

Large Scale Software Design - course description

General information	
Course name	Large Scale Software Design
Course ID	11.3-WK-IIeP-PSI-W-S14_pNadGenLP3ME
Faculty	Faculty of Mathematics, Computer Science and Econometrics
Field of study	Informatics and Econometrics
Education profile	academic
Level of studies	First-cycle studies leading to Bachelor's degree
Beginning semester	winter term 2018/2019

Course information	
Semester	5
ECTS credits to win	4
Course type	obligatory
Teaching language	polish
Author of syllabus	dr inż. Janusz Jabłoński

Classes forms					
The class form	Hours per semester (full-time)	Hours per week (full-time)	Hours per semester (part-time)	Hours per week (part-time)	Form of assignment
Lecture	30	2	-	-	Exam
Laboratory	30	2	-	-	Credit with grade

Aim of the course

Zapoznanie z teoretycznymi i praktycznymi aspektami analizy i projektowania systemów informatycznych z wykorzystaniem metodyk, technik i narzędzi wspomagających zarządzanie projektami.

Prerequisites

Programowanie komputerów, Technologie Informatyczne.

Scope

Wykład

1. Zakres materiału, literatura oraz omówienie formy i treści zaliczenia przedmiotu (2 godz.)
2. Metodologiczne podstawy tworzenia systemów informatycznych: rodzaje systemów, zakres i składniki metodyki tworzenia. Klasyfikacja metodyk tworzenia. (4 godz.)
3. Cykl życia systemu: kaskadowy, spiralny, przyrostowy. (2 godz.)
4. Fazy cyklu: planowanie, analiza, projektowanie, wdrażanie, użytkowanie systemów. (2 godz.)
5. Modyfikacje cyklu życia systemu: prototypowanie, pakiety zastosowań, metodyki „zwinne”. (2 godz.)
6. Planowanie systemów informatycznych: cele planowania, proces planowania, formułowanie strategii informatyzacji, studium wykonalności, metody analizy sytuacyjnej, zespół projektowy. (2 godz.)
7. Model klient – server oraz architektura trójwarstwowa w realizacji systemów informatycznych. (2 godz.)
8. Metody i techniki projektowania systemów informatycznych – modele związków encji, diagramy przepływu danych, słowniki/skorowidze danych, techniki decyzyjne, diagramy struktur. Diagramy UML w modelowaniu struktury i dynamiki systemów obiektowych. (4 godz.)
9. Metodyka obiektowa - techniki i modele UML w projektowaniu systemów informatycznych. (4 godz.)
10. Metodyki społeczne. (2 godz.)
11. Komputerowo wspomagane tworzenie systemów informatycznych (2 godz.)
12. Pakiety wspomagające zarządzanie projektami: istota i generacje, rodzaje pakietów na rynku oprogramowania. (2 godz.)

Laboratorium

1. Wprowadzenie, omówienie zakres materiału i formy zaliczenia laboratorium (2 godz.)
2. Etapy cyklu życia oprogramowania jako dokumentacja projektowa - przykłady (2 godz.)
3. Podstawowe funkcjonalności systemu informatycznego w praktyce gospodarczej (2 godz.)
4. Cykl życia systemu informatycznego - Diagram Ganta w MicrosoftProject (2 godz.)
5. Modelowanie systemu transakcyjnego "Sklep" - MSVisio w modelowaniu (2 godz.)
6. Projektowanie bazy danych dla zadanych wymagań - profil klienta, profil towaru (2 godz.)
7. Projektowanie interfejsu użytkownika i programowanie zdarzeniowe w MS Excel (2 godz.)
8. Implementacja funkcjonalności systemu "Sklep" w VisualBasic-u (2 godz.)
9. Modelowanie struktury systemu - Eclipse i projektowanie diagramu klas (2 godz.)
10. JDBC i bazy danych - połączenie z bazą danych i zapytania z poziomu Eclipse (2 godz.)
11. Serwer aplikacji i architektura trójwarstwowa w realizacji systemu transakcyjnego (6 godz.)
12. Testowanie i weryfikacja systemu "Sklep" (2 godz.)
13. Sprawdzenie wiadomości i wystawienie zaliczenia laboratorium (2 godz.)

Teaching methods

Wykład z elementami prezentacji multimedialnych, pogadanka, dyskusja, praca w grupach. Praca z pakietami: MSPProject, MSExcell, MSAcces, Eclipse. Programowanie w Java.

Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
Student zna podstawowe cykle życia oprogramowania oraz techniki metodyk strukturalnych i obiektowych w wytwarzaniu oprogramowania	• K_W09	• an exam - oral, descriptive, test and other • kolokwium. sprawdzanie stopnia przygotowania do ćwiczeń laboratoryjnych oraz sprawozdań z realizowanych laboratoriów	• Lecture • Laboratory
Student rozumie konieczność systematycznej pracy, także w zespole, nad projektami o długofalowym charakterze.	• K_K05	• an evaluation test	• Laboratory
Student posiada podstawowe umiejętności analityczne i projektowe niezbędne w kontaktach ze specjalistami z dziedziny proble-mu oraz branży IT, potrafi specyfikować wymagania wobec systemu informatycznego oraz potrafi korzystać z notacji obiektowych zgodnych z UML w zakresie niezbędnym do implementacji systemów informatycznych w architekturze trójwarstwowej, korzysta z oraz narzędzia CASE w realizacji prostych systemów informatycznych.	• K_U23 • K_U27 • K_U28 • K_U34	• an exam - oral, descriptive, test and other • kolokwium. sprawdzanie stopnia przygotowania do ćwiczeń laboratoryjnych oraz sprawozdań z realizowanych laboratoriów	• Lecture • Laboratory
Student potrafi współdziałać i pracować w zespole.	• K_K09	• a test with score scale • sprawdzanie sprawozdań z realizowanych laboratoriów	• Laboratory
Student umie dobrać cykl życia oprogramowania do zaistniałych wymagań i ograniczeń.	• K_U21	• an exam - oral, descriptive, test and other • kolokwium. sprawdzanie stopnia przygotowania do ćwiczeń laboratoryjnych oraz sprawozdań z realizowanych laboratoriów	• Lecture • Laboratory
Student posiada ogólną wiedzę na temat systemów informatycznych.	• K_W01	• an exam - oral, descriptive, test and other • kolokwium. sprawdzanie stopnia przygotowania do ćwiczeń laboratoryjnych oraz sprawozdań z realizowanych laboratoriów	• Lecture • Laboratory

Assignment conditions

1. Ocena stopnia przygotowania do ćwiczeń laboratoryjnych oraz sprawozdań z realizowanych laboratoriów.
2. Kolokwium pisemne z progami punktowymi oraz zadaniami pozwalającymi ocenić, czy student osiągnął efekty kształcenia w stopniu minimalnym.

3. Egzamin pisemny składający się z pytań testowych i zadań, weryfikujący znajomość modeli, technik i metod.

Warunkiem zaliczenia przedmiotu jest pozytywna ocena z laboratorium i z egzaminu. Ostateczna ocena z przedmiotu uwzględnia ocenę z laboratorium (60%) i ocenę z egzaminu (40%), przy założeniu, że student osiągnął wszystkie zakładane efekty kształcenia w stopniu dostatecznym.

Recommended reading

1. R. Barker, C. Longman, CASE Method, modelowanie funkcji i procesów, WNT, 1996.
2. V. Sthern, C++ Inżynieria Programowania, Helion, Gliwice 2004.
3. J. Cogswell, Tworzenie użytecznego oprogramowania, Warszawa 2005.
4. J. Górski, Inżynieria oprogramowania w projekcie informatycznym, Warszawa 2000.

Further reading

1. P. Benon-Davies, Inżynieria Systemów Informacyjnych, WNT, 1999.
2. J. Roszkowski, Analiza i projektowanie strukturalne, Wydawnictwo Helion, 1998.
3. C. L. Hall, Techniczne podstawy systemów klient–serwer, WNT, 1996.

Notes

Modified by dr Robert Dylewski, prof. UZ (last modification: 04-05-2018 19:24)

Generated automatically from SylabUZ computer system