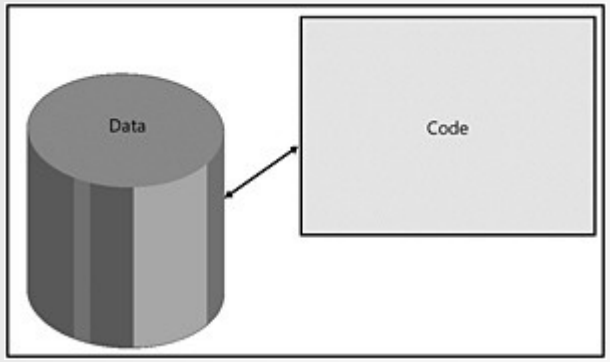
Design Patterns :: MVC

Agenda:

- What are design patterns?
- MVC to keep application cohesive
- How to create models in Java for enterprise apps
- Laboratory creating Java models (Java Beans)

Design Patterns :: MVC

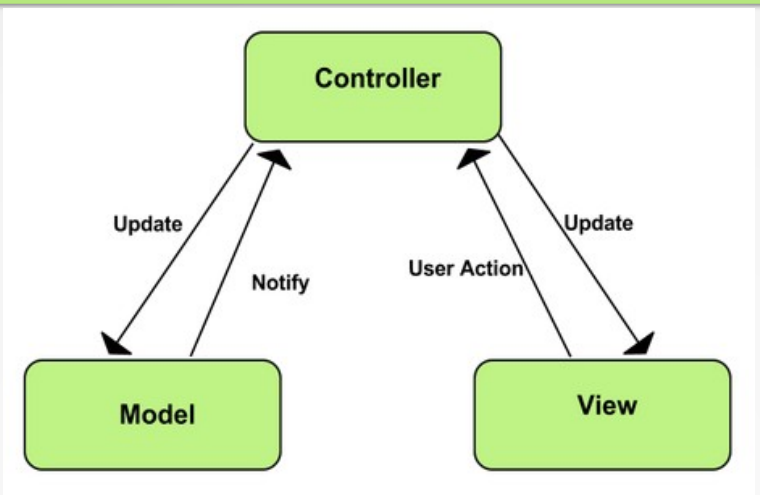


Monolithic application *

- What are design patterns
  - Solving common object oriented problems
  - Foundation on GRASP patterns
    - General Responsibility Assignment Software Patterns
    - Improve objects cohesion
    - Avoid monolithic applications
  - Monolithic Programming samples
    - Visual Basic
    - Delphi

* Source: Guerreiro, F., SQL Server 2005: Database Essentials, Microsoft Press(2007)

Design Patterns :: MVC

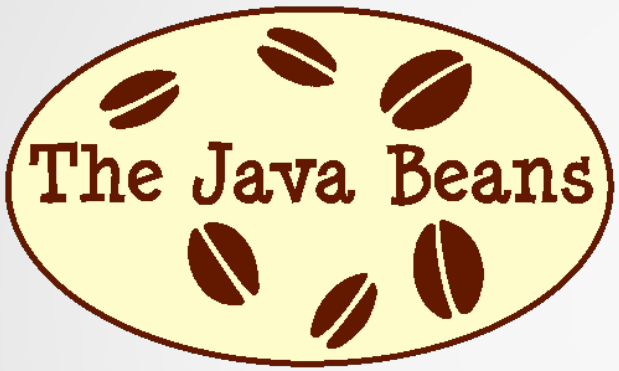


MVC model*

- MVC to keep application cohesive
  - Created in 1970s using Smalltalk
  - Divided application in cohesive blocks (MVC)
  - It allowed software development workforce division
  - Increase software reuse
  - Java Web application example
    - View: HTML; Controller: Servlet,REST service; Model: POJOs
  - Frameworks: Struts, SpringMVC, AngularJS, etc.

*Source: Chorome Apps in https://developer.chrome.com/apps/about_apps, 2017

Design Patterns :: MVC



- How to create models in Java for enterprise apps
  - POJOs are pure data structures
  - Normally implements java.io.Serializable interface
    - To transfer data between software layers
  - POJO are database mapped (JPA maps)
  - Data structure for business processes
  - Model objects need to have an ID
  - What need to be done in Javabeans:
    - Create private properties;
    - Public accessors methods: getter/setters prefixes (is for boolean)
    - Override equals and hashCode from Object class
    - Override toString method from Object class

Design Patterns :: MVC

- Laboratory : Creating Java Model
  - I want a software to control my orders, it must allow customers to include multiple products in a cart. In the end of the pushase the products must be shown in a form with totalization. There will be a payment of purchase available and must have many acceptable methods (cash; credit card; debit card, etc). The order must have many states: opened, in process, delivered, and cancelled and a report will show them by state, including summarizing them.

    1. Create an UML model (Astah)
    2. Create POJOs to support the requirements above
    3. Create unit tests (JUnit) to simulate a purchase with your data structure