

Course code: **SPRING/JSF**

Course title: **Modern Web applications based on Spring Framework, JSF, JPA and AJAX**

Days: 5

Description:

Course intended for:

The training is intended for Java programmers wanting to find out how to develop secure Web applications on the basis of Spring Framework and a data access layer, implemented via Hibernate.

Course objective:

The training objective is to get familiar with the rules of development of Web applications on the basis of Spring Framework along with such issues as: object-relational mapping and development of the layer of access to data using JPA, creation of the layer of services using the Spring IoC container, Web frontend software thanks to JSF.

In particular, the participants are to:

- Get familiar with the Dependency Injection pattern and learn to use it in practice using the IoC Spring Framework container
- Find out how to configure the components from the level of XML Spring Framework descriptors and using Java annotations
- Get familiar with the object-relational mapping and learn to develop the layer of persistence based on JPA, starting from simple entities, to complex issues, such as inheritance
- Find out how to read and save entities, ensure transactionability of operations
- Learn how to ensure data coherence and application performance
- Learn to build rich user interfaces based on Java Server Faces using JSF and facelets
- learn to use the ready JSF library components and develop their own

- learn how to increase comfort of use of the user interface thanks to AJAX
- learn about the advanced issues, such as AOP
- learn how to test these components
- get familiar with the issues of security and reliability of Web-based applications
- learn how to launch the applications developed on Tomcat and Jetty servers

Course strengths:

The curriculum includes a comprehensive and detailed overview of Web application development. Unlike most trainings, it is not focused on a single technology, but it encompasses the key issues from the selected technologies. Thanks to this, the training participants, after the training, will be able to develop full Web applications on their own, developing their knowledge further individually as needed. The curriculum is being constantly updated to take into account the emerging trends.

Requirements:

The training participants are required to have the Java programming skill (to be learned at the course J/JP), to have the basic knowledge of relational databases and SQL, and the basic knowledge of HTML. They should also be able to develop Web applications using Servlet/JSP technologies (to be learned at the course J/WEB1).

Course parameters:

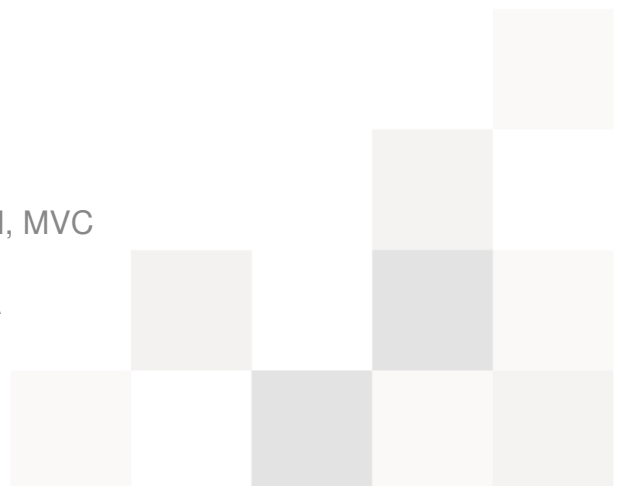
5*8 hours (5*7 net hours) of lectures and workshops (with a visible emphasis on workshops). During the workshops, apart from simple exercises, a simple complex application is designed and implemented.

Group size: no more than 8-10 participants.

Course curriculum:

1. Introduction

- Layer architecture of IT systems
- Architectural design patterns: IoC, DI, MVC
- Data access layer: JDBC, DAO, JPA



- Web layer: HTML, AJAX
- System integration issues
- Aspect oriented programming: AOP
- Issues associated with testing and security

2. Object container – Spring IoC

- Application component architecture
- Interface oriented programming
- Inversion of Control and Dependency Injection patterns
- Container initialization
- Defining and initialization of components
 - Constructors, factories
 - Simple properties
 - Complex properties, collections
 - Lazy initialization
- The range of components
 - singleton, prototype
 - own ranges
- Correlations between components
- Component life cycle
- Broadening of container functionality
- Event-oriented programming
- Data conversion in Spring
 - Property editors



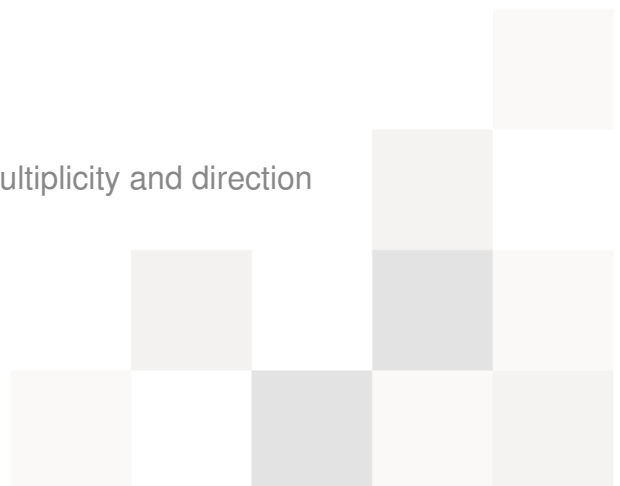
- Type conversion
- Formatting
- Specification of limitations
- Spring Expression Language
 - A review of possibilities
 - Syntax
 - Use

3. The basics of Spring AOP

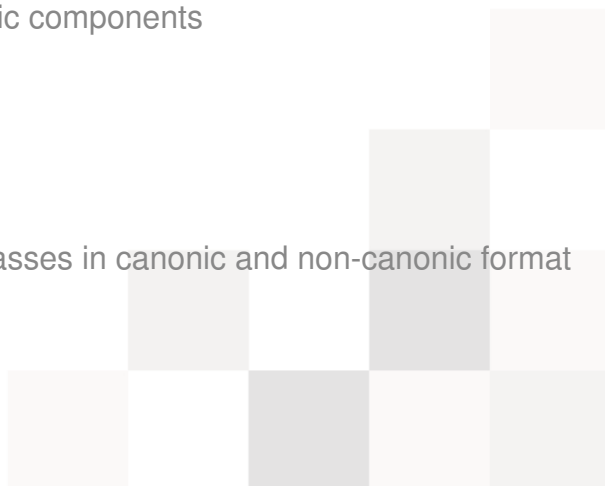
- The concept of proxy
- Basic terminology: aspect, join point, advice, pointcut
- Configuration
- Use in a Spring container
- Use in own containers
- Integration with AspectJ

4. Java Persistence API2.1

- Object-relational mapping concept
- Entity classes
 - Simple classes
 - Embedded classes
 - Fields and access to fields
 - Fields being collections
- Associations – taking into account multiplicity and direction
- Keys



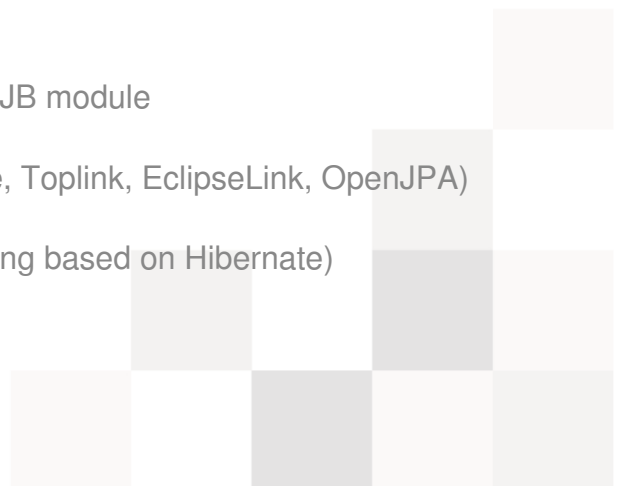
- Simple
- Complex
- Automatic key generation
- Object naming in database
 - Default mapping
 - Default mapping change
- Context of persistency (PersistentContext):
 - Container managed and application managed
 - Transaction and Extended
- Entity life cycle
- Operations on data
- Transactions
- queries
 - Development of queries in JPQL
 - Syntax
 - Query and TypedQuery
 - Named queries
 - Native queries
 - JPA2.1 Criteria API
 - Idea Criteria API, basic components
 - MetaModel
 - Concept
 - MetaModel classes in canonic and non-canonic format



- MetaModel API
 - Development of strongly typed queries
 - Development of weakly typed, dynamic queries
 - Query edition
 - Alternative approaches: LIQUidFORM
- Typical scenarios, including search-within and search-by-example
- Native queries, mapping of results
- Methods of life cycle listening
- Validation
 - Bean Validation 1.1
 - The concept of Bean Validation
 - Defining and imposing of limitations on types, fields, methods
 - Parameterization of limitations
 - In-built limitations
 - Composing of complex limitations
 - Development of validators
 - Validation process
 - Fields, objects and object graphs
 - Groups and sequences
 - Blocking of access to field and cascade access
 - API and exceptions
 - Adjustment of content of validation errors
- JPA2.1 a Bean Validation



- Configuration of validation
- The mode of functioning of Bean Validation in the context of JPA
- Validating limitations and generating of database objects
- Inheritance models and implementation
- Data cohesion issues
 - Optimistic locking
 - Pessimistic locking
 - Extensions in JPA2.1
- Cache of the 2nd level
 - The concept of the 2nd level
 - Cache configuration and work modes
 - Entity configuration
 - Cache – base interaction
 - Cache based on Hibernate (for a Hibernate based training)
 - Cache based on JPA2.1 mechanisms
 - threats
- Testing of components using JPA
- Configuration and deployment
 - As a Java SE/desktop application
 - As a Web module
 - Within the framework of an EJB module
- JPA and supplier libraries (Hibernate, Toplink, EclipseLink, OpenJPA)
- Hibernate-specific issues (for a training based on Hibernate)



- Hibernate API
 - Selected additional options
 - Hibernate subprojects: Search, Shards, Validator, Envers
 - Project patterns for the data access layer
5. Integration of the Spring container with the persistency layer
- DAO layer
 - Classical support for DAO layer development
 - Spring Data – a new approach to implementation of the persistency layer in the Spring environment
 - Spring Data JPA – a specialized version for JPA
 - Transactions
 - Transaction isolation levels
 - A transaction subsystem in Spring
 - Declarative transaction specification
 - Pessimistic and optimistic blocking
 - Long conversations
 - Unified management of exceptions
6. Web interface
- Introduction to JSF
 - Typical problems of a Web application
 - Demand processing cycle
 - Demand types
 - Cycle phases



- Full and partial processing
- A component-based interface model
- Main classes and interfaces
- Logic beans management model
 - Concept and type of ranges
 - Configuration
 - „Custom" ranges
- Expression Language
- Defining of view
 - A review of standard components available
 - HTML library
 - Core library
 - View Declaration Language
 - JSF based on facelets
 - Advantages of the technology
 - Templating mechanisms
 - In-built tags
 - Development of composite components
 - History - JSF based on JSP
 - JSF and REST
 - Support for HTML5
- Conversion models
 - In-built containers



- Development of containers
- Validation model
 - In-built validators
 - Creation of validators
 - A default validator
 - Bean Validation 1.1 and JSF
 - Events associated with validation
 - Implementation of non-typical/ complex validation scenarios
- The event-based model
 - Events and listeners – the basis for the event-oriented model
 - Action and change of value events
 - Phase change events
 - System events
- Navigation models
 - „Classical" navigation
 - Short navigation
 - Conditional navigation
 - Dynamic configuration of navigation
 - Pull and push MVC
- Main components of the JSF application and internationalization
- Typical problems encountered when developing the interface
 - The form as a table
 - „Intelligent" pagination



- Master-Detail, LOV
- Error management
- Application resources
 - Resource Request and its handling
 - Dependency on resources
 - EL objects and tags
 - Packaging and internationalization of resources
- Enriching of components – Behavior
- Development of components „from the start“
 - Development of components
 - Developments of renderers
 - Development of tags
- Packaging and configuration of the application, project phase
- Testing of interface built on the basis of JSF (JSFUnit/Selenium)
- Securing of applications

7. Development of Web applications using AJAX and JSF technologies

- AJAX concept and communication models
- A review of approaches to AJAX application development (layer – JSF, client-server – GWT, ...)
- AJAX application development „from the start“ (XMLHttpRequest)
- JSON
 - Introduction
 - Java API for JSON Processing
 - JSON in RESTFull webservice



- Basics of Prototype and JQuery libraries in the context of DOM tree manipulation and AJAX communication
- „Native" JSF2 approach
 - Tag f:ajax and Ajax Behavior
 - Ajax type request processing
 - Partial processing
 - Selective actions on the tree of components
 - Queuing
 - Management of errors
- Popular libraries working with JSF
 - PrimeFaces
 - OmniFaces
 - RichFaces
 - PrettyFaces
 - IceFaces

8. Web layer – navigation using Spring WebFlow

- Configuration
- Defining of flows
- WebFlow Expression Language
- View rendering
- Actions
- Persistent flows
- Flow security
- Flow inheritance



- Integration with JSF
- Flow testing

9. Introduction to Spring Security

- Introduction – a review of possibilities
- Configuration
- Web layer security
 - Authentication types
 - Additional options
 - HTTP/HTTPS channel confidentiality
 - Session management
 - Support for external authentication mechanisms
 - Defining of own filters

10. Securing of Sprint container component methods

