

Software modelling techniques - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Software modelling techniques
Kod przedmiotu	11.3-WE-INF-D-SoftModellTechn-Er
Wydział	Wydział Informatyki, Elektrotechniki i Automatyki
Kierunek	WIEiA - oferta ERASMUS / Informatyka
Profil	-
Rodzaj studiów	Program Erasmus drugiego stopnia
Semestr rozpoczęcia	semestr zimowy 2018/2019

Informacje o przedmiocie	
Semestr	1
Liczba punktów ECTS do zdobycia	5
Typ przedmiotu	obowiązkowy
Język nauczania	angielski
Sylabus opracował	- dr inż. Grzegorz Bazydło - dr inż. Tomasz Gratkowski

Formy zajęć					
Forma zajęć	Liczba godzin w semestrze (stacjonarne)	Liczba godzin w tygodniu (stacjonarne)	Liczba godzin w semestrze (niestacjonarne)	Liczba godzin w tygodniu (niestacjonarne)	Forma zaliczenia
Wykład	30	2	-	-	Zaliczenie na ocenę
Laboratorium	30	2	-	-	Zaliczenie na ocenę

Cel przedmiotu

- Familiarize students with bases of software engineering and programs modelling techniques.
- Shaping skills in business process modelling.
- Familiarize students with object modelling principles.
- Shaping skills in UML modeling.

Wymagania wstępne

Object-oriented programming

Zakres tematyczny

- Elements of software engineering. Software development. The Software Crisis and countermeasures.
- Conceptual modeling. The role of modeling in software design. Historical illustration of modern modeling techniques.
- Business analysis. Business process modeling in BPMN notation. Business use cases. Modeling software based on the BPMN model.
- Structural and object-oriented methods. Agile methods. Analysis and modeling of requirements. Analysis and modeling of the system domain. Architecture design solution. Software life cycle.
- Unified Modeling Language. Origin, definition, and goals of UML. UML scope. UML diagrams. UML extensions: profiles, stereotypes, labels, OCL.
- Fundamentals of object-oriented design, interrelations between objects. Modeling object relationships. Classes, inheritance, generalization, specialization, polymorphism, interfaces.
- Modeling the user interface.

Metody kształcenia

Lecture: conventional lecture.

Laboratory: laboratory exercises.

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Student knows the basics of object-oriented programming and can design programs using an object-oriented paradigm.		bieżąca kontrola na zajęciach	Laboratorium
Student can model the software using the appropriate modelling languages.		bieżąca kontrola na zajęciach	Laboratorium
Student knows the basics of UML, the most important types of UML diagrams and their use.		test	Wykład

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Student has the knowledge about languages and techniques of modelling software and business processes.		test	Wykład
Student understands the need for software modelling to facilitate its design and increase its credibility.		test	Wykład

Warunki zaliczenia

Lecture: the main condition to get a pass are sufficient marks for all written or oral tests conducted during the semester.

Laboratory: a condition of pass is to obtain positive grades from all laboratory exercises that are expected to be performed within the laboratory program.

Composition of the final grade: lecture: 50% + laboratory: 50%

Literatura podstawowa

1. Sommerville I.: Software Engineering (10th Edition), Pearson Education, 2016.
2. Rumbaugh J., Jacobson I., Booch G.: The Unified Modeling Language Reference Manual, Second Edition, Addison-Wesley, 1999.
3. Pilone D., Pitman N.: UML 2.0 in a Nutshell, A Desktop Quick Reference, O'Reilly Media, 2005.
4. Shapiro R., White S. A., Bock C., Palmer N. et al: BPMN 2.0 Handbook Second Edition, Future Strategies Inc., 2012.
5. Martin R. C.: Agile Software Development, Principles, Patterns, and Practices, Pearson Education, 2013.

Literatura uzupełniająca

1. Booch G., Rumbaugh J., Jacobson I.: The Unified Modeling Language User Guide, Second Edition, Addison-Wesley, 2005.
2. Brookes F. P.: The Mythical Man-Month, Anniversary Edition: Essays On Software Engineering, Addison-Wesley, 2010.
3. Osterwalder A., Pigneur Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers, John Wiley & Sons, 2010.
4. Rasmusson J.: The Agile Samurai: How Agile Masters Deliver Great Software, The Pragmatic Programmers LLC, 2010.

Uwagi

Zmodyfikowane przez dr inż. Grzegorz Bazydło (ostatnia modyfikacja: 28-03-2018 13:09)

Wygenerowano automatycznie z systemu SyllabUZ