

RTU studiju kurss "Vairākpavedienošana programmatūrā"**33000 Datorzinātnes, informācijas tehnoloģijas un enerģētikas fakultāte****Vispārējā informācija**

Kods	DE0280
Nosaukums	Vairākpavedienošana programmatūrā
Studiju kursa statuss programmā	Obligātais/Ierobežotās izvēles; Brīvās izvēles
Atbildīgais mācītbspēks	Pāvels Rusakovs - Doktors, Docents
Apjoms daļās un kredītpunktos	1 daļa, 3.0 kredītpunkti
Studiju kursa īstenošanas valodas	LV, EN
Anotācija	Studiju kursā tiek aplūkota vairākpavedienošanas iespēju lietošana modernajās programmēšanas valodās. Studiju kursā sniegts apskats par pavedieniem, paralēliem un laiksakritīgiem apakšprocesiem (aģentiem). Aplūkoti vairākpavedienošanas teorētiskie pamati. Programmēšanas līmenī detalizēti izskatīta pavedienu veidošana un pavedienu savstarpējā iedarbība, datu aizsardzība no konkurējošiem pavedieniem, izņēmumu apstrāde potenciāli bīstamās situācijās, semafori un monitori. Analizēti ar vairākpavedienošānu saistīti programmēšanas valodu paplašinājumi un papildu bibliotēkas. Laboratorijas darbos studentiem ir nodrošināta iespēja pilnveidot studiju kursā apgūtās teorētiskās zināšanas un programmēšanas prasmes.
Mērķis un uzdevumi, izteikti kompetencēs un prasmēs	Studiju kursa mērķis ir sagatavot studentu paralēlajai un laiksakritīgajai programmēšanai modernajās programmēšanas valodās. Studiju kursa uzdevumi: 1. Veicināt izpratni par vairākpavedienošanas rašanās iemesliem un tās pamatprincipiem; 2. Attīstīt prasmi lietot programmēšanas valodas, kuras intensīvi izmanto reāllaika sistēmās; 3. Apgūt programmēšanu platformās ar vairākpavedienošanas atbalstu; 4. Attīstīt pavedienu veidošanas un vadības prasmes; 5. Attīstīt resursu aizsardzības problēmu risināšanas prasmes.
Patstāvīgais darbs, tā organizācija un uzdevumi	Patstāvīgais darbs tiek organizēts, studentiem individuāli izpildot uzdevumus par vairākpavedienošanas izmantošanu dažādās programmēšanas valodās, tās lietošanu reāllaika sistēmās un drošuma paaugstināšanu vairākpavedienošanas kontekstā. Patstāvīga gatavošanās eksāmenam.
Literatūra	Obligātā/Obligatory: 1. Robert Robey, Yuliana Zamora. Parallel and High Performance Computing. Manning, 2021, 704 lpp. https://learning.oreilly.com/library/view/parallel-and-high/9781617296468/ 2. Alvin Ashcraft. Parallel Programming and Concurrency with C# 10 and .NET 6: A modern approach to building faster, more responsive, and asynchronous .NET applications using C#. Packt Publishing, 2022, 320 lpp. https://learning.oreilly.com/library/view/parallel-programming-and/9781803243672 3. Riccardo Terrell. Concurrency in .NET: Modern patterns of concurrent and parallel programming. Manning, 2018, 568 lpp. https://learning.oreilly.com/library/view/concurrency-in-net/9781617292996/ 4. Stephen Cleary. Concurrency in C# Cookbook: Asynchronous, Parallel, and Multithreaded Programming (2nd Edition). O'Reilly Media, 2019, 254 lpp. https://learning.oreilly.com/library/view/concurrency-in-c/9781492054498 5. Herbert Schildt. Java: The Complete Reference. McGraw Hill, 2018, 1248 lpp. https://learning.oreilly.com/library/view/java-the-complete/9781260463422/ 6. Javier Fernandez Gonzalez. Java 9 Concurrency Cookbook. Packt Publishing, 2017, 1300 lpp. https://learning.oreilly.com/library/view/java-9-concurrency/9781787124417/ Papildu/Additional: 1. Calvin Lin, Larry Snyder. Principles of Parallel Programming. Pearson, 2008, 360 lpp. 2. Maya Posch. Mastering C++ Multithreading: Write robust, concurrent, and parallel applications. Packt Publishing, 2017, 246 lpp. https://learning.oreilly.com/library/view/mastering-c-multithreading/9781787121706/ 3. Bertil Schmidt, Jorge Gonzalez-Dominguez, et al. Parallel Programming: Concepts and Practice. Morgan Kaufmann, 2017, 416 lpp. 4. Peter Pacheco, Matthew Malensek. An Introduction to Parallel Programming. Morgan Kaufmann, 2020, 496 lpp. 5. Peter Sanders, Kurt Mehlhorn, et al. Sequential and Parallel Algorithms and Data Structures. Springer, 2019, 524 lpp. 6. Matthew Fowler. Python Concurrency with asyncio. Manning, 2022, 376 lpp. https://learning.oreilly.com/library/view/python-concurrency-with/9781617298660/
Nepieciešamās priekšzināšanas	Procedurālā programmēšana, objektorientētā programmēšana, C un C++ valodu zināšanas.

Studiju kursa saturs

Saturs	Pilna un nepilna laika klātienes studijas		Nepilna laika neklātienes studijas	
	Kontakt stundas	Patstāv. darbs	Kontakt stundas	Patstāv. darbs
Procesi un apakšprocesi (pavedieni, aģenti). Vairākpavedienošana. Paralēlā un laiksakritīgā programmēšana.	4	4	0	0

Resursu aizsardzība. Semafori un monitori. Neatkarīgie un atkarīgie pavedieni. Pavedienu sinhronizācija.	4	4	0	0
Vairākpavedienošanas pamati .NET platformā.	2	2	0	0
Paralēlo uzdevumu bibliotēka TPL. Vārdu telpas System.Threading un System.Threading.Tasks. Klase Thread.	4	4	0	0
LINQ vaicājumi. Datu paralēlisms. Paralēlais LINQ (PLINQ).	4	4	0	0
C# valodas pamati.	2	2	0	0
Vairākpavedienošana C# valodā.	6	6	0	0
Sinhronā un asinhronā programmēšana. Delegāti .NET platformā.	2	2	0	0
Vairākpavedienošana Java valodā.	4	4	0	0
Funkcionālā programmēšana vairākpavedienšanā.	2	2	0	0
Iespējamo kļūdu analīze programmās ar vairākiem pavedieniem.	2	2	0	0
Reāllaika sistēmas vairākpavedienošanas kontekstā.	4	4	0	0
Kopā:	40	40	0	0

Sasniedzamie studiju rezultāti un to vērtēšana

Sasniedzamie studiju rezultāti	Rezultātu vērtēšanas metodes
Spēj izstrādāt daudzfunkcionālo programmatūru.	Laboratorijas darbu izpilde un aizstāvēšana. Kritēriji: spēj veidot programmu no procedūrām, funkcijām un pakotnēm (moduļiem).
Spēj izskaidrot paralēlās/laiksakritīgās programmēšanas īpatnības un izmantot ar to saistītas valodu konstrukcijas.	Laboratorijas darbu izpilde un aizstāvēšana. Kritēriji: spēj veidot pavedienus un kontrolēt to izpildi.
Spēj izstrādāt paralēlo/laiksakritīgo programmatūru C# valodā.	Laboratorijas darbu izpilde un aizstāvēšana. Kritēriji: spēj veidot pavedienus un nodrošināt resursu aizsardzību.
Spēj izstrādāt paralēlo/laiksakritīgo programmatūru Java valodā.	Laboratorijas darbu izpilde un aizstāvēšana. Kritēriji: spēj veidot pavedienu pūlus un grupas, kā arī nodrošināt resursu aizsardzību.
Spēj spriest par vairākiem paralelītātes/laiksakritības jautājumiem.	Eksāmens. Kritēriji: spēj atbildēt uz teorētiskajiem jautājumiem, rakstīt programmas, atklāt un izskaidrot kļūdas eksistējošās programmās.

Studiju rezultātu vērtēšanas kritēriji

Kritērijs	% no kopējā vērtējuma
Laboratorijas darbi	80
Eksāmens	20
Kopā:	100

Studiju kursa plānojums

Daļa	KP	Stundas			Pārbaudījumi			Brīvās izvēles pārbaudījumi		
		Lekcijas	Prakt d.	Laborat	Ieskaite	Eksām.	Darbs	Ieskaite	Eksām.	Darbs
1.	3.0	20.0	0.0	20.0		*			*	