DATABÁZOVÉ A ZNALOSTNÍ IS P5

2008-03-20

VYMEZENÍ POJMU „INFORMATIKA“:

Computing Sciences:
- ✓ Computer Science:
- ✓ Information Systems
- ✓ Information Technology
- ✓ Software Engineering
- ✓ Computer Engineering

Dimenze:
- ✓ **1. dimenze:**
  - 1. vrstva: Computer Hardware and Architecture
  - 2. vrstva: Systems Infrastructure
  - 3. vrstva: Software Methods and Technologies
  - 4. vrstva: Application Technologies
  - 5. vrstva: Organizational Issues and Information Systems
- ✓ **2. dimenze:**
  - Therory Principles Innovation
  - Application Deployment Comfiguration

UML:

UML – Unified Modeling Language:
- ✓ **Vývoj:**

OMT-1 → OMT-2
Boch 91 → Booch 93
Other methods } Unified Method 0.8 (OOSE) } UML 0.9 & 0.91 } UML 1.0 (UML Partners' Expertise)

- ✓ UML je ISO normou.
- ✓ UML is a graphical langure for visualizing, specifying, constructing and documenting the artifacts of a software-intensive system.
- ✓ **UML – diagramy:**
  - **Use Case Diagram**
  - **Class Diagram**
  - **Behavior Diagrams:**
    - Statechart Diagram
    - Activity Diagram
    - Interaction Diagrams:
      - Sequence Diagram
      - Collaboration Diagram
  - **Implementation Diagrams:**
    - Component Diagram
    - Deployment Diagram
- ✓ UML definuje symboly, konceptuální pojmy a vazby, ze kterých lze jednotlivé diagramy poskládat. Digramy lze využívat na vyjádření detailu jednotlivých pojmů, je to podobné dekompozici.
- ✓ **Modelování v UML** – v podstatě existuje jeden model, který se skládá z různých pohledů navzájem propojených.
- ✓ **Metodika využití UML** není definována ve standardu, UML není metoda.
- ✓ **Vývoj** – většinou se začíná u Use Case Diagramu (model použití systému), každý scénář rozkreslíme do Sequence Diagramu nebo Collaboration Digramu. Jako pomocný si můžeme vytvořit Class Diagram, jehož

části lze rozkreslit do Statechart Diagramu. Use Case Diagram a Class Digram propojuje Activity Diagram. Atd.

Use Case Diagram:
- ✓ Využívá aktory (panáčci) a subsystém (obdélník), jehož části jsou use case, čáry představují vyjádření vazeb.
- ✓ Předdefinovaný standard říká, že vazby jsou dvě – input (skládání) a ex…

Class Diagram:
- ✓ Nutno rozlišovat mezi třídou a instancí. Texty ve třídě nejsou podtržené a u atributů se může psát dvojtečka, datový typ. U instancí by se měl název objektu podtrhnout, jedná se objekt, který je instancí nějaké třídy. Jsou-li konkrétní, je možné napsat jméno=hodnota.
- ✓ Nepodtrhávat atributy, že se jedná o primární klíč, ten se vyznačuje stereotypem, závorkami <<text>>.
- ✓ Interfaces – rozhraní, obdélníček s „tykadly“, „anténkami“.
- ✓ Packages – balíček, lze ho dekomponovat na další diagram.
- ✓ Associations – asociace - malý trojúhelníček u textu znamená směr čtení, ne směr vazby. Vazbu lze pojmenovat 2x, ve směru tam i zpět. Je-li vazba vícenásobná, je mezi vazbami kosočtverec.
- ✓ Qualifiers – kvalifikátor, říká, jak objekt vidí druhý objekt.
- ✓ Aggregation & Composition
- ✓ Generalization – generalizace, šipky jsou od podtypu k nadtypu, v angličtině jde šipka vždy směrem, jakým jde věta – něco obsahuje něco.

Sequence Diagram & Collabration Digram:
- ✓ Jedná se o stejný diagram, ale trojrozměrný. Jsou to dva různé pohledy, při kreslení jednoho automaticky doplňujeme druhý.
- ✓ Sequence Diagram – objekt v různé výšce vzniká a křížkem končí, během činnosti může nebo nemusí být aktivní, pokud není aktivní, značí se přerušovanou čárou. Pokud si jeden objekt s druhým „povídá“, spojuje se horizontálními šipkami. Lze stanovit časové rozpětí, opakování, podmínky, výhybky atd. Objekty zhora dolů jsou většinou konkrétní, stejné věci ve různých diagramech se malují stejně.
- ✓ Collaboration Diagram – značí skládání, dědění atd., což by v Sequence Diagramu nešlo kvůli křížení, je nutné se vzdát času.

Statechart Diagram:
- ✓ Perzistence – kolečko s písmenkem, H
- ✓ Paralelismus
- ✓ Vnořování uvnitř vztahu je automat, dekompozice.

Activity Diagram:
- ✓ Zdokonalený vývojový diagram. Lze ho kreslit klasicky i objektově.

Component Diagram:
- ✓ Komponenty vyjadřují, co je kde nainstalované.

Deployment Diagram:
- ✓ Navíc „krabičky“, které značí fyzické rozložení.

OCL:

OCL – Object Constraing Language:
- ✓ Programovací jazyk UML, vychází ze Smalltalku.
- ✓ Lze pomocí něj popsat všechny nakreslené diagramy. Umí víc, než jednotlivé diagramy.
- ✓ Diagram používáme jako datovou strukturu jazyka.