



MODULIO APRAŠAS

Modulio pavadinimas	Kodas
Programų sistemų inžinerija	

Dėstytojas	Padalinys
Koordinuojantis: Saulius Ragaišis	Programų sistemų katedra Informatikos institutas Vilniaus universitetas
Kitas (-i): -	

Studijų pakopa	Dalyko tipas
Pirmoji	Privalomasis

Igyvendinimo forma	Vykdymo laikotarpis	Vykdymo kalbos
Auditorinė	7 semestras	Lietuvių, anglų

Reikalavimai studijuojančiajam
Išankstiniai reikalavimai: Programavimo pagrindai; Objektinis programavimas; Duomenų bazių valdymo sistemos; Kompiuterių tinklai.

Modulio apimtis kreditais	Visas studento darbo krūvis	Kontaktinio darbo valandos	Savarankiško darbo valandos
5	150	66	84

Modulio tikslas: studijų programos ugdomos kompetencijos		
Modulio tikslas: ugdyti pagrindines programų sistemų inžinerijos kompetencijas: projekto planavimą ir valdymą, reikalavimų analizę ir specifikavimą, programų sistemų projektavimą, verifikavimą ir validavimą. Taip pat siekiama ugdyti komandinio darbo, bendravimo ir viešo rezultatų pristatymo įgūdžius.		
Modulio studijų siekiniai	Studijų metodai	Vertinimo metodai
<ul style="list-style-type: none">- bendrauti programų sistemų kūrimo klausimais, viešai pristatyti rezultatus, atlikti darbus laiku,- parinkti tinkamą programinės įrangos gyvavimo ciklą ir parengti programų sistemos projekto planą,- surinkti reikalavimus programų sistemai, parengti reikalavimų specifikaciją,- projektuoti programų sistemas,- objektinės analizės ir projektavimo modelius išreikšti UML,- kuriamai programų sistemai parinkti tinkamus verifikavimo ir validavimo būdus,- suprasti programų kūrimo proceso vertinimo ir gerinimo principus,- taikyti tinkamus programų sistemų priežiūros metodus.	<p>Probleminis dėstymas, atvejų analizė, grupinės diskusijos, komandinis darbas</p> <p>Probleminis dėstymas, atvejų analizė, grupinės diskusijos</p>	<p>Egzaminavimas raštu, komandinės užduotys, darbų gynimas, nedidelės asmeninės užduotys</p> <p>Egzaminavimas raštu, nedidelės asmeninės užduotys</p>

Temos	Kontaktinio darbo valandos							Savarankiškų studijų laikas ir užduotys	
	Paskaitos	Konsultacijos	Seminarai	Pratybos	Laboratoriniai darbai (LD)	Praktika	Visas kontaktinis darbas	Savarankiškas darbas	Užduotys
Įvadas į programų sistemų inžineriją. Kurso struktūra ir atsiskaitymo reikalavimai	1						1	1	Individualus skaitymas, pirma komandinė užduotis, nedidelės asmeninės užduotys
Programinės įrangos gyvavimo ciklas ir proceso modeliai: programinės įrangos gyvavimo ciklo samprata; gyvavimo ciklo procesai ir kuriami darbo produktai; proceso modeliai: tradicinis nuoseklus, inkrementinis, spiralinis, judrusis, unifikuotas procesas; prototipavimas	2				1		3	4	
Programų sistemų projekto valdymas: komandos valdymas; tvarkaraščio sudarymas; programinės įrangos matavimo ir apimtys vertinimo metodai; rizikos analizė; kokybės užtikrinimas; konfigūracijos valdymas ir versijų kontrolė; projekto valdymo įrankiai	6				6		12	18	
Judriosios programų kūrimo metodikos: manifestas, vertybės ir principai; Ekstremalus programavimas (XP), Scrum, DSDM; judriųjų ir tradicinių metodikų derinimas	2				1		3	3	
Reikalavimų analizė ir specifikavimas: sistemos lygio aspektai; reikalavimų programų sistemai surinkimas; reikalavimų analizės modeliavimo metodai; funkciniai ir nefunkciniai reikalavimai; pagrindinės formalių specifikavimo būdų sąvokos; reikalavimų analizės ir projektavimo įrankiai	5				2		7	10	Individualus skaitymas, antra komandinė užduotis, nedidelės asmeninės užduotys
Objektiškai orientuota analizė: sistemos idėjos ir tikslo apibrėžimas; suinteresuotų asmenų, veiklos procesų ir naudojimo atvejų identifikavimas; reikalavimų apibrėžimas; veiklos klasių identifikavimas; naudojimo atvejų proceso modelio sudarymas; sistemos interfeisų apibrėžimas; interfeisų prototipavimas	3				4		7	10	
UML pagrindinės diagramos: naudojimo atvejų diagramos; klasių diagramos; elgsenos diagramos; įgyvendinimo diagramos	3				2		5	5	
Formalūs metodai: sąvokos, specifikavimas ir verifikavimas; formalios specifikavimo kalbos Z ir OCL; Cleanroom metodas	2						2	1	
Programų sistemų projektavimas: pagrindinės projektavimo sąvokos ir principai; kontraktų paskirtis ir naudojimas; projektavimo šablonai; programų sistemų architektūra; struktūrinis projektavimas; komponentų projektavimas; projektavimo kokybė; vidinė kokybė (žema sankiba, aukštas rišlumas, informacijos slėpimas, efektyvumas); išorinė kokybė (patikimumas, prižiūrimumas, naudojamumas, veikimas); kiti metodai: duomenimis grįstas, aspektinis, funkcinis, paslaugomis grįstas, judrusis; projektavimas pakartotiniam panaudojimui; atviro kodo resursų	5						5	5	Individualus skaitymas, nedidelės asmeninės užduotys

panaudojimas									
Objektiškai orientuotas projektavimas: taikomosios sistemos architektūros apibrėžimas; dalykinės srities komponentų identifikavimas; komponentų klasių modelio sukūrimas; komponentų interfeisų projektavimas; bendradarbiavimo modelių sukūrimas; komponentų testų sukūrimas; klasių testų sukūrimas; atributų apibrėžimas; dialogų specifikavimas	3						3	3	
Programų sistemų konstravimas: gerosios praktikos, kodavimo standartai, potencialios saugumo problemos	2						2	2	
Interneto paslaugos: apibrėžimas (WSDL), protokolai (SOAP, REST), paslaugų architektūra	2						2	2	
Programinės įrangos patikimumas: sąvokos, patikimumo modeliai, metrikos, gerinimo būdai	2						2	2	
Verifikavimas ir validavimas: verifikavimo ir validavimo skirtumai; statiniai ir dinaminiai metodai; validavimo planavimas ir dokumentavimas; skirtingi testavimo tipai: žmogaus kompiuterio interfeiso, naudojamumo, patikimumo, saugumo, atitikimo specifikacijai; testavimo pagrindai (testavimo plano sudarymas, testų kūrimas, juodos dėžės ir baltos dėžės testavimo metodai; defektų pasiskirstymas; modulių, integravimo, validavimo ir sistemos testavimas; objektiškai orientuotas testavimas; matavimai: proceso, projekto, programos; ne programinio kodo (dokumentacijos, pagalbos, mokymų medžiagos) verifikavimas ir validavimas; defektų registravimas, sekimas ir tam naudojami įrankiai; regresinis testavimas; inspekcijos, peržiūros, auditi	6						6	4	Individualus skaitymas, nedidelės asmeninės užduotys
Programų kūrimo procesas: programų kūrimo proceso gebėjimo brandos modeliai; asmeninis programų kūrimo procesas (PSP), komandinis programų kūrimo procesas (TSP); programų kūrimo proceso gerinimo principai; proceso vertinimo modeliai; programų kūrimo proceso matavimas	2						2	2	
Programų sistemų evoliucija: programų sistemų priežiūra; nesudėtingai prižiūrimos programų sistemos charakteristikos; reinžinerija; „pasenusios“ sistemos; kodo pertvarkos; pakartotinis panaudojimas	2						2	2	
Pasiruošimas egzaminui, egzaminas raštu		2					2	10	
Iš viso	48	2			16		66	84	

Vertinimo strategija	Svoris proc.	Atsiskaitymo laikas	Vertinimo kriterijai
Nedidelės asmeninės užduotys	50	Semestro metu	Paskaitų ir laboratorinių darbų metu studentams bus siūlomos nedidelės asmeninės užduotys. Balai skiriami už teisingus ir greitus užduočių sprendimus.
Komandinės užduotys		8-a ir 13-a semestro savaitė	2 komandinės užduotys: - projekto planas, - reikalavimų specifikacija. Vertinant atsižvelgiama į teisingumą, atitikimą reikalavimams, skaitomumą, pateikimą darbo laiku ir viešą darbo gynimą.
Egzaminas raštu	50	Egzaminų sesijos metu	Egzaminą leidžiama laikyti tik atlikus 2 komandines užduotis. Egzaminas sudarytas iš teorinių klausimų ir praktinių užduočių (skirtingo sudėtingumo). Studento žinios ir gebėjimai vertinami taip: 5 – puikios žinios ir gebėjimai;

			4 – labai geros žinios ir gebėjimai; 3 – geros žinios ir gebėjimai; 2 – patenkinamos žinios ir gebėjimai; 1 – žinios ir gebėjimai neatitinka minimalių reikalavimų; 0 – jokių žinių ir gebėjimų.
--	--	--	--

Reikalavimai dalyko vertinimui eksterno būdu	
Įvertinimas galimas eksterno būdu:	Taip
Studentas turi būti anksčiau įvykdęs reikalavimus laikyti egzaminą. Įskaitomi ankstesni darbo semestro metu įvertinimai.	

Autorius	Leidi mo metai	Pavadinimas	Periodinio leidinio Nr. ar leidinio tomas	Leidimo vieta ir leidykla ar internetinė nuoroda
Privalomoji literatūra				
Saulius Ragaišis	2012	Programų sistemų inžinerija. Paskaitų skaidrės		http://www.mif.vu.lt/~ragaisis/PSI_inf2012/
Rogers S. Pressman	2005	Software engineering: a practitioner's approach (6th edition)	ISBN 0071238409	McGraw Hill Higher Education
Bernd Oestereich	2002	Developing Software with UML: Object-Oriented Analysis and Design in Practice (2nd edition)	ISBN 020175603X	Addison-Wesley Professional
Papildoma literatūra				
Ian Sommerville	2007	Software engineering (8th edition)	ISBN 0321313798	Addison-Wesley
Craig Larman	2010	Applying UML and patterns: an introduction to object- oriented analysis and design and iterative development	ISBN 0131489062	Prentice Hall
Alistair Cockburn	2006	Writing effective use cases	ISBN 0201702258	Addison-Wesley