INFORMAČNÝ LIST PREDMETU

Vysoká škola: Univerzita Komenského v Bratislave

Fakulta: Fakulta matematiky, fyziky a informatiky

Kód:

Názov predmetu: Architectures of Software Systems

Web stránka predmetu:

Druh, rozsah a metóda vzdelávacích činností: 2 (prednáška)

Počet kreditov: 3

Odporúčaný semester/trimester štúdia:
aplikovaná informatika
aplikovaná informatika (konverzný program)
informatika
počítačová grafika a geometria
počítačová grafika a geometria (konverzný program)
1-2/L

Stupeň štúdia: magisterský

Podmieňujúce predmety:

Podmienky absolvovania predmetu:
Priebežné hodnotenie: midterm
Skúška: písomná alebo ústna skúška
Orientačná stupnica hodnotenia: A 90%, B 80%, C 70%, D 60%, E 50%
Váha priebežného / záverečného hodnotenia: 40/60

Výsledky vzdelávania:
Deeper knowledge of software engineering, architectural styles and patterns, design patterns, creating models and modeling the architectures. After this course students will be able to study and implement styles and patterns, use prefactoring and refactoring to optimize software design.

Stručná osnova predmetu:

1. Architectural styles I. (Garlan & Shaw: Abstract Machine, Pipes and Filters, Client-Server, Object Model, Repository, Blackboard)
2. Architectural styles II. (Interpreter, Modern Canonical Compiler, Rule-Based System, Aspect Oriented Architectures, MVC, Mickrokernel)
3. Distributed architectures, CORBA, Service Oriented Architectures.
4. Architectural patterns I (Buschmann: POSA IV: Pattern Oriented Software Architecture for Distributed Computing).
5. Architectural patterns II. (Reactor, Proactor, Requestor, Invoker, Acceptor, Connector, ACT, Facade, Master-Slave, ...)
6. Architectural patterns III (Memento, Context Object, DTO, Adapter, Iterator, Interceptor, ...)
7. Design patterns I (model and source code level, Gamma et al.).
   Creational Patterns (Builder, Abstract Factory, Factory method, ...).
8. Design patterns II. Structural Patterns (Bridge, Decorator, Composite, Proxy, ...)

9. Design patterns III. Behavioral Patterns. (Command, Mediator, State, Strategy, Visitor, Observer, ...)
10. From Refactoring to Patterns (Kerievsky).
11. Refactoring and Prefactoring (Fowler, Pugh).
12. UML and its new features. Superstructure, Infrastructure, meta-models, and Object Constraint Language. Consistency and interconnection of models. XMI, HUTN and PlantUML. 3DUML and xDUML.
13. Agile Modeling and development process. Lean Architecture (Coplien).

Odporúčaná literatúra:

1. Buschmann F. et al.: Pattern-oriented software architecture: a pattern language for distributed computing, Vol. 4. New York : John Wiley & Sons, 2007.
2. Shaw M L., Garlan D.: Software architecture: Perspectives on an emerging discipline. Prentice Hall, 1996.
3. Arlow J., Neustadt I. UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design. Addison-Wesley, 2006.
4. Kerievsky J.: Refactoring to Patterns. Addison Wesley, 2008.
5. Gamma E. et al.: Design Patterns. Elements of Reusable Object-Oriented Software. Addison Wesley, 1994.
6. Fowler M.: Refactoring. Improving the Design of Existing Code. Wesley Longmann, 2000.
7. Pugh K.: Prefactoring, O'Reilly, 2005
8. Coplien O. J., Bjornvig G.: Lean Architecture for Agile Software Development. J. Wiley, 2014.

Jazyk, ktorého znalosť je potrebná na absolvovanie predmetu: slovenský, anglický

Poznámky:

Vyučujúci:
doc. Ing. Ivan Polášek, PhD.

Dátum poslednej zmeny: 17.8.2020

Schválil: prof. RNDr. Roman Ďurikovič, PhD.