VENTSPILS AUGSTSKOLA

Inženieru iela 101, Ventspils, LV-3601, Tālrunis (371) 636 28 303; Fakss (371) 636 29 660
Reģ.№ 90000362426 e-pasts: venta@venta.lv

Ventspilī

11.04.2018. Nr. 1-147

Augstākās izglītības kvalitātes aģentūras (AIKA)
Studiju akreditācijas komisijai

Par pirmā līmeņa profesionālās augstākās izglītības
studiju programmas "Programmēšanas speciālists"
īstenošanu angļu valodā

Lūdzu apstiprināt izmaiņas Ventspils Augstskolas Informācijas Tehnoloģiju fakultātes īstenotajā studiju virzienā "Informācijas tehnoloģija, datortehnika, elektronika, telekomunikācijas, datorvadība un datorzinātne", pirmā līmeņa profesionālās augstākās izglītības studiju programmā "Programmēšanas speciālists" - studiju programmas īstenošana angļu valodā, pamatojoties uz Ventspils Augstskolas Attīstības stratēģijas laika posmam no 2016. līdz 2020. gadam (apstiprināta ar Ventspils Augstskolas Senāta lēmumu Nr.16-93) studiju attīstības mērķiem: 1. "Palielināt uzņemto studentu skaitu un samazināt studentu atbirumu", 2. "Palielināt pilna laika ārvalstu studentu skaitu VeA" un 3.e. "Studiju programmu īstenošana svešvalodās", un pamatojoties uz Ventspils pilsētas Attīstības programmas 2014. - 2020.gadam uzdevumu Nr. S-2-1 "Sekmēt Ventspils Augstskolas, kā starptautiski atzīta Baltijas līmeņa augstākās izglītības un zinātnes centra attīstību" un pasākumu Nr. S-2-1-2 "Augstākās izglītības eksportspēju veicināšana, studiju programmu ārvalstu studentiem atvēršana", lai nodrošinātu iespēju ārvalstu studentiem apgūt VeA īstenotās studiju programmas angļu valodā.

Pielikumā:

1. Izraksts Nr.5 no Ventspils Augstskolas Senāta 2018. gada 4. aprīļa sēdes protokola ar lēmumu Nr. 18-35: Apstiprināt pirmā līmeņa profesionālās augstākās izglītības studiju

programmas “Programmēšanas speciālists” izmaiņu pieteikšanai (studiju programmas īstenošanai angļu valodā) sagatavotos dokumentus un pilnvarot Karīnu Šķirmanti tos iesniegt Akadēmiskās informācijas centram

2. Dokumentu kopums pirmā līmeņa profesionālās augstākās izglītības studiju programmas “Programmēšanas speciālists” izmaiņu pieteikuma iesniegšanai Akadēmiskās Informācijas Centram.

Ventspils Augstskolas rektors K.Krēsliņš

Sagatavoja
K. Šķirmante
26890524

IZRAKSTS

VENTSPILS AUGSTSKOLA

IZRAKSTS NO SENĀTA

SĒDES PROTOKOLA

04.04.2018. Ventspilī Nr. 5.

Sēdi vada: G. Hiļķeviča

Sēdē piedalās: senatori: V. Avotiņš, V. Balama, J. Baldunčiks, R. Didrihsons, G. Dreijers, M. Ēlerts, J. Freimanis, G. Hiļķeviča, S. Hiļķevičs, K. Krēsliņš, L. Ločmele, L. Resele, R. Rollande, I. Vizule, E. Vītola

Nepiedalās: I. Balode, A. Dubova, F. Kamiševs, A. Mažaika, S. Šama,

Sēdi protokolē: K. Matule

Sēdes sākums: plkst. 16:10

DARBA KĀRTĪBA:

***

9. Par Informācijas tehnoloģiju fakultātes realizētās pirmā līmeņa profesionālās augstākās izglītības studiju programmas "Programmēšanas speciālists" izmaiņām studiju programmas īstenošanai angļu valodā (ziņo E. Vītola)

***

9. Par Informācijas tehnoloģiju fakultātes realizētās pirmā līmeņa profesionālās augstākās izglītības studiju programmas "Programmēšanas speciālists" izmaiņām studiju programmas īstenošanai angļu valodā

***

Lēmums Nr. 18 – 35: Apstiprināt pirmā līmeņa profesionālās augstākās izglītības studiju programmas "Programmēšanas speciālists" izmaiņu pieteikšanai (studiju programmas īstenošanai angļu valodā) sagatavotos dokumentus un pilnvarot Karīnu Šķirmanti tos iesniegt Akadēmiskās informācijas centram.

Pielikumā:

1. *Ventspils Augstskolas Informācijas tehnoloģiju fakultātē īstenotā pirmā līmeņa profesionālās augstākās izglītības studiju programmas "Programmēšanas speciālists" dokumentu kopums (veicot izmaiņas studiju programmas īstenošanai angļu valodā).*
2. *Izraksts no Ventspils Augstskolas Informācijas tehnoloģiju fakultātes Domes sēdes, lēmums Nr.18-03-05.*

***

Senāta sēdes vadītāja: G Hiļķeviča

Sēdes protokolētāja: K. Matule

IZRAKSTS PAREIZS:

VeA Senāta sekretāre

Matule

K. Matule

Ventspilī 2018.gada 11. aprīlī

VENTSPILS AUGSTSKOLA

SENĀTS

LĒMUMS

Ventspilī

2018. gada 4. aprīlī Nr. 18 - 35

Par Informācijas tehnoloģiju fakultātes
realizētās pirmā līmeņa profesionālās augstākās
izglītības studiju programmas "Programmēšanas speciālists"
izmaiņām studiju programmas īstenošanai angļu valodā

Noklausījies Informācijas tehnoloģiju fakultātes dekāna Māra Ēlerta ziņojumu un ņemot vērā Informācijas tehnoloģijas fakultāte Domes 2018. gada 21. marta sēdes lēmumu Nr.18-03-05,
Ventspils Augstskolas Senāts
nolemj:

apstiprināt pirmā līmeņa profesionālās augstākās izglītības studiju programmas "Programmēšanas speciālists" izmaiņu pieteikšanai (studiju programmas īstenošanai angļu valodā) sagatavotos dokumentus un pilnvarot Karīnu Šķirmanti tos iesniegt Akadēmiskās informācijas centram.

Pielikumā:

1. Ventspils Augstskolas Informācijas tehnoloģiju fakultātē īstenotā pirmā līmeņa profesionālās augstākās izglītības studiju programmas "Programmēšanas speciālists" dokumentu kopums (veicot izmaiņas studiju programmas īstenošanai angļu valodā).
2. Izraksts no Ventspils Augstskolas Informācijas tehnoloģiju fakultātes Domes sēdes, lēmums Nr.18-03-05.

Senāta priekšsēdētāja G. Hiļķeviča

K. Šķirmante
26890524

Kopijas:
1-ITF
1-mācību daļai
3.eks.lietā

Izraksts no ITF domes sēdes

Ventspilī

2018. gada 21. martā Nr. 3

Sēdi vada: ITF domes priekšsēdētājs A. Krauze
Protokolē: ITF domes sekretāre: K. Matule
Piedalās: asoc. prof.: G. Hiļķeviča
docenti: M. Ēlerts, R. Rollande,
lektori: J. Šate, G. Neimanis
studenti: A. Lurins, K. Reinis Ozols, P. Bitāns, R. R. Vecmanis
vispārējais personāls: K. Matule

__Dienas kārtība:__

1. *
2. *
3. *
4. *
5. **Studiju programmas "Programmēšanas speciālists" sagatavoto dokumentu kopumu apstiprināšana iesniegšanai AIC studiju programmas īstenošanai angļu valodā**
6. *
7. *
8. *

***

5. Studiju programmas "Programmēšanas speciālists" sagatavoto dokumentu kopumu apstiprināšana iesniegšanai AIC studiju programmas īstenošanai angļu valodā

***

Lēmums Nr. 18-03-05: **Lūgt VeA Senātam apstiprināt studiju programmas "Programmēšanas speciālists" dokumentu kopumu iesniegšanai Akadēmiskās informācijas centram būtisku izmaiņu veikšanai studiju programmā.**

***

Izrakstu sagatavoja: Matule ITF domes sekretāre K. Matule

VENTSPILS AUGSTSKOLA
IT FAKULTĀTE
LATVIJAS REPUBLIKA

VENTSPILS AUGSTSKOLA

INFORMĀCIJAS TEHNOLOĢIJU FAKULTĀTE

PIRMĀ LĪMEŅA PROFESIONĀLĀS AUGSTĀKĀS IZGLĪTĪBAS STUDIJU PROGRAMMA

PROGRAMMĒŠANAS SPECIĀLISTS

DOKUMENTU KOPUMS
(VEICOT IZMAIŅAS STUDIJU PROGRAMMAS ĪSTENOŠANAS VALODĀ)

Ventspils 2018

ATBALSTĪTS APSTIPRINĀŠANAI VeA SENĀTĀ
VeA ITF Domes sēdē 2018. g. 21. martā, lēmums Nr. 18-03-05
VeA ITF Domes priekšsēdētājs A. Krauze

APSTIPRINĀTS
VeA Senāta sēdē 2018. g. 4. aprīlī, lēmums Nr. 18-35
VeA Senāta priekšsēdētājs G. Hiļķeviča

SATURS

STUDIJU PROGRAMMAS APRAKSTS 3

STUDIJU PROGRAMMAS SATURS ANGĻU VALODĀ 4

STUDIJU PLĀNA ĪSTENOŠANA ANGĻU VALODĀ 5

PRASĪBAS ATTIECĪBĀ UZ IEPRIEKŠĒJO IZGLĪTĪBU 7

PIELIKUMI 8

STUDIJU PROGRAMMAS APRAKSTS

Studiju programmas nosaukums angļu valodā	**Programming Specialist**
Studiju programmas nosaukums latviešu valodā	**Programmēšanas speciālists**
Studiju programmas veids un līmenis	Pirmā līmeņa profesionālā augstākā izglītība
KP	80 KP
Kvalifikācija	Programmētājs
Kvalifikācijas līmenis	4
Studiju veids	Pilna laika studijas
Izglītības kvalifikācijas kods	41484
Profesijas kvalifikācijas kods	2512 05

STUDIJU PROGRAMMAS SATURS ANGĻU VALODĀ

No. (Nr.)	Title (Nosaukums)	Credit Points (KP)
	General study courses (VispārizglĪtojošie mācĪbu kursi) (20 Credit Points (KP))	**20**
1	Mathematics for Programmers	8
2	English for Programming - C part (2 KP)	
3	Security of Information systems	2
4	Communication and Professional Ethics	2
5	Business and Economics	2
6	Basics of IT industry rules & regulations & standards	2
7	IT project management	2
8	Fundamentals of the Latvian Language	2
	Field related study courses (Nozares mācĪbu kursi) (36 Credit Points (KP))	36
	Compulsory selection part (Obligātie mācĪbu kursi)	***16***
9	Data Structures and Algorithms(JAVA)	2
10	Database	4
11	Fundamentals of JAVA Programming	4
12	JAVA Programming	2
13	Software Design Patterns	4
	Courses for a particular profession (Konkrētās profesijas mācĪbu kursi)	***18***
14	Software engineering	6
15	Web page programming	4
16	Development operations tools	2
17	Software testing and debugging technologies and principles	2
18	Embedded Systems	4
	Optional courses (Izvēles mācĪbu kursi)	***2***
19	Software Architecture Design	2
20	Internet and computer networks	2
21	UNIX like system administration	2
22	Parallel computing	2
	Internship (Prakse)	16
23	Internship	16
	Qualification paper (Kvalifikācijas darbs)	8
24	Qualification paper	8
	Total (Kopā)	**80**

STUDIJU PLĀNA ĪSTENOŠANA ANGĻU VALODĀ

				Semeter(Semestris)					
No. (Nr.)	**Title in English(Nosaukums angļu valodā)**	**Title in Latvian(Nosaukums latviešu valodā)**	**Credit Points (KP)**	**I**	**II**	**III**	**IV**	**Form of evaluation (Pārbaudes forma)**	**Lecturer (Pasniedzēji)**
	General study courses (Vispārizglītojošie mācību kursi)		**20**						
1	Mathematics for Programmers	Matemātika programmētājiem	8	8				Examination (Eksāmens)	J. Mihailova; E.Vītola
2	English for Programming - C part (2 KP)	Angļu valoda - C daļas kurss	2					Examination (Eksāmens)	V. Balama
3	Security of Information systems	Informācijas sistēmu drošība	2			2		Examination (Eksāmens)	S.Meijere
4	Communication and Professional Ethics	Saskarsme un profesionālā ētika	2		2			Examination (Eksāmens)	G. Dreijers
5	Business and Economics	Ekonomika un komercdarbība	2		2			Examination (Eksāmens)	V. Zeps
6	Basics of IT industry rules & regulations & standards	IT nozares tiesību pamati un standarti	2		2			Examination (Eksāmens)	S.Meijere
7	IT project management	IT projektu vadīšana	2	2				Test (ieskaite)	U. Kuplis
8	Fundamentals of the Latvian Language	Latviešu valodas pamati			2			Test (ieskaite)	S.Ozoliņa
	Field related study courses (Nozares mācību kursi)		**36**						
	Compulsory selection part (Obligātie mācību kursi)		***16***						
9	Data Structures and Algorithms(JAVA)	Datu struktūras un algoritmi (JAVA)	2	2				Examination (Eksāmens)	K. Šķirmante
10	Database	Datu bāzes	4	2	2			The 1th semester - examination, the 2nd semester - examination (1. semestrī-eksāmens, 2.semestrī - eksāmens)	G. Neimanis
11	Fundamentals of JAVA Programming	Programmēšanas pamati JAVA (BLOKĀ 1 mēnesis)	4	4				Examination (Eksāmens)	K. Šķirmante
12	JAVA Programming	Programmēšana JAVA	2	2				Examination (Eksāmens)	K. Šķirmante

13	Software Design Patterns	Programmatūras izstrādes šabloni	4		2	2		The 2nd semester - examination, the 3rd semester - examination (2. semestrī-eksāmens, 3.semestrī - eksāmens)	R.Lasmanis
	Courses for a particular profession (Konkrētās profesijas mācību kursi)		***18***						
14	Software engineering	Programmatūras inženierija	6		2	4		The 2nd semester - test, the 3rd semester - examination (2. semestrī-ieskaite, 3.semestrī - eksāmens)	R. Rollande; K. Šķirmante
15	Web page programming	WEB aplikāciju izstrāde	4		4			Examination (Eksāmens)	A. Traškovs
16	Development operations tools	Rīki programmatūras risinājumu nodrošināšanai	2			2		Test (ieskaite)	K. Zvaigzne
17	Software testing and debugging technologies and principles	Programmatūras testēšana un atkļūdošanas tehnoloģijas un prinipi	2		2			Examination (Eksāmens)	A.Frišfelds, E.Grīnfelds
18	Embedded systems	Iegultās lietojumprogrammas	4			4		Examination (Eksāmens)	G.Dreifogels
	Optional courses (Izvēles mācību kursi)		***2***						
19	Software Architecture Design	Programmatūras arhitektūras projektēšana	2			2		Examination (Eksāmens)	E. Palacis
20	Internet and computer networks	Interneta un datortīkla tehnoloģijas	2			2		Examination (Eksāmens)	M.Koloda
21	UNIX like system administration	UNIX sistēmu administrēšana	2			2		Examination (Eksāmens)	M.Koloda
22	Parallel computing	Paralēlā programmēšana	2			2		Examination (Eksāmens)	K.Gromovs
	Internship (Prakse)		**16**					Defence (aizstāvēšana)	
23	Internship	Prakse	**16**			4	12		
	Qualification paper (Kvalifikācijas darbs)		**8**					Defence (aizstāvēšana)	
24	Qualification paper	Kvalifikācijas darbs	**8**				8		
		Total (Kopā)	**80**	**20**	**20**	**20**	**20**		

PRASĪBAS ATTIECĪBĀ UZ IEPRIEKŠĒJO IZGLĪTĪBU

Tiesības studēt pirmā līmeņa profesionālās augstākās izglītības studiju programmā “Programmēšanas speciālists” (“Programming Specialist”), kas tiek īstenota angļu valodā, ir Latvijas Republikas (LR) pilsoņiem un personām ar LR nepilsoņa pasi, kā arī personām, kurām ir izsniegtas pastāvīgās uzturēšanās atļaujas. Ārzemnieki var studēt programmā saskaņā ar LR Augstskolu likuma 83. un 85. pantu.

Programmā tiek uzņemti reflektanti, kuriem ir vispārējā vidējā izglītība vai profesionālā vidējā izglītība (3.kvalifikācijas līmenis saskaņā ar LR Profesionālās izglītības likuma 5.panta 3) apakšpunktu). Uzņemot studiju programmā, reflektantu kopējais vērtējums veidojas proporcionāli no divām daļām:

1) centralizētā eksāmena matemātikā kopvērtējuma (ja reflektants vidējo izglītību ir ieguvis LR) vai matemātikas vērtējuma noteikšanai ir jākārto Ventspils Augstskolas organizēts iestājpārbaudījums (ja reflektants vidējo izglītību ir ieguvis ārvalstīs) (50%)
2) centralizētā eksāmena angļu valodā kopvērtējuma (ja reflektants vidējo izglītību ir ieguvis LR) vai starptautiskas testēšanas institūcijas angļu valodā pārbaudījuma rezultāta (saskaņā ar MK noteikumiem Nr.543 “Noteikumi par svešvalodas centralizētā eksāmena vispārējās vidējās izglītības 5 programmā aizstāšanu ar starptautiskas testēšanas institūcijas pārbaudījumu svešvalodā”) vai angļu valodas vērtējuma noteikšanai ir jākārto Ventspils Augstskolas organizēts iestājpārbaudījums (ja reflektants vidējo izglītību ieguvis ārvalstīs un nav kārtots starptautiskas testēšanas institūcijas angļu valodas pārbaudījums) (50%).

Uzņemšanas kārtību nosaka VeA uzņemšanas noteikumi.

PIELIKUMI

1. Pirmā līmeņa profesionālās augstākās izglītības studiju programmas “Programmēšanas speciālists” īstenošana angļu valodā (“Programming Specialist”) - studiju programmā īstenošanā iesaistāmo docētāju saraksts, docētāju piekritība vadīt kursus angļu valodā un docētāju apstiprinājums par vismaz B2 angļu valodas līmeņa atbilstību
2. Pirmā līmeņa profesionālās augstākās izglītības studiju programmas “Programmēšanas speciālists” īstenošana angļu valodā (“Programming Specialist”) - docētāju CV angļu valodā
3. Pirmā līmeņa profesionālās augstākās izglītības studiju programmas “Programmēšanas speciālists” īstenošana angļu valodā (“Programming Specialist”) - kursu apraksti angļu valodā

Pirmā līmeņa profesionālās augstākās izglītības studiju programmas “Programming Specialist” īstenošanā iesaistāmo mācībspēku saraksts

N.p. k.	Vārds, uzvārds	Izglītība: akadēmiskais vai zinātniskais grāds	Amats, institūcija	Ievēlēts vai uz laiku pieņemts darbā	Pasniedzamie studiju kursi	Paraksts par piekritību pasniegt norādīto studiju kursu vai kursus angļu valodā	Paraksts par vismaz B2 angļu valodas līmeņa atbilstību
1	Agris Traškovs	Mg.sc.comp., studē doktorantūrā LU	VeA lektors, SIA “Accenture” programmētājs.	Līgums (tiks slēgts)	Web page programming (WEB lietojumprogrammu izstrāde); Development operations tools (Rīki programmatūras risinājumu nodrošināšanai)		
2	Estere Vītola	Mg.paed., studē doktorantūrā RTU	VeA ITF lektors	Ievēlēta	Mathematics for Programmers (Matemātika programmētājiem)		
3	Ralfs Lasmanis	Bsc.sc.comp	SIA “Accenture” programmētājs.	Līgums (tiks slēgts)	Software Design Patterns (Programmatūras izstrādes šabloni)		

Pirmā līmeņa profesionālās augstākās izglītības studiju programmas “Programming Specialist” īstenošanā iesaistāmo mācībspēku saraksts (turpinājums)

N.p. k.	Vārds, uzvārds	Izglītība: akadēmiskais vai zinātniskais grāds	Amats, institūcija	Ievēlēts vai uz laiku pieņemts darbā	Pasniedzamie studiju kursi	Paraksts par piekritību pasniegt norādīto studiju kursu vai kursus angļu valodā	Paraksts par vismaz B2 angļu valodas līmeņa atbilstību
4	Edgars Palacis	Mg.sc.ing.	SIA “VISMA” vecākais sistēmanalītiķis	Līgums (tiks slēgts)	Software Architecture Design (Programmatūras arhitektūras projektēšana)		
5	Sintija Ozoliņa	Mg.philol.	VeA TSF lektors	Līgums (tiks slēgts)	Fundamentals of the Latvian Language (Latviešu valodas pamati)		
6	Mārcis Koloda	Mg.sc.comp	VeA datorlaboratorijas vadītājs	Līgums (tiks slēgts)	UNIX like system administration (UNIX sistēmu administrēšana); Internet and computer networks (Interneta un datortīkla tehnoloģijas)		

Pirmā līmeņa profesionālās augstākās izglītības studiju programmas “Programming Specialist” īstenošanā iesaistāmo mācībspēku saraksts (turpinājums)

N.p. k.	Vārds, uzvārds	Izglītība: akadēmiskais vai zinātniskais grāds	Amats, institūcija	Ievēlēts vai uz laiku pieņemts darbā	Pasniedzamie studiju kursi	Paraksts par piekritību pasniegt norādīto studiju kursu vai kursus angļu valodā	Paraksts par vismaz B2 angļu valodas līmeņa atbilstību
7	Uldis Kuplis	Mg.oec.	SIA “Baltijas Datoru Akadēmija” lektors	Līgums (tiks slēgts)	IT project management (IT projektu vadīšana)		
8	Gints Neimanis	Mg.sc.soc.	VeA ITF lektors	Ievēlēts	Database technologies (Datu bāzes)		
9	Guntars Dreijers	Dr. philol.	VeA TSF docents	Ievēlēts	Communication and Professional Ethics (Saskarsme un profesionālā ētika)		
10	Jeļena Mihailova	Mg.math.	VeA ITF lektors	Ievēlēta	Mathematics for Programmers (Matemātika programmētājiem)		

Pirmā līmeņa profesionālās augstākās izglītības studiju programmas "Programming Specialist" īstenošanā iesaistāmo mācībspēku saraksts (turpinājums)

N.p.k.	Vārds, uzvārds	Izglītība: akadēmiskais vai zinātniskais grāds	Amats, institūcija	Ievēlēts vai uz laiku pieņemts darbā	Pasniedzamie studiju kursi	Paraksts par piekritību pasniegt norādīto studiju kursu vai kursus angļu valodā	Paraksts par vismaz B2 angļu valodas līmeņa atbilstību
11	Raita Rollande	Dr.sc.ing.	VeA ITF docente	Ievēlēta	Software engineering (Programmatūras inženierija)		
12	Vita Balama	Dr. paed.	VeA ITF docente	Ievēlēta	Business English for Programming (Angļu valoda programmētājiem) - C daļas kurss		
13	Sanita Meijere	MBA, studē doktorantūrā RTU	SIA "PlayGineering Systems"	Līgums (tiks slēgts)	Basics of IT industry rules & regulations & standards (IT nozares tiesību pamati un standarti); Security of Information systems (Informācijas sistēmu drošība)		

Pirmā līmeņa profesionālās augstākās izglītības studiju programmas “Programming Specialist” īstenošanā iesaistāmo mācībspēku saraksts (turpinājums)

N.p. k.	Vārds, uzvārds	Izglītība: akadēmiskais vai zinātniskais grāds	Amats, institūcija	Ievēlēts vai uz laiku pieņemts darbā	Pasniedzamie studiju kursi	Paraksts par piekritību pasniegt norādīto studiju kursu vai kursus angļu valodā	Paraksts par vismaz B2 angļu valodas līmeņa atbilstību
14	Katrīna Zvaigzne	Studē VeA maģistra studiju programmā “Datorzinātnes” 2. kursā	SIA “Accenture” programmētāja	Līgums (tiks slēgts)	Development operations tools (Rīki programmatūras risinājumu nodrošināšanai)		
15	Karina Šķirmante	Mg.sc.comp., studē doktorantūrā LU	VeA Inženierzinātņu institūta “VSRC” zinātniskā asistente	Ievēlēta	Data Structures and Algorithms(JAVA) (Datu struktūras un algoritmi (JAVA));Fundamentals of JAVA Programming(Programmēšanas pamati JAVA);JAVA Programming (Programmēšana JAVA);Software engineering (Programmatūras inženierija)		

Pirmā līmeņa profesionālās augstākās izglītības studiju programmas “Programming Specialist” īstenošanā iesaistāmo mācībspēku saraksts (turpinājums)

N.p. k.	Vārds, uzvārds	Izglītība: akadēmiskais vai zinātniskais grāds	Amats, institūcija	Ievēlēts vai uz laiku pieņemts darbā	Pasniedzamie studiju kursi	Paraksts par piekritību pasniegt norādīto studiju kursu vai kursus angļu valodā	Paraksts par vismaz B2 angļu valodas līmeņa atbilstību
16	Viesturs Zeps	Mg.sc.soc., studē doktorantūrā Banku augstskolā	InnoEnergy SE projektu vadītājs, SIA “TransfoElectric” vadītājs/līdzīpašnieks	Līgums (tiks slēgts)	Business and Economics (Ekonomika un komercdarbība)		
17	Gints Dreifogels	Mg. Sc. Ing.	VeA institūts “Ventspils Starptautiskais Radioastronomijs Centrs “, kosmosa komunikāciju inženieris	Līgums (tiks slēgts)	Embedded systems (Iegultās sistēmas)	Dreifogels	Dreifogels

Pirmā līmeņa profesionālās augstākās izglītības studiju programmas "Programming Specialist" īstenošanā iesaistāmo mācībspēku saraksts (turpinājums)

N.p. k.	Vārds, uzvārds	Izglītība: akadēmiskais vai zinātniskais grāds	Amats, institūcija	Ievēlēts vai uz laiku pieņemts darbā	Pasniedzamie studiju kursi	Paraksts par piekritību pasniegt norādīto studiju kursu vai kursus angļu valodā	Paraksts par vismaz B2 angļu valodas līmeņa atbilstību
18	Kristaps Gromovs	Mg.sc.comp	SIA "MT Services" programmētājs	Līgums (tiks slēgts)	Parallel computing (Paralēlā programmēšana)	Gromovs	Gromovs
19	Andrejs Frišfelds	Mg.sc.comp.	VeA lektors, SIA "TestDevLatb" valdes loceklis	Līgums (tiks slēgts)	Software testing and debugging technologies and principles (Programmatūras testēšana un atkļūdošanas tehnoloģijas un principi)		
20	Ervīns Grīnfelds	Mg.sc.comp.	VeA lektors, SIA "TestDevLatb" valdes loceklis	Līgums (tiks slēgts)	Software testing and debugging technologies and principles (Programmatūras testēšana un atkļūdošanas tehnoloģijas un principi)		

Web page programming
(WEB lietojumprogrammu izstrāde)

Author	Agris Traskovs
Course Code	DatZ1032
Form of evaluation	Exam
Credit point (ECTS credit points)	4 **(6 ECTS)**
Prerequisites	OOP, Web development 101
Course group	Industry study course

Objective

Give the students insight on modern web application frameworks, REST API usage and practical experience creating responsive web applications.

Learning outcomes

Students have insight in web technologies currently used in industry.
Students have attained practical skills in creating web applications.
Can use most common data structures and formats in web communication.
Have an understanding how different Javascript frameworks function

Organization mode of students individual assignment

The course consists of contact hours with students and their independent work with literature and internet resources. Contact hours include theoretical lectures and practical sessions, where students strengthen their theoretical knowledge implementing the things learned in the theoretical part of the course.

Evaluation of learning outcomes

Final grade is composed of 50% grade in the final exam, 50% grades in practical assignments, both must be higher or equal to 4 on a scale of 10.

Course outline

Week	**Topic**	**Type (lecture, seminar, laboratory work)**
1	Introduction to currently used technologies. Architectural influence in web application development. Single page applications.	Lecture, seminar
2	**Fundamentals of web page creation**. HTML5, CSS3. Images and different media usage.	Lecture, seminar
3	Responsive design, flexible grids. DOM and its usage in element distinction and search	Lecture, seminar
4	**Frontend templates and tools.** LESS/SASS, Gulp, ES6 templates, Handlebars, Git	Lecture, seminar

5	**JQuery.** What is JQuery, when to use it. Basic formatting functions, advanced data processing.	Lecture, seminar
6	**Communications used in web applications**. Ajax calls JSON, XML data structures. REST API, their usage.	Lecture, seminar
7	**Javascript Frameworks**. Controllers and data binding. EmberJS	Lecture, seminar
8	**Javascript Frameworks**. Services. Data manipulation operations. Components and directives. AngularJS	Lecture, seminar
9	**NodeJS.** NPM,Callbacks, usage and different features	Lecture, seminar
10	**Webpage security**: Authorization/ Authentication. OAuth. Testing functionality and security. Communication security issues.	Lecture, seminar
11	Webpage architecture, Project planning, process automation.	Lecture, seminar
12-16	Course Project in groups	Laboratory work

Basic literature

- Responsive Web Design with HTML5 and CSS3 – Ben Frain – ISBN:978-1-78439-893-4
- AngularJS documentation: https://angularjs.org
- NodeJS documentation: https://nodejs.org
- EmberJS documentation https://guides.emberjs.com/

Supplementary literature

Other source of information

Development operations tools
(Rīki programmatūras risinājumu nodrošināšanai)

Author	Mg.sc.comp. Katrina Zvaigzne
Course Code	DatZ2017
Form of evaluation	Test
Credit point (ECTS credit points)	2 (3 ECTS)
Prerequisites	Java
Course group	Industry study course

Objective
The course is a practical introduction to the tools commonly used in the world of DevOps, Software engineering and Continuous Delivery. With a focus on lab-based learning, students will be introduced to, and experience first-hand, the latest advanced engineering techniques which, when applied, can apply a huge benefit to project delivery.

Learning outcomes
- Students have the knowledge about DevOps fundamentals
- Students have the knowledge and practical skills to use the cloud and build environments using industrialized techniques
- Students have the knowledge of core tools, and their roles in continuous delivery and DevOps
- Students have the knowledge and practical skills on how to stand up a working tools environment capable of being used on real clients

Organization mode of students individual assignment
The course consists of contact hours with students and work with literature and internet resources. Contact hours include practical sessions, where students gain the necessary practical skills.

Evaluation of learning outcomes
Final grade is composed of 70% practical assignment grades and 30% of final test. Both must be higher than 4 on a scale of 10.

Course outline

Week	**Topic and subtopic**	**Type** (lecture, seminar, laboratory work)
1	DevOps fundamentals. Code versioning tools and code repositories. Git. Gerrit. Code versioning, management, Maven	Seminar

2	Cloud services.Amazon Web Services, private clouds, virtual machines/servers.	Seminar
3	Continuous delivery tools. Jenkins	Seminar
4	Code Quality tools. Sonar, Quality gates, integration with other tools and pipelines.	Seminar
5	Configuration Management. Infrastructure automation. Chef, infrastructure building.	Seminar
6	Beyond virtual machines. Containers vs virtual machines, Docker	Seminar
7	Amazon web services and stacks. AWS Cloudformation. Instances. Instance management.	Seminar
8	Operations. Monitoring in cloud. Monitoring tools. ELK (Elacticsearch, logstash, Kibana), Triggers and alarms.	Seminar

Basic literature

The Goal: A Process of Ongoing Improvement, Eliyahu M. Goldratt, Jeff Cox, David Whitford, ISBN: 0884271951
Continuous Delivery, Humble J. and Farley D., Addison-Wesley Professional, 2010, ISBN-10: 0321601912
The DevOps Handbook, Kim G., Debois P., Willis J., Humble J., IT Revolution Press, 2016, ISBN-10: 1942788002

Supplementary literature

https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elk-stack-on-ubuntu-14-04
https://learn.chef.io/tutorials/
https://docs.docker.com/windows/started/

Other source of information

- https://www.accenture.com/us-en/blogs/blogs-accenture-devops
- http://continuousdelivery.com/
- https://blog.accenture.com/martin_croker/martin_croker
- http://docs.aws.amazon.com/gettingstarted/latest/awsgsg-intro/gsg-aws-tutorials.html

Data Structures and Algorithms (JAVA)
Datu struktūras un algoritmi (JAVA)

Author	Mg.sc.comp. Karina Šķirmante
Course Code	DatZ1024
Form of evaluation	Examination
Credit point (ECTS credit points)	2 (ECTS 3)
Prerequisites	Proficiency in JAVA
Course group	Industry study course

Objective
The aim of this course is to provide students with information about fundamental data structures, including worst-case space/time efficiency and implementation details. Relevant algorithms related to the data structures will be covered as appropriate.

Learning outcomes
Upon successful completion of the course, students should

- be able to describe data structures from three perspectives–logical, application, and implementation,
- be able to implement fundamental data structures such as queues, stacks, trees, heaps, graphs.

Organization mode of students individual assignment
Systematic work during semester includes:

- regular learning using lecture materials, literature, internet resources,
- completion of home assignments,
- preparation for tests and the exam,

weekly teacher consultations.

Evaluation of learning outcomes
Final exam consists of two parts:

- theory (30% of total grade)
- programming (70% of total grade)

To be allowed to take the final exam student has to submit all home assignments given during the semester and the average grade for the home assignments has to be at least 4. If the average grade of home assignments is 8 or higher, the student can choose not to write the programming part of the final exam. In this case 70% of total grade is replaced by the average grade of home assignments.
During the semester students have to take two theoretical tests. If the average result of these tests is 8 or higher, the student can choose not to write the theory part of the exam. In this case 30% of total grade is replaced by the average grade for tests.

Course outline

Week	**Topic and subtopic**	**Type** (lecture, seminar, laboratory

		work)
1.	Concept of data and data type. Data structures and their classification. Commonly used data structures. Arrays. Records. Strings.	Lecture
2.	Lists: operations and implementation.	Seminar
3.-4.	Stack and queue: implementation, operations and application	Lecture, seminar
5.-6.	Linked structures and their operations	Lecture, seminar
7.-8.	Priority queues and heaps: operations, implementation	Lecture, seminar
9.-10.	First test. Trees: terminology, binary trees, binary search trees, operations, full binary tree, balanced tree, tree traversal and implementation.	Test, lecture, seminar
11.-12.	Implementation of binary search tree, copying trees, trees implemented with arrays, expression trees.	Lecture, seminar
13.-14.	Graphs: basics and terminology, implementation, traversals	Lecture, seminar
15.-16.	Second test. Graphs: minimum spanning tree, shortest path algorithms - Dijkstra's algorithm, Bellman–Ford algorithm and others algorithms	Test, lecture, seminar

Basic literature

1. Adam Drozdek, *Data Structures and Algorithms in JAVA, Second Edition*, Thomson Course Technology, ISBN: 0-534-49252-5, 2005
2. Donald E. Knuth, *The Art of Computer Programming, Volume 1: Fundamental Algorithms. Third Edition* (Reading, Massachusetts: Addison-Wesley, 1997), xx+650pp. ISBN 0-201-89683-4
3. Donald E. Knuth, *The Art of Computer Programming, Volume 3: Sorting and Searching. Second Edition* (Reading, Massachusetts: Addison-Wesley, 1998), xiv+780pp.+foldout. ISBN 0-201-89685-0
4. Donald E. Knuth, *The Art of Computer Programming, Volume 4, Fascicle 4: Generating all Trees, History of Combinatorial Generation*, (Addison-Wesley, February 6, 2006) vi+120pp, ISBN 0-321-33570-8

Supplementary literature

1. Douglas Baldwin, Greg W. Scragg, *Algorithms and Data Structures, The Science of Computing*, Charles River Media, 2004.

Other source of information

MIT OpenCourseWare, Massachusetts Institute of Technology, course “Computer Algorithms in Systems Engineering” topics “Data Structures”, https://ocw.mit.edu/courses/

Mathematics for Programmers
(Matemātika programmētājiem)

Author	mg. math. Jeļena Mihailova mg. paed. Estere Vītola
Course Code	Mate1022
Form of evaluation	Examination
Credit point (ECTS credit points)	8 (12 ECTS)
Prerequisites	Secondary School Mathematics
Course group	General education study course

Objective

Understand the importance of mathematics in programming and programming capabilities in solving mathematical problems. Learn mathematical and algorithmic thinking.

Acquire the basics of linear algebra, analytic geometry, vector algebra, mathematical analysis and probability theory using programming.

Acquire the skills to create simple computer programs using the programming language *Python* to solve mathematical problems.

Learning outcomes

Students understand the basic concepts and rules of set theory and mathematical logic and can to apply that for problems solve.

Students are able to take basic operations with matrices and to solve linear systems by Cramer's rule, by Gaussian elimination, by finding the inverse of the coefficient matrix.

Students are able to take operations with vectors in plane in geometrical form and in component form. Understand and use the dot product for problems solved.

Students are able to find distance between two points in plane; to write the equation of the straight-line in plane and space and the equation of the plane.

Students understand the basic concepts and rules of mathematical analysis, to solve standard problems (finding function limit, derivate functions, integrate functions, construct function graphics).

Students are able to calculate a probability of the random events; to understand the difference between discrete and continuous random variables; to calculate the numerical characteristics (mathematical expectation, variance, etc.). Students know the most important probability distributions of the random variables. Students are able to understand the basic concept of sampling and processing of statistical data.

Students can apply the acquired mathematical knowledge to solve practical problems.

Students are able to develop algorithms and create simple computer programs using the programming language *Python* to solve mathematical problems. Students are able to write a well-structured, readable and properly commented source code; to detect and correct errors in the source code; students are able to find an appropriate solution of the problem, explain and justify it.

Students are able to work independently with literature and Internet resources.

Students are able to write bachelor paper in accordance with the methodological instructions of the faculty; use spreadsheets.

Organization mode of students individual assignment

Regular learning using lecture materials, internet resources, software help systems. Individual assignment completion. Weekly teacher consultations.

Preparations for the tests and exams

Evaluation of study results

Final evaluation includes:

1) Regular and independent work during the semester (50%) that includes:
    - active participation in seminars, practical classes and lectures (5%);
    - practical assignments/home works (20%)
    - tests (25%)
2) Final exam (50%)

Course outline

No. of the class/topic	Title of the topic	Type of class (lectures, seminars, practical classes, laboratory work), amount of academic hours
1.	**Introduction to programming with *Python***	8 h
	Characteristics of the programming language *Python*. *Python* environment. Creating a *Python* program (script). Variables and data types (integers and floats, large integers, and others). *Python* memory management. Counting systems. Arithmetic operators. Input and output. *Python* modules. Module *math*.	Lectures – 2 h Practical classes – 6 h
2.	**Basic concept of Set Theory and Mathematical Logic**	14 h
	Concept of a set. Description of the set. Operations with sets. Proposition. Logical operations. Truth tables. Tautologies. Logical Equivalence. Normal forms. Principle of duality. The predicate. Logical operations with predicates. Relation and logic operations. Loops and branching constructions. Sequential Data Types (List, Set, Tuple). Various actions with Sequential Data Types.	Lectures – 3 h Seminars – 3 h Practical classes – 8 h
3.	**Matrices. Determinants. Linear Systems**	24 h
	Introduction to Matrices and Linear Systems. Matrix types. Basic operation of matrix (adding, subtracting, multiply of a matrix by a constant). Multiplication of matrices. Algebraic properties of matrix operations. Determinants of a matrix of order 2 and of order 3. Properties of the determinant. Minors and cofactors. Determinants of matrices of higher order. General formula for the determinant.	Lectures – 6 h

	Cramer`s Rule to solve the linear systems. Invertible matrix. Finding the inverse A^{-1} of the invertible $n \times n$ matrix A . Matrix equations. Solving the linear system by finding the inverse of the coefficient matrix. Gaussian elimination. Solving the linear systems by Gaussian elimination. The rank of a matrix. Kronecker-Capelli's theorem. Homogeneous systems. Definition of functions. Two-dimensional array as matrix representation. Basic operations on arrays - matrix operations. Solving linear equation systems. *Python* extension module *NumPy*.	Seminars – 6 h Practical classes – 12 h
4.	**Vektor Algebra**	12 h
	Definition of a vector. Types of vectors. Operations on vectors (adding, subtracting, multiply by a scalar). Properties of these operations. Projection of a vector. Vectors in two- and three- dimensional Cartesian coordinates. The component of a vector. Operations on vectors in component form. The dot product. Properties of the dot product. Array as vector representation. Operations on vectors. *Python* extension module *NumPy*.	Lectures – 3 h Seminars – 3 h Practical classes – 6 h
5.	**Coordinate Systems in plane. Line in plane. Lines and surfaces in space**	22 h
	Cartesian coordinate system. Polar coordinate system. Line in the plane and its equation; polar equations. Parametric equations of line. Straight-line in plane, its equation and direction coefficient. Angle between straight lines. General straight-line equation and its line segment equation. Intersection of straight lines, conditions of parallelism and congruence. Normal equation of straight line. Basic concept of the conic sections: the ellipse, the hyperbola, the parabola (general form and standard form). Equation of a plane in space. Equation of a straight line in the space, its canonical and parametric forms. Basic concept of the second-order surfaces (ellipsoid; one-sheet and two-sheet hyperboloid; second-order cones and cylinders; elliptic and hyperbolic parabolic). Data visualization. Graphic design. *Python* module *Turtle*. *Python* extension module *Matplotlib*.	Lectures – 5 h Semināri – 5 h Practical classes – 12 h
6.	**Introduction to Mathematical Analysis**	22 h
	Basic concept of a limit of function. The algebraic properties	Lectures – 8 h Seminars – 8 h

	of limits. One-sided limits. Infinite limits. The problems with indeterminate forms. The derivative of a function, it’s geometric and mechanics interpretations. Continuity of a differentiable function. Differentiability of a sum, of a product and of a quotient. Differentiability of composite function (The Chain Rule). Derivatives of some Elementary Functions. Higher order derivatives. Curve sketching using differentiation. A primitive (antiderivative) of function. Indefinite integral. Properties of indefinite integrals. Basic integration formulas. The simplest integration substitutions. The problem of area and the definite integrals. Python extension modules NumPy and SymPy.	Practical classes - 6 h
7.	**Basic concepts of Probability Theory and Mathematical Statistic**	26 h
	Basic concepts of the probability theory. The random events and algebra of events. Definition of probability. Addition and multiplication laws of probabilities. Conditional probability. Total probability and Bayes’ formula. Bernoulli Trials. Random variable (definition and classification). Functions of a random variable (distribution and density functions). Discrete random variables. Expected value (mathematical expectation), variance and standard deviation of a discrete random variable; properties. The most important probability distributions of discreet random variables: uniform, hypergeometric, binomial, geometric, the Poisson distributions. Continuous random variable. Probability density function and distribution function. Expected value and variance. The most important probability distributions of continuous random variables (exponential, uniform, normal, t-distribution). Strong law of large numbers. Introduction to Statistic. Descriptive statistics (collecting and presentation of statistical data; cumulative sample distribution function). Inductive statistics (random sampling and sampling distributions). Reading information from a file (text file and .csv file). Plotting. *Python* module *statistics*. *Python* extension module *NumPy*.	Lectures – 8 st. Seminars – 6 h Practical classes - 12 h

Basic literature

1) Rosen K. *Discrete Mathematics and Its Applications 7th Edition*. - McGraw-Hill Education, 2011., 1072 p. (ISBN-13: 978-0073383095)
2) Saha A. *Doing Math with Python*. – No Starch Press, Inc. 2015., 264 p. (ISBN-13: 978-1-59327-640-9)

Supplementary literature

1) Devlin K. *Introduction to Mathematical Thinking*. - Keith Devlin, 2012., 102 p. (ISBN-13: 978-0615653631)
2) Fangohr H. *Introduction to Python for Computational Science and Engineering (A beginner's guide)* . - Faculty of Engineering and the Environment University of Southampton, 2015.
3) Janacek G. *Mathematics for Computer Scientists*. - GarethJ. Janacek, Mark Lemmon Close & Ventus Publishing APS – 2011. (ISBN 978-87-7681-426-7)
4) Stavely A. M. *Programming and Mathematical Thinking. A Gentle Introduction to Discrete Math Featuring Python*. - The New Mexico Tech Press, 2014., 246

Other sources of information

1) www.algebra.com
2) www.algebrahelp.com
3) www.bymath.com
4) Python course http://www.python-course.eu/index.php
5) https://www.python.org/
6) https://www.scipy.org/
7) Software user guides and built-in help systems.

Security of Information systems

(Informāciju sistēmu drošība)

Author	Sanita Meijere
Course Code	DatZ2013
Form of evaluation	Exam
Credit point (ECTS credit points)	2 (6 ETCS)
Prerequisites	n/a
Author	Sanita Meijere
Course group	General education study course

Course description

Create awareness of the significance of information security and continuous growing of risks. Explain key information security protection principles and mechanisms, global best practices in the field and Information Security Management Standard`s requirements. Information systems` security in the internal infrastructure and cloud if outsourced. Secure programming principles. Secure information systems` architecture.

Learning outcome

By completing the course students will be aware of key definitions of information security, necessity of information systems` protection and protection mechanisms. The same students will be aware of global best practices in organising information systems` security, its management and risk treatment.

Students independent work – study of the materials

Grading scheme 70% class participation, 30% final exam

Course schedule

Lectures	Topic	Type
1	Basics of information systems security	Lecture & seminar
2	Cybercrime	Lecture & seminar
3	Social engineering	Lecture & seminar
4	Secure internal IT infrastructure	Lecture & seminar
5	Secure IT infrastructure in cloud	Lecture & seminar
6	Policy and principles of secure programming	Lecture & seminar
7,8	Information systems` security management	Lecture & seminar

Literature:

- **Rules and regulations:** Law on Physical person data protection, Criminal Law, Information security law.
- **Standards:** ISO ISO 27 001 + Annex A
- **Web:** CERT, ISACA, MIT, GARTNER, IDC

Software engineering
(Programmatūras inženierija I un II)

Author	Dr.sc.ing., doc. Raita Rollande, mg.sc.comp. lekt. Karina Šķirmante
LAIS course code	DatZ1029 and DatZ2016
Form of evaluation	Test and exam(After 2nd semester - test, after 3rd - exam)
Academic credit points (ECTS credit points)	6 (9 ECTS)
The total number of contact lessons	48
The number of lectures	12
The number of practical classes	36
Prerequisites	JAVA Programming, Data bases
Part of the study programme	Industry study courses

Study course objective

The objective of the course is to introduce students to design theoretical aspects of information systems analysis and design, and to develop practical skills in information systems development.

Study results

Having acquired the study course, a student:

- Is able to understand the role of design in the development of information systems.
- Is able to understand the software development stages, models and works to be executed in each phase, phase deliverables and documentation.
- Is able to choose the information systems development model based on the characteristics of the developed system.
- Is able to apply the information system design standards, create the concept description, software requirements specifications description, and software design description documentation.
- Is able to prepare a user manual of IS
- Is able to plan IS development project, to predict the run time of the tasks and its capacity, to plan individual tasks and control its, to take a part in discussions about project progress
- Is able to plan the IS project, to develop programming guidelines, to read and understand the requirements of the IS specifications, to understand the standards of the IS design outlines, to prepare and describe an architecture of the software, to analyse a different technical solutions and to choose the most suitable, to create a conceptual and physical data model, to create a relational model.
- Is able to design, construct and describe algorithms

- Is able to design user interfaces
- Is able to prepare documentation of the design project
- Is able to program and write a code based on the design documentations and guidelines
- Is able to write a code, to read and understand software design outlines
- Is able to analyse the input and output data of the program
- Is able to represent the project. Is able to present the project for large audience.

Organization mode of students' individual work

The independent work of students include:

- Teamwork. Based on the proposed problem, students in groups develop: a description of the concept, software requirements specification, software design description of the system, design prototypes, carry out testing. Preparation for the exam.

Evaluation of study results

- Team work (100%)

Study course outline

No.	Title of the topic
1.	General conception of information system analysis and design. Types of information systems. Work characteristics of information system analysts, consultants, and experts
2.	With the development of information systems related to professional standards.
3.	Information system architecture. Information system development. Information system life cycle models Lecture (waterfall life cycle model, shell model; rapid prototyping model, etc.). Agile modeling
4.	Students division into teams for system development. Preperation of the development enviroment.
5.	System requirements collection methods. Interviews. Interviews organization. Questions formation. Interview process. Summing up and analysis of acquired data
6.	Interview planning. Meeting with system contractors. Interviewing contractors. Analysis of the results of the interview.
7.	Agreements. Types of agreements. Information system documentation standards. Software development standards. Software engineering standards. Necessary documentation for software usage and maintenance. Information systems documentation - concept description. Content of the concept description
8.	Project Planning. Sprint Planning
9.	The use of diagrams in information system analysis and design. Organization diagrams, entity relationship diagrams, data flow diagrams, business process diagrams.

10.	Meeting with system contractors. Clarification of issues related with concept description development. The use of diagrams in information system analysis and design
11.	System requirements analysis. Prototypes. The role of prototype in information system design
12.	The 1st Sprint Review.
13.	Information systems documentation - concept description. Content of the concept description.
14.	Concept description presentation and delivery. Nest sprint Planning.
15.	Information systems documentation - software requirements specification description. Content of the software requirements specification description
16.	Meeting with system contractors. Clarification of system requirements
17.	Project Planning
18.	The 2nd Sprint Planning. The 2nd Sprint Realization
19.	The 2nd Sprint Realization
20.	The 2nd Sprint Documentation
21.	The 2nd Sprint Review
22.	The 2nd Sprint Retrsospective and the 3rd Sprint Planning
23.	Information System user guide. User guide for the programmer and the end user
24.	The 3rd Sprint Realization
25.	The 3rd Sprint Documentation
26.	The 3rd Sprint Review
27.	The 3rd Sprint Retrsospective and the 4th Sprint Planning
28.	Quality assurance. The efficient usage of resources. Cost control. Customer satisfaction
29.	The 4th Sprint Realization
30.	The 4th Sprint Documentation
31.	The 4th Sprint Review
32.	The 4th Sprint Retrsospective and the 5th Sprint Planning
33.	The introduction and maintenance of the information system. Maintenance of the information system and service contract.
34.	The 5th Sprint Realization
35.	The 5th Sprint Documentation
36.	The 5th Sprint Review and Retrospective.
37.	Final Presentation

Study course schedule

No. of the class	Title of the topic	Type of class (lectures, seminars, practical classes, laboratory work), amount of academic hours
The second semester		

1.	General conception of information system analysis and design. Types of information systems. Work characteristics of information system analysts, consultants, and experts	Lecture
1.	With the development of information systems related to professional standards.	Practical classes
2.	Information system architecture. Information system development. Information system life cycle models Lecture (waterfall life cycle model, shell model; rapid prototyping model, etc.). Agile modeling	Lecture
2.	Students division into teams for system development. Preperation of the development enviroment.	Practical classes
3.	System requirements collection methods. Interviews. Interviews organization. Questions formation. Interview process. Summing up and analysis of acquired data	Lecture
3.	Interview planning. Meeting with system contractors. Interviewing contractors. Analysis of the results of the interview.	Practical classes
4.	Agreements. Types of agreements. Information system documentation standards. Software development standards. Software engineering standards. Necessary documentation for software usage and maintenance. Information systems documentation - concept description. Content of the concept description	Lecture
4.	Project Planning. Sprint Planning	Practical classes
5.	The use of diagrams in information system analysis and design. Organization diagrams, entity relationship diagrams, data flow diagrams, business process diagrams.	Lecture
5.	Meeting with system contractors. Clarification of issues related with concept description development. The use of diagrams in information system analysis and design	Practical classes
6.	System requirements analysis. Prototypes. The role of prototype in information system design	Lecture
6.	The 1st Sprint Review.	Practical classes
7.	Information systems documentation - concept description. Content of the concept description.	Lecture
7.	Concept description presentation and delivery. Nest sprint Planning.	Practical classes
8.	Information systems documentation - software requirements specification description. Content of the software requirements specification description	Lecture
8.	Meeting with system contractors. Clarification of system requirements	Practical classes

The third semester		
1.	Project Planning	Lecture
1.	The 2nd Sprint Planning. The 2nd Sprint Realization	Practical classes
2.	The 2nd Sprint Realization	Practical classes
2.	The 2nd Sprint Documentation	Practical classes
3.	The 2nd Sprint Realization	Practical classes
3.	The 2nd Sprint Realization	Practical classes
4.	The 2nd Sprint Review	Practical classes
4.	The 2nd Sprint Retrsospective and the 3rd Sprint Planning	Practical classes
5.	Information System user guide. User guide for the programmer and the end user	Lecture
5.	The 3rd Sprint Realization	Practical classes
6.	The 3rd Sprint Documentation	Practical classes
6.	The 3rd Sprint Realization	Practical classes
7.	The 3rd Sprint Realization	Practical classes
7.	The 3rd Sprint Realization	Practical classes
8.	The 3rd Sprint Review	Practical classes
8.	The 3rd Sprint Retrsospective and the 4th Sprint Planning	Practical classes
9.	Quality assurance. The efficient usage of resources. Cost control. Customer satisfaction	Lecture
9.	The 4th Sprint Realization	Practical classes
10.	The 4th Sprint Documentation	Practical classes
10.	The 4th Sprint Realization	Practical classes
11.	The 4th Sprint Realization	Practical classes
11.	The 4th Sprint Realization	Practical classes
12.	The 4th Sprint Review	Practical classes
12.	The 4th Sprint Retrsospective and the 5th Sprint Planning	Practical classes
13.	The introduction and maintenance of the information system. Maintenance of the information system and service contract.	Lecture
13.	The 5th Sprint Realization	Practical classes
14.	The 5th Sprint Documentation	Practical classes
14.	The 5th Sprint Realization	Practical classes
15.	The 5th Sprint Realization	Practical classes
15.	The 5th Sprint Realization	Practical classes
16.	The 5th Sprint Review and Retrospective.	Practical classes
16.	Final Presentation	Practical classes

Basic literature

1. Bell D. Software Engineering for Students. – Pearson Education (US), 2005.
2. J. L. Whitten, L.D. Bentley,K. C. Dittman, Systems analysis and design methods, McGraw-Hill, 2000

3. Jeffrey A. Hoffer, Joey F. George, Joseph S. Valacich, Modern Systems Analysis and Design, 4/E, Prentice Hall, 2001
4. Joseph S. Valacich, Joey F. George, Jeffrey A. Hoffer, Essentials of Systems Analysis and Design, 2/E, Prentice Hall, 2003
5. Kenneth E. Kendall Julie E. Kendall, Systems Analysis and Design, 6/E, Prentice Hall, 2004
6. Kenneth E. Kendall Julie E. Kendall, Systems Analysis and Design, 5/E, Prentice Hall, 2001
7. Shouhong Wang, Hai Wang, Information Systems Analysis and Design, Universal-Publishers, 2012. http://www.bookpump.com/upb/pdf-b/2330754b.pdf
8. Sommerville I. Software Engineering. – 8th Edition. – Addison-Wesley Pub. Comp, 2006.
9. Van Vliet H. Software Engineering. Principles and Practice. – John Wiley & Sons, 2008. http://sunset.usc.edu/available_tools/index.html

Supplementary literature

Other sources of information

UNIX like system administration

(UNIX sistēmu administrēšana)

Author	Lector, Mg.oec. , Gints Neimanis
LAIS course code	DatZ2019
Form of evaluation	Exam
Academic credit points (ECTS credit points)	2 (3 ECTS)
The total number of contact lessons	16
The number of lectures	8
The number of practical classes	8
Prerequisites	
Part of the study programme	Industry study courses

Study course objective

The study course objective is to provide knowledge of the fundamentals of the UNIX like operating systems and give skills to work with their tools, to configure and to manage UNIX like systems.

Study results

Having acquired the study course, a student:

- Is able to describe Unix un Linux like systems history.
- Is able to describe the architecture and environment of Linux like systems.
- Is able to identify different system services.
- Is able to use command line tools to manage Linux like systems.
- Is able to monitor and troubleshoot the system and services.

Organization mode of students' individual work

The individual work of students include:

- a regular learning of the course by using study literature, internet resources,
- a regular learning of the course by using lecture materials,
- hands-on labs
- course paper development,
- preparations for the labs, tests and exams.

Evaluation of study results

Final mark depends on Exam (100%). It is mantadory to pass all labs.

Study course outline

No.	Title of the topic
1.	Unix like operating system hisotry.
2.	Introduction to command line tools.
3.	Various Package management systems.
4.	Management of users, groups and access rights.
5.	Network configuration and services.
6.	Management of system runlevels, boot order.
7.	Automatization of tasks, backup management.
8.	System resource and proceses monitoring un management.
9.	Managing of system daemons.

Study course schedule

No. of the class	Title of the topic	Type of class (lectures, seminars, practical classes, laboratory work), amount of academic hours
1.	Unix like operating system hisotry.	Lecture 2h
2.	Introduction to command line tools.	Lecture 2h, practical classes 2h
3.	Package management systems.	Lecture 2h, practical classes 2h
4.	Management of users, groups and access rights.	Lecture 2h, practical classes 2h
5.	Network configuration and services.	Lecture 2h, practical classes 2h
6.	Management of system runlevels, boot order.	Lecture 2h, practical classes 2h
7.	Automatization of tasks, backup management.	Lecture 2h, practical classes 2h
8.	System resource and proceses monitoring un management.	Lecture 2h, practical classes 2h
9.	Managing of system daemons.	Practical classes 2h

Basic literature

- Linux Administration: A Beginner’s Guide, Seventh Edition 7th Edition by Wale Soyinka ISBN-13: 978-0071845366
- How Linux Works: What Every Superuser Should Know 2nd Edition by Brian Ward ISBN-13: 978-1593275679
- Essential System Administration: Tools and Techniques for Linux and Unix Administration, 3rd Edition 3rd Edition by Æleen Frisch ISBN-13: 978-0596003432

Supplementary literature

Other sources of information

Fundamentals of the Latvian Language

(Latviešu valodas pamati)

Author	Mg.philol., lecturer Sintija Ozoliņa
LAIS course code	New course
Form of evaluation	Test
Academic credit points (ECTS credit points)	2 (3 ECTS)
The total number of contact lessons	16
The number of lectures	-
The number of practical classes	16
Prerequisites	-
Part of the study programme	General education study courses

Study course objective

The study course objective is to develop written and spoken communication skills in the Latvian language.

Study results

Having acquired the study course, a student:

- Is capable of communicating in the Latvian language with or without a dictionary;
- Is able to introduce others to himself/herself, to use Latvian when shopping and in other daily activities, as well as is able to write simple messages in Latvian and to fill out questionnaires, and is able to form a dialogue.

Organization mode of students' individual work

The independent work of students include:

- a regular learning of the course by using lecture materials, study literature, internet resources,
- homework assignment completion,
- situation interpretations and group work during classes,
- preparations for the tests and exams.

Evaluation of study results

The end result is made of:

- Homework 10%
- Performance in classes 30%
- Exam 60%

Study course outline

No.	Title of the topic
1.	Pronunciation of the sounds in the Latvian language.
2.	Creation and building of vocabulary.
3.	Nouns and their categories.
4.	Numerals and their categories.
5.	Pronouns and their categories.
6.	Verbs and their categories.

.

Study course schedule

No. of the class	Title of the topic	Type of class (lectures, seminars, practical classes, laboratory work), amount of academic hours
1.	Introduction. The Latvian alphabet.	Lecture, 2 academic hours
2.	Pronunciation.	Practical class, 2 academic hours
3.	Greetings, forms of the verb “to be”.	Practical class, 2 academic hours
4.	Languages, nationalities.	Practical class, 2 academic hours
5.	Noun cases, singular and plural forms, genders.	Practical class, 2 academic hours
6.	Personal pronouns. Family.	Practical class, 2 academic hours
7.	Numerals.	Practical class, 2 academic hours
8.	Clock times.	Practical class, 2 academic hours
9.	The adverb. Locations, directions.	Practical class, 2 academic hours
10.	At the shop/cafe/restaurant.	Practical class, 2 academic hours
11.	The verb. Leisure time. Dative forms of nouns.	Practical class, 2 academic hours
12.	Education and work. Culture.	Practical class, 2 academic hours
13.	Lifestyle.	Practical class, 2 academic hours
14.	Years. Months. Days of the week. Daily schedule.	Practical class, 2 academic hours
15.	At the doctor’s office.	Practical class, 2 academic hours
16.	Revision.	Practical class, 2 academic hours

.Basic literature

1.C.Moseley. Colloquial Latvian. Routledge: 1996.
2. I. Auziņa u.c. Māci un mācies. Latviešu valodas aģentūra: 2015.
3. Kursa vadītāja izstrādātie materiāli tēmu un gramatikas apguvei.

Supplementary literature

Budviķe, B.Šiliņa, R.Vizule. *Palīgā! Mācībgrāmata pieaugušajiem*. Zvaigzne ABC, Rīga: 1998

Other sources of information

Internet resources:
1.www.sazinastilts.lv Valodas apguve
2.http://maciunmacies.valoda.lv/valodas-apguve/e-nodarbibas#tab2 E-nodarbības
3. http://www.valoda.lv/Papildus_Materiali/eapmaciba2/default.htm Papildmateriāli I
4. http://www.valoda.lv/Papildus_Materiali/LVA-2-dala/default.htm Papildmateriāli II

Internet and computer networks
(Interneta un datortīkla tehnoloģijas)

Author	Mag.oec. Lecturer Gints Neimanis
LAIS course code	DatZ2018
Form of evaluation	Exam
Academic credit points (ECTS credit points)	2 (3 ECTS)
The total number of contact lessons	16
The number of lectures	8
The number of practical classes	8
Prerequisites	
Part of the study programme	Industry study courses

Study course objective

Provide knowledge of local and global computer networks and their technologies. Get skills to configure, use and troubleshoot computer networks.

Study results

Having acquired the study course, a student:

- Is able to explain OSI and TCP/IP model.
- Is able to describe network functions on various OSI levels and Internet protocols.
- Is able to configure network settings on computer systems.
- Is ables to troubleshoot network errors.

Organization mode of students' individual work

The individual work of students include:

- studying Cisco NetAcademy courses and other information sources,
- repeating hands on labs

Evaluation of study results

Final mark depends on laboratory works, tests and final exam. It is mandatory to pass all laboratory works and tests before exam.

Study course outline

No.	Title of the topic
1.	Introduction to computer networks
2.	OSI network model
3.	TCP/IP model. IP addressing, routing and network protocols.

4.	Configuring and troubleshooting network settings in varioous operating systems.

Study course schedule

No. of the class	Title of the topic	Type of class (lectures, seminars, practical classes, laboratory work), amount of academic hours
1.	Introduction to computer networks	Lecture 2h, seminar 2h
2.	OSI network model	Lecture 2h, seminar 2h
3.	OSI Physical and Data link layer . CSMA/CD and CSMA/CA. Ethernet (IEEE 802.3) and WLAN (IEEE 802.11).	Lecture 2h, practical classes 2h
4.	TCP/IP and IP addressing	Lecture 2h, practical classes 2h
5.	Internet and routing.	Lecture 2h, practical classes 2h
6.	Configuring network settings in various operating systems.	Lecture 2h, practical classes 2h
7.	Application level protocols at TCP/IP stack.	Lecture 2h, practical classes 2h
8.	Network troubleshooting	Lecture 2h, practical classes 2h

Basic literature

1. Lammle, Todd. CompTia Network + study guide : exam N10-006 / Todd Lammle. - 3rd ed. - Indianapolis, IN : John Wiley and Sons, 2015. - xxv, 906 p. : il., tables. - Index: p. 865.- 906.
ISBN 9781119021247.
2. Cisco NetAcademy course materials

Supplementary literature

http://www.zytrax.com/books

Other sources of information

English for Programming

(Angļu valoda)

Author	Dr.paed. Vita Balama, Assist Prof
Course Code	Valo1064
Form of evaluation	Exam
Credit point (ECTS credit points)	2 (3 ECTS)
Prerequisites	Secondary school level knowledge of English language (B2 level according to European Language Framework)
Course group	General education study course
Objective	To provide knowledge of business communication (oral and written), introduce comprehension of texts in the given field of IT programming.
Learning outcomes	Students are able to read, comprehend and analyze the text in English on programming topics. Individually prepare the presentations on given / chosen topics, and present them publically, answer the questions on certain issues, defend their idea, viewpoint, opinion. Students are able to correspond for business, academic and personal issues.

Organization mode of students individual assignment

During lectures, students perform individual listening, reading and writing tasks. Besides pair- and group- work is practiced during the classes. Presentations are prepared both individually and in groups. Individual tasks are assigned for text reading and analysis. The use of internet resources for searching the information is highly advised.

Evaluation of learning outcomes

Successful evaluation of learning outcomes is given when the following requirements are met:

1) Course attendance is at least 75% of classes (i.e. all class assignments are completed successfully);
2) Successfully passed all written midterm tests (at least 2 in a term);
3) Prepared final complete report on the chosen topic and presented during the class publically to the group;
4) Successfully passed the final exam during the exam session.

Course outline

Week	Topic and subtopic	**Type** (lecture, seminar, laboratory work)
1	Written and Oral Communication	practical classes
2-3	Communications: Email and Newsgroups www Website Designer	practical classes
4	Graphics and Multimedia	practical classes
5-6	Presentation Design, Presentation Skills	practical classes
7-8-9	Business Correspondence	practical classes
10-11	Entrepreneurial Skills for Programming Writing Speaking	practical classes
12-13-14	Carriers in Programming Writing the Motivation Letter Filling in Application Form Designing CV	practical classes
15-16	Final Presentation on Chosen Topic	practical classes

Basic literature

1. Glendinning E. H., McEwan J. Oxford English for Information Technology,- OUP, 2001,
2. Glendinning E.H., McEwan J., Basic English for Computing,- OUP, 2000,
3. Brown C., Boeckner K. Oxford English for Computing,- OUP, 1999
4. Glendinning E. H., McEwan J. Oxford English for Electronics,- OUP, 1993,
5. Shirley Taylor, Model Business Letters, e-mails and Other Business Documents, Zvaigzne ABC, Rīga, 2006.

Supplementary literature

6. Angļu-latviešu-krievu Informātikas Vārdnīca.- Rīga, Avots, 2001.- 657 lpp.
7. Personālie datori, angļu-latviešu-krievu skaidrojošā vārdnīca, Rīga, a/s Dati, 1998.- 255 lpp
8. Corey Sandler, 1001 Business Letters for All Occasions: From Interoffice Memos and Employee Evaluations to Company Policies and Business Invitations - Templates for Every Situation, Adams Media; Pap/Cdr edition, 2008
9. Robert W. Bly, The Encyclopedia of Business Letters, Faxes, and Emails: Features Hundreds of Model Letters, Faxes, and E-mails to Give Your Business Writing the Attention It Deserves, Career Pr Inc, 2009.
10. PC Magazine
11. Windows magazine

Other source of information

IT Project Management

(IT projektu vadīšana)

Author	Mg.oec Uldis Kuplis
Course Code	Citi1041
Form of evaluation	Test
Credit point (ECTS credit points)	2 (3ECTS)
Prerequisites	None
Course group	General education study course

Objective

The course objective is to understand theoretical basics of IT projects management processes and practical case study.

The course content is focused on organizing and management of software development projects. It ensures students comprehension of working as a team, management processes inside a project and organizational activities during project lifecycle.

Learning outcomes

The course outcomes are:

- To understand basic concepts of IT project management
- To be able to plan a project and document it
- To create project organizational structure
- To understand project lifecycle and its processes
- To understand responsibility of project manager role
- To estimate project workload
- To understand project cost calculations and risk management
- To be able to choose the most appropriate software development methodology

Organization mode of students individual assignment

All study resources (presentations, tasks and links to additional reading texts) will be available in electronic format.

Evaluation of learning outcomes

Students will be graded on their work during a semester. The grade will be calculated on their contribution, quizzes, group work and final test.

Grading for the course is as follows:

Assignments	Points
Contribution in class discussions	20
Quizzes	20
Team presentation	30
Final test	30
Total	100

Minimum points are 40 in order to pass the course evaluation.

Course outline

Week	**Topic and subtopic**	**Type** (lecture, seminar, laboratory work)
1. (2x90min)	Project and management basic concepts	Lecture
	Types of software development projects and appropriate methodologies	Seminar
2. (2x90min)	Project planning, project team, structure and roles	Lecture
	Project risk and resources management	Seminar
3. (4x90min)	The most popular software development methodologies – waterfall, agile, etc.	Seminar
4. (2x90min)	Project control, delegating, project manager’s role, project closing	Lecture
	Case study and analysis	Practical work
5. (2x90min)	Project management tools	Seminar
	Final test	Test
6. (4x90min)	Final test review	Seminar
	Teamwork presentation	Practical work

Basic literature
Tridibesh Satpathy, “A Guide to the SCRUM Body of Knowledge (SBOK Giude)”, Scrum Study, 2016.

Supplementary literature
Lecturer prepared reading materials

Other source of information

Basics of IT industry rules & regulations & standards

(IT nozares tiesību pamati un standarti)

Author	Sanita Meijere
Course Code	Ties1002
Form of evaluation	Exam
Credit point (ECTS credit points)	2 (6 ETCS)
Prerequisites	n/a
Course group	General education study course

Course description

Explained basics of IPR, Copyright law, Patent Law, Law on physical persons` data protection, classification of cybercrime, Information security law. Explained quality management and Quality management standard ISO 9001. Explained Information security management standard ISO 27 001.

Learning outcome

By completing the course students will be aware of key aspects of IPR, its necessity and mechanisms and the most essential international and national rules and regulations in the field. Besides that, students will be aware of key quality and information security management tendencies and requirements based on the best practices.

Students independent work – study of the materials

Grading scheme 70% class participation, 30% final exam

Lectures	Topic	Type
1	IPR	Lecture & seminar
2	Copyright	Lecture & seminar
3	Patents	Lecture & seminar
4	Physical person data protection	Lecture & seminar
5	Cybercrime	Lecture & seminar
6	Information security	Lecture & seminar
7,8	Quality management	Lecture & seminar

Literature:

- **Rules and regulations:** Copyright law, Patent law, Law on Physical person data protection, Criminal Law, Information security law.
- **Standards:** ISO 9001, ISO 27 001
- **Web:** CERT, ISACA

Embedded systems

(Iegultās lietojumprogrammas)

Author	Mg. Sc. Ing., lektors, Gints Dreifogels
LAIS code	DatZ2021
Test form	Exam
Credit points (ECTS)	4 (6 ECTS)
Total number of contact lessons	32
Total number of lectures	10
Total number of workshops	22
Required knowledge to start the course	Not needed
Part of study programm	Industry study course

Study course aim

Study goal is to give theoretical and practical knowledge about applications of microcontrollers, including their architecture, technical characteristics, specific properties and programming skills using C and assembly languages. Using specific hardware and firmware development environment, provide possibility to defense their practical works and presenting results of coursework.

Study results

Student successfully mastering the study course:

- can program microcontrollers in C and assembly programming languages
- using specialized hardware and firmware development environment;
- can translate C structures to assembly instructions;
- can work with microcontroller peripheral devices;
- can develop embedded system firmware;
- can defense results of practical works;
- can present results of coursework;
- can characterise microcontroller architecture main buildings blocks.

Student independent work organization type

Student individual work includes:

- regular study of course content, using lecture materials, educational literature, internet resources;
- preparing for lecture tests;
- development of practical work;
- development of coursework.

Evaluation of study results

Final mark depends on:

- Practical works 50%
- Coursework 40%
- Tests 10%

Study course content

No.	Topic
1.	Introduction. Applications of microcontrollers and their architecture. Getting familiar with hardware platform ATmega328P Xplained Mini in usage for practical works.
2.	Basics of digital electronics: • Boolean algebra; • logic gates: NOT, AND, OR, NAND, NOR, XOR, XNOR; • half and full adder, multiplexer, demultiplexer, encoder, decoder; • sequential logic: trigers, registers; • ALU; • practical examples using computer simulation programm Logism.
3.	Microcontroller ATmega328P architecture. Instruction Set. Directives. Getting familiar with Atmel Studio 7 in usage for practical works.
4.	Introduction: Programming basics for microcontroller: • C and assembly programming languages; • Elementary Input and Output Part 1; • practical examples using C and assembly; • macro definitions, bit masks.
5.	Programming basics for microcontroller: • Elementary Input and Output Part 2; • practical examples using C and assembly; • translation from C structures to assembly instructions; • macro definitons, bit masks.
6.	Memory organization. Memory addresing modes. Stack.
7.	Power management and sleep modes. Interrupts.
8.	Peripheral devices: taimers/counters, (watchdog)
9.	Peripheral devices: analog-to-digital converter (ADC)
10.	Peripheral devices: USART, SPI, I2C

Calendar plan of study course

Lesson number	Topic	Lesson type (lectures, workshops, practical lessons, laboratory works), total number of academic lessons
1.	Introduction. Applications of microcontrollers and their architecture.	Lecture, 2
2.	Basics of digital electronics Part 1.	Lecture, 2
3.	Basics of digital electronics Part 2.	Lecture, 2

4.	Microcontroller ATmega328P architecture.	Lecture, 2
5.	Introduction: Programming basics for microcontroller	Lecture, 2
6.	Programming basics for microcontroller	Lecture. 2
7., 8.	Practical Work 1: Elementary Input and Output Part 1	Workshop, 2
9. 10.	Practical Work 2: Elementary Input and Output Part 2	Workshop, 2
11., 12., 13.	Defending of PW 1 and PW 2	Workshop, 6
14.	Memory organization	Lecture, 2
15., 16.	Practical Work 3: Memory organization	Workshop, 2
17.	Power management and sleep modes. Interrupts.	Lecture, 2
18., 19.	Practical Work 4: Interrupts and sleep modes	Workshop, 2
20., 21., 22.	Defending of PW 3 and PW 4	Workshop, 6
23.	Peripheral devices: taimers/counters, (watchdog)	Lecture, 2
24., 25.	Practical Work 5: timers/counters	Workshop, 2
26.	Peripheral devices: USART, SPI, I2C	Lecture, 2
27., 28.	Practical Work 6: Serial Communication - UART	Workshop, 2
29., 30., 31.	Defending of PW 5 and PW 6	Workshop, 6
32.	Presentation of course work ideas	Workshop, 2

Basic literature

1.S. F. Barrett, D. J. Pack. Atmel AVR Microcontroller Primer: Programming and Interfacing. Morgan & Claypool Publishers, 2008. ISBN-10: 1598295411, ISBN-13: 978-1598295412.

2. R. Barnett, S. Cox, L. O'Cull. Embedded C Programming and the Atmel AVR. 2nd Revised edition. Delmar Cengage Learning, 2006. ISBN-10: 1418039594, ISBN-13: 978-1418039592.

Additional literature

1.D. V. Gadre. Programming and Customizing the AVR Microcontroller. Tab Electronics; 2000. ISBN-10: 007134666X, ISBN-13: 978-0071346665.

2.Muhammad Ali Mazidi, Sarmad Naimi, Sepehr Naimi. AVR Microcontroller and Embedded Systems: Using Assembly and C. Prentice Hall; 2010. ISBN-10: 0138003319, ISBN-13: 978-0138003319.

Other information sources

1. 8-bit AVR Microcontrollers ATmega328/P, DATASHEET COMPLETE, Atmel Corporation, 2016
2. AVR Microcontrollers AVR Instruction Set Manual, Atmel Corporation, 2016

3. ATmega328P Xplained Mini, Microchip Technology Inc., 2017, ISBN: 978-1-5224-2082-8

4. http://www.microchip.com/wwwproducts/en/atmega328p
5.http://www.microchip.com/developmenttools/productdetails.aspx?partno=atmega328p-xmini
6. http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_instruction_list.
7. https://www.microchip.com/
8. http://www.avr-tutorials.com/
9. http://www.avrfreaks.net/
10. https://www.tutorialspoint.com/assembly_programming/assembly_macros.htm
11. http://extremeelectronics.co.in/category/electronics/
12. https://electrosome.com/category/tutorials/atmel-avr/
13. https://www.freertos.org/

Database
(Datu bāzes I un II)

Author	**G. Neimanis**
Course Code	DatZ1025 and DatZ1026
Form of evaluation	Exam
Credit point (ECTS credit points)	4 (6 ECTS)
Prerequisites	Basic computer skills
Course group	Industry study course

Objective

The aim of this course ir to provide knowledge about database management systems, their history, architecture, usability, exploitation as well as to acquire practical skills in creation of databases through practical project.

Learning outcomes

Understanding of database types, objects. Skills to normalize data, use SQL for data manipulation and data definition. Understanding and skills to use views, transactions, stored procedures and triggers.

Organization mode of students individual assignment
Regular studies of course material, literature and online resources; homework assignments and development of course project; consultations with lecturer.

Evaluation of learning outcomes
Course project – 50%.
Final exam - 50%;

Course outline

Week	**Topic and subtopic**	**Type** (lecture, seminar, laboratory work)
1 - 2	DBMS history, types.	Lecture and seminars
3 - 4	Database design and database objects (tables, relations, keys, indexes).	Lecture and seminars
5 - 6	Data normalization	Lecture and seminars

7 - 8	SQL. DML usage	Lecture and seminars
9 - 10	SQL. DDL and account management commands	Lecture and seminars
11 - 12	Views and transactions. ACID	Lecture and seminars
13 - 14	Stored procedures, triggers	Lecture and seminars
15 - 16	Course project presentations	Seminars

Basic literature

1. Oracle Academy – Data base design, Database Programming with SQL.
2. MySQL Documentation: MySQL Reference Manuals
 http://dev.mysql.com/doc/

Supplementary literature

Documentation PostgreSQL
 - www.postgresql.org/docs/

Other source of information -

Business and Economics
(Ekonomika un komercdarbība)

Author:	**M.Soc.Sc. Viesturs Zeps**
Code	Ekon1017
Type of Test:	Exam (including practical assignment)
Credit Points (ECTS):	2 (3 ECTS)
Pre-conditions to participate:	Proficient understanding of English
Group	General education study courses

Aim of the Course
Main aim of the course is to provide knowledge and information to students on creation of modern high tech companies and management of innovation projects. The course enables students to understand key principles of business modelling, experiment and test business related hypothesis, develop their innovation projects (or start-up projects), prepare pitch-decks and present them.

Course covers the main principles of *Lean Start-Up*, explains the typical business and product development cycles, and provides information on most suitable support incentives and fundraising (venture capital, grant schemes, loans and mixed incentives)

Expected Results of the studies
It is expected that students (individually or in groups) are able to develop their own business model, prepare financial projects of the project (start-up) that is based on real data, prepare and present pitch deck for investors or any other financial scheme.

Organization of individual assignments
Course is organized in lectures and seminars, a set of home reading and some case study analysis are required by students. Individual assignments generally are based on development of the business model as well as ability to present it.

Evaulation of the resuls
Total evaluation of course are combined as follows:

50% Written Assignment
10% Participation in Lectures
40% Presentation of Pitch Deck

Contents of the Course

Week	**Subject and Sub-Subject**	**Type** (lectures, seminars, individual takss, laboratory activities)

1	Setting up of a company, typical cycles of a company development, general legal aspects on setting up a company. Knowledge, skills, team, funding and other fundamentals to start commercial activities.	Lecture / Seminar (4 lectures)
2	Development of Business Model Canvas (Profile of a company): - Business model Canvas - Analysis of other business models - Fundamentals of Pitch Deck	Lecture / Seminar (4 lectures)
3	Development of Business model canvas - Development and analysis of BM Canvas - Development and analysis of Financial Plan - Analysis of Competition, Segmentation etc	Lecture / Practical Assignment (4 lectures)
4	Fund rising - Development of MVPs - Analysis of costs - Presentations of a Pitch Deck	Seminar (4 lectures)

Basic Literature

Ries, Eric., *Lean Start-up, 2011*
Metrick, Andrew., *Venture capital and the finance of innovation*, 2007
Osterwalder, A., *Business Model Generation*, 2010

Additional Literature

Dogson,M., Gann, D.M., Phillips, N., *The Oxford Handbook of Innovation Management*, 2013.
Blank, S., Dorf, B., *The Startup Owner's Manual, The Step-by-Step Guide for Building a Great Company,* 2012

Other sources of information:

http://www.businessmodelgeneration.com/
https://em.gov.lv/lv/es_fondi/
http://cfla.gov.lv/lv/es-fondi-2014-2020/vispariga-informacija
http://www.altum.lv/ [http://atbalstaprogrammas.lv/]
http://liaa.gov.lv/lv/fondi/es-fondi
http://www.izm.gov.lv/lv/es-strukturfondi/2014-2020
http://www.esfondi.izm.gov.lv/
http://www.esfondi.lv/es-fondi-2014---2020

Communication and Professional Ethics
(Saskarsme un profesionālā ētika)

Author	doc. Dr. philol. Guntars Dreijers
Course Code	KomZ1005
Form of evaluation	examination
Credit point (ECTS credit points)	2 (3ECTS)
Prerequisites	none
Course group	General education study course

Objective
To provide practical skills in communication and professional ethics needed in the work with customers, colleagues, and people from the academic environment – students and the teaching staff.

Learning outcomes
Students will learn about the basic concepts, models and their applications in communication and professional ethics; they will also be able to explain, analyze the practical applicability of the concepts and the models.

Organization mode of students individual assignment
Learning of concepts,home reading, drawing up and preparing presentations, preparing for the final examination

Evaluation of learning outcomes
Students will be provided with a feedback (oral and written assessment) about acquired concepts in communication and professional ethics, about discussions and negotiations, about oral presentations, and an assessment of the written examination

Course outline

Week	**Topic and subtopic**	**Type** (lecture, seminar, laboratory work)
1-2	Components of language in communication. The role of language in communication. Lingustic and extralinguistic components of communication.	1lecture, 1 seminar
3-4	Ethical considerations in communication. The concept of ethics. Communication ethics.	1lecture, 1 seminar
5-6	Psycholinguistic aspects of communication. Personality, character, behaviour in communication.	1lecture, 1 seminar

7-8	Discussion, confrontation. The role of *word*(s) in a communication. Discussion, confrontatin, objections, pretexts and excuses, manipulation.	1lecture, 1 seminar
9-10	Business negotiations and business relations. Persuasion, influencing, motivation, forming and expressing an opinion.	2 seminars
11-12	Speech. Prepared and *ex promptu* speech.	1 lecture, 1 seminar
13-14	Attitude, coherence, calibrating. Attitude to speech situation, participants. Congruence skill. Calibrating or the skill to observe and notice differences in human behaviour.	2 seminars
15-16	Oral communication in presentations. Preparing the final term task.	2 seminars

Basic literature
Apsalons, Edmunds (2013). Komunikācijas competence. Kā saprasties un veidot attiecības. Rīga : Zvaigzne ABC.

Supplementary literature
Brēdemeiers, Karstens (2010). *Vārda spēks*. Rīga : Zvaigzne ABC.

Apsalons, Edmunds (2011). *Valodas lietojuma loģika.* Rīga : Zvaigzne ABC.
Other source of information
youtube.com (speeches, presentations, discussions, confrontations, communication models)

Software Design Patterns

(Programmatūras izstrādes šabloni I un II)

Author	Ralfs Lasmanis
LAIS course code	DatZ2014 and DatZ2015
Form of evaluation	Exam
Academic credit points (ECTS credit points)	6 ECTS
The total number of contact lessons	32
The number of lectures	16
The number of practical classes	16
Prerequisites	Basics of JAVA programming language.
Part of the study programme	Industry study course

Study course objective

The study course objective is to learn object-oriented programming (OOP) design principles and various OOP design patterns in Java programming language, to be able to solve various design problems by using established best practice approaches.

Study results

Having acquired the study course, a student:

- Is able to articulate the challenges involved in building complex software systems and the need for a systematic engineering approach to cope with these challenges.
- Is able to describe how patterns can be used to create reusable abstractions in the design of complex systems.
- Is able to compare and contrast how generic programming and pure object-orientated programming are used in pattern design.
- Is able to differentiate between idioms, design patterns, and architectural patterns.
- Is able to explain the consequences of using anti-patterns in software design.
- Is able to describe the design, use, and consequences of the Gang of Four and GRASP design patterns.
- Is able to describe the relationships and dependencies between GRASP patterns, Gang of Four Patterns, and the principles of object-oriented design.
- Is able to identify and use available resources to research various patterns.

- Is capable of applying OOP principles, GRASP, and Gang of Four design patterns to overcome real design challenges.
- Is capable of using UML diagrams for documenting the design of software.
- Is capable of identifying the use of various patterns in Java SE.
- Is capable of creating a clean and maintainable object-oriented designs by using best industry practices.

Organization mode of students' individual work

The independent work of students includes:

- a regular learning of the course by using lecture materials, study literature, internet resources,
- homework assignment completion,
- course project development,
- preparations for the tests and exams.

Evaluation of study results

The end result is made of:

- Performance in classes 10%
- Practical classes and homework 15%
- Tests 25%
- Course project 50%

Or:

- Exam 100%

Study course outline

No.	Title of the topic
1.	Software development challenges and complexities.
2.	UML diagrams, and their use in describing object-oriented design.
3.	Principles of object-oriented design.
4.	GRASP design patterns.
5.	Gang of Four design patterns.
6.	Clean code, code maintainability, and code refactoring techniques.

Study course schedule

No. of the class	Title of the topic	Type of class (lectures, seminars, practical classes, laboratory work), number of academic hours

1.	Course introduction. Software development challenges and complexities. Architecture vs design. Patterns. Design patterns.	1x lecture, 1x practical class
2.	UML diagrams.	1x lecture, 1x practical class
3.	Principles of object-oriented design.	2x lectures 2x practical classes
4.	GRASP design patterns.	2x lectures, 2x practical classes
5.	Gang of Four (GoF) design patterns, their categorization. Creational GoF patterns.	2x lectures, 2x practical classes
6.	Structural GoF patterns.	3x lectures, 3x practical classes
7.	Behavioral GoF patterns.	3x lectures, 3x practical classes
8.	Clean code, code maintainability. Code refactoring techniques.	2x lectures, 2x practical classes

Basic literature

1. Gamma, Erich, and Helm, Richard, and Johnson, Ralph, and Vlissides, John. *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education, 1994.
2. Freeman, Eric, and Freeman, Elisabeth, and Sierra, Kathy, and Bates, Bert. *Head First Design Patterns*. O'Reilly, 2004.
3. McConnel, Steve. *Code Complete: A Practical Handbook of Software Construction, 2nd Edition*. Microsoft Press, 2004.
4. Larman, Craig. *Applying UML and Patterns – An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Edition.* Prentice Hall, 2005

Supplementary literature

1. Martin, Robert C. *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall, 2009.

JAVA Programming
(Programmēšana JAVA)

Author	Mg. sc. comp. Karina Šķirmante
Course Code	DatZ1028
Form of evaluation	Exam
Credit point (ECTS credit points)	2 (3 ETCS)
Prerequisites	Basics of programming language JAVA (course "Fundamentals of JAVA programming")
Course group	Industry study course

Objective

Objective of this course is to introduce students to JAVA programming language and different modern technologies connected to them as well as to give students basic understanding about the advantages and shortcomings of these technologies and their main cases of usage.

Learning outcomes

After finishing this course students must have basic knowledge about JAVA programming language. Students must understand the possibilities and advantages of using these technologies and they must be able to develop some basic applications using JAVA about the topics covered throughout the course.

Organization mode of students individual assignment

Students must attend lectures or read the according information from the provided presentations or other sources (including internet). Laboratory work. Weekly consulting session with the lecturer is available.

Evaluation of learning outcomes

Final evaluation includes:

- exam/semester project (50%),
- theoretical tests (10%)
- practical assignments (20%)
- practical tests (20%)

Week	**Topic and subtopic**	**Type** (lecture, seminar, laboratory work)
1	OOP : variables and their types, concepts of program, commands, operators	Lecture and

		seminar
2	OOP: inheritance, polymorphism, interfaces	Lecture and seminar
3	Exceptions; JAVA IO	Lecture and seminar
4	Version control; JAVA Unit Tests	Lecture and seminar
5	Threading	Lecture and seminar
6	Networking	Lecture and seminar
7	Connections with database, queries and implementation of results	Lecture and seminar
8	Practical work	Laboratory work

Course outline

Basic literature

1. 'Learning Java, 4th Edition', Patrick Niemeyer, Daniel Leuck, O'Reilly Media, ISBN: 144-9-319-246
2. 'Effective Java Second Edition', Joshua Bloch, Prentice Hall, ISBN:978-0-321-35668-0
3. 'Java How to Program, 7th Edition', H.M.Deitel, Prentice Hall, ISBN:0132222205

Supplementary literature

4. 'Core Web Programming, Second Edition', Marty Hall, Larry Brown, Prentice Hall, ISBN: 978-0-13-089793-0
5. 'Pro Java Programming, Second Edition ', Brett Spell, ISBN:1-59059-474-6
6. 'The Complete Reference JAVA, Seventh Edition', Herbert Schildt, Mc Graw Hill Companies, ISBN: 978-0-07-163177-8

Other source of information

1. The JAVA Tutorial website: http://download.oracle.com/javase/tutorial/index.html
2. Web developer information website: http://www.w3schools.com/

Fundamentals of JAVA Programming
(Programmēšanas pamati JAVA)

Author	Mg. sc. comp. Karina Šķirmante
Course Code	DatZ1027
Form of evaluation	Exam
Credit point (ECTS credit points)	4 (6 ETCS)
Prerequisites	n/a
Course group	Industry study course

Objective

To acquire basic knowledge of algorithms and program development process. Learn algorithmic thinking. Understand and be able to apply procedural programming approach to program development process by using the programming language JAVA.

Learning outcomes

- Able to develop applications (programs) using the programming language JAVA in accordance with good programming practice.
- Able to detect and correct errors in the source code.
- According to requirements of the problem are able to find an appropriate solution and to justify it.
- Able to analyze and explain the JAVA source code.
- Able to work independently with literature and internet resources.

Organization mode of students individual assignment

Systematic work during semester includes:

- regular learning using lecture materials, literature, internet resources,
- completion of home assignments,
- preparation for tests and the final exam,
- weekly teacher consultations.

Evaluation of learning outcomes

Course assessment consists of three parts:

- average grade for the home assignments (30% of total grade)
- average grade for the theoretical tests (20% of total grade)
- exam grade (50% of total grade)

During the semester students have to take two practical tests. If the result of each test is 8 or higher, the student can choose not to write the exam. In this case exam grade is replaced by the average grade for practical tests.

Week	**Topic and subtopic**	**Type** (lecture, seminar, laboratory work)

week1
day1
	Introduction to programming;
	Creating JAVA programs;
	Compiling and executing a program;
	Dealing with errors;
	Comments;
		Lecture and seminars

week1 day2	Variables, arithmetic operators and their priorities; Usage of System.in and System.out classes for input/output; Logical operators, making decisions: if statement.	Lecture and seminars
week1 day3	Loops for, while, do-while, operators break and continue; Making decisions: switch statement.	Lecture and seminars

week1
day4
	Theoretical test No.1;
	Practical exercises;
		Seminar, test

week2 day1	Initializing and using arrays; Initializing and using two-dimensional arrays.	Lecture and seminars
week2 day2	Defining a function; Arguments and parameters of a function; Returning values from functions; Functions with no returning value; Variable scope.	Lecture and seminars
week2 day3	Theoretical test No.2; Practical test No.1	Seminars, tests
week2 day4	Basic concepts of Object-oriented programming (OOP); Class as an abstract data type. Object as a class instance; Class implementation in the programming language JAVA; Access levels; Encapsulation and hiding.	Lecture and seminars

week3
day1
	Constructors;
	The default constructor;
	Overloaded constructors
		Lecture and seminars

week3 day2	Various associations between classes, such as a-kind-of, part-of, has-a; Using UML use to represent association between classes; The composition and aggregation; Inheritance: the base class and inherited class.	Lecture and seminars
week3 day3	Practical exercises	Seminars

week3 day4	Theoretical test No.3; Practical test No.2	Tests

Course outline

Basic literature

1. 'Learning Java, 4th Edition', Patrick Niemeyer, Daniel Leuck, O'Reilly Media, ISBN: 144-9-319-246
2. 'Effective Java Second Edition', Joshua Bloch, Prentice Hall, ISBN:978-0-321-35668-0
3. 'Java How to Program, 7th Edition', H.M.Deitel, Prentice Hall, ISBN:0132222205

Supplementary literature

4. 'Core Web Programming, Second Edition', Marty Hall, Larry Brown, Prentice Hall, ISBN: 978-0-13-089793-0
5. 'Pro Java Programming, Second Edition ', Brett Spell, ISBN:1-59059-474-6
6. 'The Complete Reference JAVA, Seventh Edition', Herbert Schildt, Mc Graw Hill Companies, ISBN: 978-0-07-163177-8

Other source of information

1. The JAVA Tutorial website: http://download.oracle.com/javase/tutorial/index.html
2. Web developer information website: http://www.w3schools.com/

Parallel computing
(Paralēlā programmēšana)

Author	Mg.sc.comp. Kristaps Gromovs
LAIS course code	DatZ2020
Form of evaluation	Exam
Academic credit points (ECTS credit points)	2 (3 ECTS)
The total number of contact lessons	16
The number of lectures	6
The number of practical classes	10
Prerequisites	Basics of JAVA programming language
Part of the study programme	Industry study courses

Study course objective

The study course objective is to master parallel computing and programming application basic concepts, develop knowledge base to apply it in solving simple tasks that require parallel computing, develop knowledge base for programming syntax used for parallel computing in different programming languages.

Study results

Having acquired the study course, a student:

- Is capable of distinguishing tasks that can be solved using parallel computing approach.
- Understands parallel computing paradigm and why there is necessity for parallel computing.
- Is aware of multicore processing unit architecture.
- Is able to explain the difference between concurrent and parallel computing approaches.
- Is capable of parallelizing algorithms and is able to identify algorithms (or their parts), that can't be parallelized.
- Is able to solve simple tasks using parallel computing paradigm.
- Is aware of newest technologies and industry trends for applications that are using parallel computing paradigm.

Organization mode of students' individual work

The independent work of students include:

- a regular learning of the course by using lecture materials, study literature, internet resources,
- course project development,

- work with parallel computing samples,
- preparations for the exam.

Evaluation of study results

The end result is made of:

- Course project 20%,
- Exam 80%

Study course outline

No.	Title of the topic
1.	Introduction to parallel computing.
2.	Introduction to multicore processing unit architectures.
3.	Concurrent and parallel programming models.
4.	Introduction to parallel computing application concepts in different programming languages.
5.	Parallelization of algorithms.
6.	Introduction to different parallel computing language syntaxes.
7.	Newest technologies and trends in parallel applications.

Study course schedule

No. of the class	Title of the topic	Type of class (lectures, seminars, practical classes, laboratory work), amount of academic hours
1.	Introduction to parallel computing. (necessity for parallel computing)	1x lecture, 1x practical class
2.	Introduction to multicore processing unit architectures.	1x lecture, 1x practical class
3.	Concurrent and Parallel programming models and their differences.	1x lecture, 1x practical class
4.	Introduction to parallel computing application concepts in different programming languages.	1x lecture, 1x practical class
5.	Parallelization of algorithms.	1x lecture, 1x practical class
6.	Introduction to different parallel computing language syntaxes.	1x lecture, 1x practical class
7.	Newest technologies and trends in parallel applications.	2x practical class
8.	Course summary. Preparations for the exam.	2x practical class

Basic literature

Rauber, Thomas, and Gudula Rünger. Parallel programming: *For multicore and cluster systems*. Springer Science & Business, 2013.

Supplementary literature

Chandra, Rohit, ed.*Parallel programming in OpenMP*. Morgan Kaufmann, 2001.
Sanders, Jason, Kandrot, Edward. *CUDA by Example: An Introduction to General-Purpose GPU Programming 1st (first) Edition*. Addison-Wesley Professional, 2010.

Other source of information

Other parallel computing languages (developer sites):
http://groups.csail.mit.edu/cag/streamit/ - StreamIT
http://www.parallelpython.com/ - Parallel Python
http://www.jppf.org/ - Java Parallel Processing Framework

Software Architecture Design
(Programmatūras arhitektūras projektēšana)

Author	Mg.sc.ing. Edgars Palacis
LAIS course code	DatZ1031
Form of evaluation	Exam
Academic credit points (ECTS credit points)	2 (3 ECTS)
The total number of contact lessons	16
The number of lectures	8
The number of practical classes	8
Prerequisites	Knowledge of JAVA programming basics
Part of the study programme	Industry study course

Study course objective

The study course objectives are to introduce students with the requirements of the architecture of the modern application; to create understanding of the preference of the most suitable architecture regard to functional requirements of the application and target markets.

Study results

Having acquired the study course, a student:

- Is capable of understand of several architectures types of applications, its advantages and disadvantages
- Is able to avaluate the most suitable architecture solution for a specific project
- Is able to create a simple Web service based on REST API

Organization mode of students' individual work

The independent work of students include:

- a regular learning of the course by using lecture materials, study literature, internet resources,
- development of seminar assignments,
- weekly consulting session with the lecturer

Evaluation of study results

The end result is made of:

- exam/semester project (50%),
- theoretical tests (10%)

- practical assignments (20%)
- practical tests (20%)

Study course outline

No.	Title of the topic
1.	Service oriented architecture, types, specific features and the most popular application (SOAP, REST, WCF, ActiveMQ, etc)
2.	REST services, the most important aspects of the implementation. REST development tools
3.	Data exchange formats: XML and JSON, advantages and disatvantages. Development tools
4.	Types of the application architecture. Monolithic applications, modular application, web services, microservices
5.	Conventional abstraction level of the application structure
6.	Domain driven design pattern
7.	Architecture of the microservices, its features, nuance of the implementation
8.	Server-less arcihtecture

Study course schedule

No. of the class	Title of the topic	Type of class (lectures, seminars, practical classes, laboratory work), amount of academic hours
1.	Service oriented architecture, types, specific features and the most popular application (SOAP, REST, WCF, ActiveMQ, etc)	Lecture and laboratory work
2.	REST services, the most important aspects of the implementation. REST development tools	Lecture and laboratory work
3.	Data exchange formats: XML and JSON, advantages and disatvantages. Development tools	Lecture and laboratory work
4.	Types of the application architecture. Monolithic applications, modular application, web services, microservices	Lecture and laboratory work
5.	Conventional abstraction level of the application structure	Lecture and laboratory work
6.	Domain driven design pattern	Lecture and laboratory work
7.	Architecture of the microservices, its features, nuance of the implementation	Lecture and laboratory work
8.	Server-less arcihtecture	Lecture and laboratory work

Basic literature

1. "SOA with REST. Principles, Patterns & Constraints for Building Enterprise Solutions with REST", Thomas Erl, Benjamin Carlyle, Cesate Pautasso and Raj Balasubramanian, Prentice Hall, ISBN 978-0-13-701251-0

2. "Building microservices", Sam Newman, O'Reilly Media, ISBN 978-1-491-95035-7

3. "Domain driven design", Eric Evans, Addison-Wesley, ISBN 978-0-13-218127-3

Supplementary literature

Other sources of information

Software testing and debugging technologies and principles

(Programmatūras testēšana un atkļūdošanas tehnoloģijas un principi)

Author	Mg.sc.comp. Ervīns Grīnfelds, Mg.sc.comp. Andrejs Frišfelds
Course code	DatZ1030
Examination form	Exam
Credit points (ECTS credit points)	2 (3ECTS)
Preconditions for starting of the course	JAVA programming
Course group	Industry study course

Course objective

The course objective is to provide students with knowledge on testing processes in software development, to give understanding about practical implementation of testing, using various testing methods and environments.

Study outcomes

- to understand and know basic concepts of testing
- to understand the role of testing in the software development process
- to understand testing methods
- to be able to apply software testing tools
- to be able to plan software testing
- to understand testing implementation in Agile projects
- to demonstrate skills in webpage testing, server testing and security testing

Type of organisation of students' independent work

For strengthening their knowledge, students use lecture materials, materials of practical classes and relevant literature.

Assessment of study outcomes

The final assessment is comprised of

- 25% exam
- 65% - practical works that have to be conducted during studies (mobile apps automation task, development of test examples, testing, development of REST API test examples, development of automation of web solutions)
- 10% - activity in lectures, practical works

Course contents

Week	**Topic and subtopic**	**Type** (lectures, workshops, practical classes, laboratory works)
	Introduction (testing necessity, industry standards, practical examples)	lecture (90 min.)
	Team work - to distribute to teams real webpages/apps etc. and give a certain period of time to find the main problems.	Practical class (90 min.)

	When the time is out, to collect the list of all problems and ask questions that had to be considered/understood to be able to start testing a solution/product, and what types of testing were implemented by students to find deficiencies/problems. Work with the test management tool Tarantula or TestRail (basics). Objective = to come to the next topic - types of testing, in order to understand, that there are several levels of testing - starting from code level to UI level, specification level etc., as well as functional and non-functional testing.	
	Types of testing (black box, white box, functional, non-functional testing)	lecture (90 min.)
	To demonstrate on a simple JAVA code, how a unit test example looks like, how it is written, saved and ran. Another important thing to show in this lecture is how unit test “fails” and what message is displayed to the programmer. To give students access to a code and assign each student to write 3 unit tests for an app (even if it is a calculator written in JAVA code) The task for team work - together with the teacher figure out conceptually, what should be tested/planned for testing, when developing a social network, which will operate on mobile/web platforms, and on which 10M users are expected	Practical class (90 min.)
	Development of testing strategy, test planning and management	Lecture (90 min.)
	Practical works: * A task is given to conduct testing of NotifyUs.net app server, where the tester has limited resources and time, funds, available workforce, devices etc. - how it is planned, how it will be done (group work, where each group proposes their own vision and each group has different limitations - for one group - time, for another group - budget, for another one - human resources, and some groups have to plan server PATCH to product environment, if the problem is already in the product) * The task is to develop a general testing concept (in its essence, not a formal plan) with things to be tested while planning a product of mobile app + server app * To become acquainted with Jira project management tool, to go through together with the teacher, how defining of tasks is done, how sprint planning and closing of tasks is implemented, without forgetting, how testing during sprint and accept testing for versions is done * To become acquainted in-depth with the test management tool Tarantula or TestRail with the task to define in the tool	Practical class (180 min.)

	test examples for a specific system/app. * To perform testing on any IT product, to give a task to define in Jira task management tool a problem message for any found problem and deficiencies	
	Testing in various elaboration methodologies, in-depth insight into Agile methodology.	Lecture/Practical class (180 min.)
	White box testing - debugging, unit testing, integration testing.	Lecture with examples (90 min.)
	Testing of webpages (used tools, technologies, types of testing)	Lecture (45 min.)
	Practical tasks: * To conduct testing on any of most popular webpages, using free web solution validators * By using Selenium tool, to develop a web automation task for web solutions as www.testdevlab.com or www.notifyus.net (task comprises of 2 parts - defining of test examples in the test management tool TestRail or Tarantula, and their actual automation by using the Selenium tool. Approximately 5 test examples with 15 validations should be automated)	Practical class (135 min.)
	Testing of mobile devices (used tools, technologies, types of testing)	Lecture (45 min.)
	* The elaboration environment has to be installed on computers with android SDK for this task!!! Students are given a simple, ready android app product. A few integration tests are written during the practical work on JAVA for android app. * Students are given a simple android app, which does not present wide functionality and is only a simple client's app. Students are given a task to develop automated tests for this app by using Robotium framework (the task as in the case with 5.2 Web - first to define, and then - automate. Up to 5 test examples with 15 validations should be automated) * Students are shown, how Calaba.sh framework is used to set a mobile device testing and development continuous integration solution, where it is ensured, that tests are booted after a specific period of time or particular command (learned/discovered risks - Jenkins, Teamcity, Calaba.sh)	Practical class (135 min.)
	Server (backend) testing (used tools, technologies, types of testing) + practical task to perform REST API testing, using JMeter and Apimation tools	Lecture/Practical class (180 min.)
	Security testing (theory, necessity, practical examples and tasks)	Lecture/Practical class (90 min.)

Basic literature

1. Abbas N., Gravell A. M., Wills G. B. Historical Roots of Agile Methods: Where did “Agile Thinking” come from? // Agile Processes in Software Engineering and Extreme Programming. – 2008. – pp. 94–103.
2. Dustin E., Rashka J., Paul J. Automated Software Testing: Introduction, Management and Performance. – Boston, MA, USA, 1999. – 608 p.
3. Kan S. H. Metrics and Models in Software Quality Engineering. 2nd ed. – Boston, MA, USA: Addison-Wesley, 2002. – 560 p.
4. Kaner C., Bach J., Pettichord B. Lessons Learned in Software Testing. – New York, NY, USA: Wiley, 2001. – 352 p.

Additional literature

Other sources of information