+RDB, hibernate)
(cca 50 vzorů)
- Integration Patterns
- Real-Time design patterns

Literatura

- E.Gamma, R.Helm, R.Johnson, J.Vlissides: *Design Patterns: Elements of Reusable Object-Oriented Software* (23 vzorů)
- M.Fowler: *Analysis patterns*
- M.Fowler: *Pattern of Enterprise Application Architecture*
Enterprise Architecture Patterns - (OOP+RDB, hibernate)
(cca 50 vzorů)
- Integration Patterns
- Real-Time design patterns

Literatura

- E.Gamma, R.Helm, R.Johnson, J.Vlissides: *Design Patterns: Elements of Reusable Object-Oriented Software* (23 vzorů)
- M.Fowler: *Analysis patterns*
- M.Fowler: *Pattern of Enterprise Application Architecture*
Enterprise Architecture Patterns - (OOP+RDB, hibernate)
(cca 50 vzorů)
- Integration Patterns
- Real-Time design patterns

Literatura

- E.Gamma, R.Helm, R.Johnson, J.Vlissides: *Design Patterns: Elements of Reusable Object-Oriented Software* (23 vzorů)
- M.Fowler: *Analysis patterns*
- M.Fowler: *Pattern of Enterprise Application Architecture*
Enterprise Architecture Patterns - (OOP+RDB, hibernate)
(cca 50 vzorů)
- Integration Patterns
- Real-Time design patterns

Literatura

- E.Gamma, R.Helm, R.Johnson, J.Vlissides: *Design Patterns: Elements of Reusable Object-Oriented Software* (23 vzorů)
- M.Fowler: *Analysis patterns*
- M.Fowler: *Pattern of Enterprise Application Architecture*
Enterprise Architecture Patterns - (OOP+RDB, hibernate)
(cca 50 vzorů)
- Integration Patterns
- Real-Time design patterns

Modulární nůžky

Rozdělení projektu na nezávislé, samostatné fungující části

Re-Use

Nezávislé, samostatně fungující části lze použít i v jiných projektech

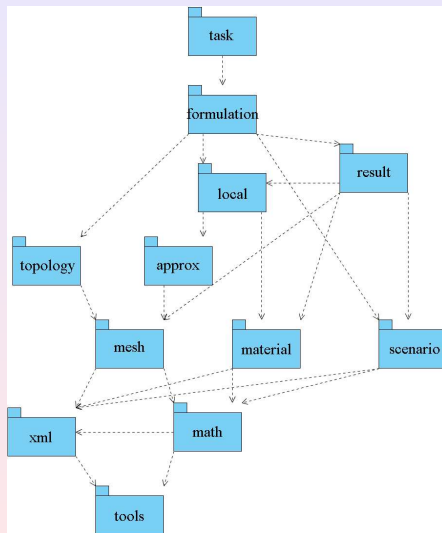
Modulární nůžky

Rozdělení projektu na nezávislé, samostatné fungující části

Re-Use

Nezávislé, samostatně fungující části lze použít i v jiných projektech

OOP pohled na FEM



Literatura

- The Object Management Group, <http://www.omg.org>
- Unified Modeling Language, <http://www.uml.org>
- I.Kraval: Objektové modelování pomocí UML v praxi, 2005, <http://www.objects.cz>
- Examples to Accompany: Design Patterns
<http://acervus.nti.tul.cz/NMP/References/pattern-examples.pdf>
- B.Eckel: Thinking in JAVA <http://www.bruceeckel.com>
- B.Eckel: Thinking in Patterns <http://www.bruceeckel.com>
- J.Schmuller: Myslíme v jazyku UML, GRADA Publishing, Praha, 2001

Děkuji za pozornost