

RTU studiju kurss "Objektorientētās programmatūras attīstība"

33000 Datorzinātnes, informācijas tehnoloģijas un enerģētikas fakultāte

Vispārējā informācija

Kods	DE0759
Nosaukums	Objektorientētās programmatūras attīstība
Studiju kursa statuss programmā	Obligātais/Ierobežotās izvēles
Atbildīgais mācītbspēks	Pāvels Rusakovs - Doktors, Asociētais profesors
Apjoms daļās un kredītpunktos	1 daļa, 6.0 kredītpunkti
Studiju kursa īstenošanas valodas	LV, EN
Anotācija	Studiju kurss veltīts objektorientētās pieejas (OOP) principu implementēšanai vairākās programmēšanas valodās. Runa ir kā par fundamentāliem OOP principiem (abstrakcija, iekapsulēšana, mantošana, polimorfisms), tā arī par papildus jēdzieniem (vairākpavedienošana, tipizācija, izņēmumu apstrāde). Uzmanības centrā ir trīs pietiekami atšķirīgas objektorientētās programmēšanas valodas: Ada, Java un Python. Studiju kursa ietvaros aplūko arī dažus specifiskus jautājumus: grafisko lietotņu izstrādi un notikumu apstrādi Java valodā, sarežģīto datu struktūru veidošanu Python valodā un citus. Tiek aplūkotas arī objektorientēto valodu funkcionālās iespējas, tajā skaitā lambda – izteiksmes. Apskatīta dažu valodu mehānismu evolūcija, tajā skaitā interfeisu koncepcijas attīstība Java valodā. Laboratorijas darbos studentiem jādemostrē apgūtās teorētiskās zināšanas un jāapgūst programmēšanas prasmes.
Mērķis un uzdevumi, izteikti kompetencēs un prasmēs	Studiju kursa mērķis ir sagatavot kvalificētus programmatūras izstrādes speciālistus, kuri var pielietot projektos objektorientētās programmēšanas valodas ar dažām konceptuālām atšķirībām. Studiju kursa uzdevumi: - attīstīt prasmi noteikt objektorientētās pieejas implementēšanas īpatnības vairākās objektorientētajās programmēšanas valodās; - iepazīstināt ar paralēlās/laiksakritīgās programmēšanas jēdzieniem un principiem; - veicināt izpratni par Java grafisko lietotņu izstrādi un notikumu apstrādi; - attīstīt prasmi rakstīt Python servera skriptus.
Patstāvīgais darbs, tā organizācija un uzdevumi	Patstāvīgais darbs tiek organizēts, studentiem individuāli apstrādājot laboratorijas darbos iegūtos datus un gatavojoties laboratorijas darbu aizstāvēšanai. Patstāvīga gatavošanās eksāmenam.
Literatūra	Obligātā. / Obligatory: 1. Grady Booch. Object Oriented Analysis And Design With Applications. Pearson India, 3rd edition, 2009, 691 lpp. 2. John Barnes. Programming in Ada 2012. Cambridge University Press, 2014, 967 lpp. 3. Andrew T. Shvets. Beginning Ada Programming: From Novice to Professional, 2019. ASIN: B082MF2MRF, 366 lpp. 4. Bruce Eckel. Thinking in Java. Pearson, 4th edition, 2006, 1150 lpp. 5. Herbert Schildt. Java: The Complete Reference. McGraw Hill, 11th edition, 2018, 1248 lpp. 6. Joyce Farrell. Java Programming. Cengage Learning, 9th edition, 2018, 870 lpp. 7. Clive Campbell. Python Programming: 3 Books in 1. Independently published, 2020, 602 lpp. 8. David Amos, Dan Bader, et al. Python Basics: A Practical Introduction to Python 3. Real Python, 2021, 635 lpp. Papildu. / Additional 9. John English. Ada 95: The Craft of Object-Oriented Programming. Prentice Hall, 1996, 486 lpp. 10. Alan Burns. Concurrent and Real-Time Programming in Ada. Cambridge University Press, 3rd edition, 2007, 476 lpp. 11. Mordechai Ben-Ari. Ada for Software Engineers: With Ada 2005. Springer, 2nd edition, 2009, 524 lpp. 12. Joshua Bloch. Effective Java. Addison-Wesley Professional, 3rd edition, 2017, 416 lpp. 13. Raoul-Gabriel Urma, Mario Fusco et al. Modern Java in Action: Lambdas, streams, functional and reactive programming. Manning, 2nd edition, 2018, 592 lpp. 14. Fabrizio Romano, Heinrich Kruger. Learn Python Programming: An in-depth introduction to the fundamentals of Python. Packt Publishing, 3rd edition, 2021, 552 lpp. 15. Francois Chollet. Deep Learning with Python. Manning, 2nd edition, 2021, 504 lpp.
Nepieciešamās priekšzināšanas	Objektorientētās programmēšanas principi, algoritmu konstruēšana.

Studiju kursa saturs

Saturis	Pilna un nepilna laika klātienes studijas		Nepilna laika neklātienes studijas	
	Kontakt stundas	Patstāv. darbs	Kontakt stundas	Patstāv. darbs
Ada valodas veidošanas vēsture, pamatprincipi un attīstības tendences.	4	4	0	0
Informācijas ievade/izvade Ada valodā. Noskaņošana un parametriskais polimorfisms.	6	6	0	0
Abstrakcija, iekapsulēšana un mantošana Ada valodā. Izņēmumu apstrāde.	4	4	0	0
Paralēla programmēšana Ada valodā. Uzdevumi. Satikšanās.	6	6	0	0
Aizsargātie tipi Ada valodā. Sinhronizācija. Semafori.	4	4	0	0
Java valodas veidošanas vēsture, pamatprincipi un attīstības tendences.	2	2	0	0
Abstrakcija, iekapsulēšana un mantošana Java valodā.	4	2	0	0
Masīvu apstrāde un kārtošana Java valodā. Interfeisi. Daudzkāršs mantošanas simulēšana.	4	6	0	0
Refleksija Java valodā. Tipu identifikācija programmas izpildes laikā (RTTI).	4	2	0	0
Izņēmumu apstrāde Java valodā.	2	2	0	0
Pakotnes 'java.util' lietošana Java valodā (klases, interfeisi, kolekcijas).	4	6	0	0
Citi Java programmēšanas aspekti: klases-čaulas, parametru nodošana.	2	2	0	0
Paralēla programmēšana Java valodā: pamatprincipi. Pavedieni, sinhronizācija.	4	4	0	0
Paralēla programmēšana Java valodā: papildu iespējas. Pakotne java.util.concurrent.	4	6	0	0
Grafisko lietotņu programmēšana Java valodā. Notikumu apstrāde. Izvietošanas menedžeri.	6	4	0	0
Python valodas koncepcija un pamatprincipi. Datu tipi, to pārveidošana, operatori.	4	2	0	0
Saraksti, korteži, masīvi, vārdnīcas Python valodā. Daudzlīmeņu struktūras. Izņēmumu apstrāde.	6	6	0	0
Apakšprogrammas Python valodā. Failu apstrāde. Lambda – izteiksmes.	4	6	0	0
Klases un objekti Python valodā. Grafiskās iespējas.	6	6	0	0
Kopā:	80	80	0	0

Sasniedzamie studiju rezultāti un to vērtēšana

Sasniedzamie studiju rezultāti	Rezultātu vērtēšanas metodes
Spēj implementēt četrus objektorientētās programmēšanas bāzes principus Ada, Java un Python valodās.	Laboratorijas darbu izpilde un aizstāvēšana. Kritēriji: spēj veidot objektorientētās programmas dažādās valodās.
Spēj saprast dažas atšķirības starp Ada'83, Ada'95, Ada'2005 un Ada'2012 standartiem.	Laboratorijas darbu izpilde un aizstāvēšana. Kritēriji: spēj saprast Ada valodas attīstības tendences.
Spēj veidot paralēlās programmas Ada un Java valodās.	Laboratorijas darbu izpilde un aizstāvēšana. Kritēriji: spēj veidot programmas ar vairākiem pavedieniem.
Spēj lietot Java valodas kolekcijas.	Laboratorijas darbu izpilde un aizstāvēšana. Kritēriji: spēj veidot daudzveidīgas Java datu struktūras un apstrādāt tās.
Spēj izstrādāt Java grafiskās lietotnes.	Laboratorijas darbu izpilde un aizstāvēšana. Kritēriji: spēj pielietot vadības elementus, izvietošanas menedžerus un apstrādāt notikumus.
Spēj izstrādāt daudzfunkcionālos skriptus Python valodā.	Laboratorijas darbu izpilde un aizstāvēšana. Kritēriji: spēj apstrādāt Python sarakstus, vārdnīcas, daudz līmeņu datu struktūras un objektus.
Spēj saprast Python grafisko programmu pamatprincipus.	Laboratorijas darbu izpilde un aizstāvēšana. Kritēriji: spēj saprast Python grafiskās lietotnes struktūru un darbības principus.
Spēj demonstrēt teorētiskās un praktiskās zināšanas par objektorientētajām iespējām dažās programmēšanas valodās citam speciālistam.	Eksāmena darba rakstīšana. Kritēriji: spēj orientēties valodu konstrukcijās, meklēt kļūdas eksistējošos risinājumos un piedāvāt savus risinājumus.

Studiju rezultātu vērtēšanas kritēriji

Kritērijs	% no kopējā vērtējuma
Laboratorijas darbi	50
Eksāmens	50
Kopā:	100

Studiju kursa plānojums

Daļa	KP	Stundas			Pārbaudījumi		
		Lekcijas	Prakt d.	Laborat	Ieskaite	Eksām.	Darbs
1.	6.0	32.0	0.0	32.0		*	