

RTU studiju kurss "Programmatūras izstrādes tehnoloģija"**33000 Datorzinātnes, informācijas tehnoloģijas un enerģētikas fakultāte****Vispārējā informācija**

Kods	DIP383
Nosaukums	Programmatūras izstrādes tehnoloģija
Studiju kursa statuss programmā	Obligātais/Ierobežotās izvēles
Atbildīgais mācītbspēks	Natālija Prokofjeva - Doktors, Vadošais pētnieks
Mācītbspēks	Sabina Kataļņikova - Asistents Oksana Zavjalova - Lektors Jānis Amoliņš - Lektors Inese Simkeviča - Augstskolas darbinieks ar uzņēmuma līgumiem Ingus Terbets - Pasniedzējs
Apjoms daļās un kredītpunktos	1 daļa, 3.0 kredītpunkti
Studiju kursa īstenošanas valodas	LV, EN
Anotācija	Programmatūras dzīves cikls. Programmatūras izstrādes paradigmas. Prasību analīze un definīcijas. Programmatūras specifikācija. Programmatūras projektēšana. Arhitektūras un komponentu projektēšanas metodes. Projektēšanas kvalitātes novērtējums. Lietotāju interfeiss un tā novērtēšana. Verifikācija un validācija. Testēšanas mērķis un metodes. Testēšanas process: moduļu testēšana, sistēmu testēšana, testēšanas stratēģijas. Programmatūras uzturēšana.
Mērķis un uzdevumi, izteikti kompetencēs un prasmēs	Priekšmeta mērķis ir iepazīstināt studentus ar programminženierijas modeļiem un metodēm, kā arī iemācīt viņiem izstrādāt un dokumentēt programmatūras sistēmas, lietojot tehnoloģiskās zināšanas un metodes. Priekšmeta uzdevumi: 1) aplūkot programmatūras dzīves ciklu, analizējot katra posma mērķus un uzdevumus; 2) izanalizēt programmatūras izstrādes modeļus: kādām programmatūras klasēm ir lietderīgs, kādas tehnikas un kā tiek lietotas, pozitīvās īpašības un trūkumi; 3) iemācīt studentiem praktiski lietot iegūtās zināšanas uzdotās programmatūras sistēmas izstrādē.
Patstāvīgais darbs, tā organizācija un uzdevumi	Laboratorijas darbu izpilde: 1) programmatūras specifikēšana; 2) programmatūras projektēšana; 3) programmatūras implementēšana; 4) programmas testēšana (uzdevums katram laboratorijas darbam ir izklāstīts atsevišķā dokumentā).
Literatūra	Obligātā/Obligatory: 1. Harper R. Practical Foundations for Programming Languages – 2nd Edition – Cambridge University Press, 2016. – 512 p. 2. Martin R. Clean Architecture: A Craftman's Guide to Software Structure and Design – 1st Edition – Pearson Education (US), 2017. – 432 p. 3. Bierig R., Brown S., Galvan E., Timoney J. Essentials of Software Testing – 1st Edition – Cambridge University Press, 2021. – 318 p. 4. Kodors S. Ievads prasību inženierijā. – 1. izd. – Rēzekne: Rēzeknes Tehnoloģiju akadēmija, 2019. – 68 lpp. 5. Ammann P., Offutt J. Introduction to Software Testing – 2nd Edition – Cambridge University Press, 2016. – 364 p. 6. Kneuper R. Software Processes and Life Cycle Models – 1st Edition – Springer, 2018 – 365 p. Papildu/Additional: 1. Zaiceva L. Programmatūras izstrādes tehnoloģija. – 2. izd. – Rīga : RTU, 2002. – 244 lpp. 2. Sommerville I. Software Engineering. – 9th Edition. – Addison-Wesley Pub. Comp, 2010. – 792 p. 3. Van Vliet H. Software Engineering. Principles and Practice. – John Wiley & Sons, 2008. – 740 p. 4. Pfleeger S.L., Atlee J.M. Software Engineering: Theory and Practice. – 4th Edition. – PRENTICE HALL, 2010. – 792 p. 5. Bell D. Software Engineering for Students. – Pearson Education (US), 2005. – 448 p. 6. McCaffrey J.D. Software Testing: Fundamental Principles and Essential Knowledge. – BookSurge Publishing, 2009. – 118 p. 7. Гагарина Л.Г., Кокорева Е.И., Виснадул Б.Д. Технология разработки программного обеспечения : учебное пособие. – М.: ИД «ФОРУМ»: ИНФРА-М, 2008. – 400 с.
Nepieciešamās priekšzināšanas	Programmēšanas valodas (Java, C++, PHP u.c.), datubāzu pārvaldības sistēmas

Studiju kursa saturs

Saturs	Pilna un nepilna laika klātienes studijas		Nepilna laika neklātienes studijas	
	Kontakt stundas	Patstāv. darbs	Kontakt stundas	Patstāv. darbs
Programminženierijas jēdziens un klasifikācija. Programmatūras dzīves cikls. Programmatūras izstrādes procesa modeļi.	2	2	0	0
Prasību definēšana, specifikēšana un dokumentēšana. Funkcionālās un nefunkcionālās prasības.	4	4	0	0
Programmatūras projektēšanas process, posmi un metodes.	4	4	0	0
Datu struktūru un lietotāja interfeisa projektēšana. Projektējuma kvalitātes kritēriji. Lietotāja interfeisa novērtēšana.	4	4	0	0
Programmatūras izstrādes principi strādājot komandā. Versiju pārvaldība.	4	4	0	0

Programmatūras implementēšanas metodes. Programmu atklādošana.	4	4	0	0
Programmatūras izstrādes procesa automatizācija. Nepārtraukta integrācija.	2	2	0	0
Programmatūras izstrādes procesa riski un to pārvaldība.	2	2	0	0
Verifikācijas un validācijas procesa mērķi un tehnikas. Statiskā verifikācija. Melnās kastes testēšanas metodes.	4	4	0	0
Baltās kastes testēšanas metodes. Programmatūras testēšanas dokumentācija.	4	4	0	0
Programmatūras uzturēšanas process. Programmētāja personiskās īpašības.	2	2	0	0
Kopā:	36	36	0	0

Sasniedzamie studiju rezultāti un to vērtēšana

Sasniedzamie studiju rezultāti	Rezultātu vērtēšanas metodes
Pārzina programmatūras izstrādes posmus un modeļus, darbus, kas jāizpilda katrā posmā, posmu nodevumus un dokumentus.	Eksāmens
Spēj sadarboties grupā, veicot dažādus ar programmatūras izstrādes procesu saistītus uzdevumus.	Grupu darbi
Spēj veidot sistēmas modeli, definēt un specificēt prasības, sagatavot prasību dokumentu, izstrādāt projektējumu, izstrādāt un implementēt sistēmu, izmantojot situācijai piemērotu izstrādes tehnoloģiju, kā arī veikt sistēmas testēšanu,	Laboratorijas darbi

Studiju rezultātu vērtēšanas kritēriji

Kritērijs	% no kopējā vērtējuma
Eksāmens	30
Grupu darbi	10
Laboratorijas darbi (programmatūra + dokumentācija)	60
Kopā:	100

Studiju kursa plānojums

Daļa	KP	Stundas			Pārbaudījumi		
		Lekcijas	Prakt d.	Laborat	Ieskaite	Eksām.	Darbs
1.	3.0	1.0	0.0	1.0		*	