



STUDIJŲ DALYKO (MODULIO) APRAŠAS

Dalyko (modulio) pavadinimas	Kodas
Išmaniųjų įrenginių programavimas	

Dėstytojas (-ai)	Padalinys (-iai)
Koordinuojantis: dr. Igor Katin Kitas (-i): dr. Viktor Medvedev	Matematikos ir informatikos fakultetas Duomenų mokslo ir skaitmeninių technologijų institutas

Studijų pakopa	Dalyko (modulio) tipas
Pirmoji	Pasirenkamas

Įgyvendinimo forma	Vykdyto laikotarpis	Vykdyto kalba (-os)
Auditorinė	6 semestras	Lietuvių / Anglų

Reikalavimai studijuojančiajam	
Išankstiniai reikalavimai: Naudotojo sąsajų kūrimas, Algoritmai ir duomenų struktūros, Procedūrinis programavimas, Objektinis programavimas, Operacinės sistemos.	Gretutiniai reikalavimai (jei yra):

Dalyko (modulio) apimtis kreditais	Visas studento darbo krūvis	Kontaktinio darbo valandos	Savarankiško darbo valandos
5	131	48	83

Dalyko (modulio) tikslas: studijų programos ugdomos kompetencijos		
Dalyko tikslas – siekiama, kad studentai įgytų žinias apie išmaniųjų įrenginių programavimą, supažindintų su išmaniųjų įrenginių platformomis, taikomųjų programų išmaniesiems įrenginiams kūrimo pagrindais, ugdytų praktinius gebėjimus taikyti programavimo žinias kuriant taikomąsias programas išmaniesiems įrenginiams, suprasti ir analizuoti sudėtingus uždavinius.		
Dalyko (modulio) studijų siekiniai	Studijų metodai	Vertinimo metodai
Gebės naudoti esamus kolektyvinio darbo sistemų sprendimus, projektuojant ir kuriant taikomąsias programas išmaniesiems įrenginiams.	Informacijos paieška ir analizė, Sąvokų žemėlapis, Savarankiškas ir grupinis darbas, Paskaita-diskusija, Praktinės užduotys, Rašto darbas, Pranešimo rengimas ir pristatymas.	Testavimas, Grupinių darbų pristatymas, Savarankiški praktiniai darbai, Laboratoriniai darbai, Rašto darbas, Pateikčių rengimas ir darbo pristatymas, Egzaminas.
Gebės analizuoti ir taikyti išmaniesiems įrenginiams pritaikytus programavimo ir projektavimo metodus, nustatyti sukurtos programinės įrangos tobulinimo, atnaujinimo tikslus, rengti išsamias aprašus.		
Gebės projektuoti ir įgyvendinti programinius sprendimus Android ir iOS operacinėms sistemoms bei įvairiems mobiliems įrenginiams, taip pat gebės parinkti optimalias programines įgyvendinimo priemones.		
Tobulinant bei modifikuojant sukurtas taikomąsias programas išmaniesiems įrenginiams, gebės taikyti atvirkštinės inžinerijos metodus.		
Gebės parinkti ir pritaikyti perspektyvias technologijas, naujas programavimo kalbas (Objective-C, Swift) išmaniųjų įrenginių programavimo uždaviniams spręsti.		
Gebės įgytas žinias apie išmaniųjų įrenginių programavimą pritaikyti siekiu užtikrinti debesų kompiuterijos resursų infrastruktūros prieinamumą.		

Temos	Kontaktinio darbo valandos							Savarankiškų studijų laikas ir užduotys	
	Paskaitos	Konsultacijos	Seminarai	Pratybos	Laboratoriniai darbai	Praktika	Visas kontaktinis darbas	Savarankiškas darbas	Užduotys
Įvadas. Išmaniųjų įrenginių programavimo pagrindai ir galimybės. Operacinių sistemų, naudojamų išmaniuosiuose telefonuose, apžvalga, lyginamoji analizė: Android, Windows Mobile, iOS. Taikomųjų programų išmaniesiems įrenginiams kūrimo įrankiai: Xamarin, Android Studio, Eclipse, Xcode.	4				3		7	7	Literatūros paieška ir analizė

Taikomųjų programų, skirtų Android operacinei sistemai, kūrimas. Android operacinės sistemos (OS) architektūros ypatumai. Taikomųjų programų veikimo pagrindai, ypatumai ir struktūra, programų sudedamosios dalys. Programų kūrimo įrankiai Android platformai. Android OS taikomųjų programų projektavimas, įvairių įrenginių palaikymas (Supporting Different Devices). Taikomųjų programų išmaniesiems įrenginiams gyvavimo ciklas. Vartotojo sąsajos komponentai. Pranešimų ir įvykių valdymas. Taikomųjų programų testavimas ir derinimas. Populiariausių atvirojo kodo bibliotekų apžvalga.	8				9		17	30	Literatūros paieška ir analizė; Savarankiškų praktinių užduočių atlikimas, kuriant taikomąsias programas išmaniesiems įrenginiams Android operacinei sistemai; Rašto darbo parengimas; Grupinio darbo pristatymas
Taikomųjų programų, skirtų iOS operacinei sistemai, kūrimas. iOS architektūros ypatumai. Taikomųjų programų veikimo pagrindai, programų sudedamosios dalys. iOS taikomųjų programų išmaniesiems įrenginiams projektavimas, programos gyvavimo ciklas. iOS vartotojų sąsajos komponentai. Objective-C ir Swift programavimo kalbos.	8				8		16	30	Literatūros paieška ir analizė; Savarankiškų praktinių užduočių atlikimas, kuriant taikomąsias programas išmaniesiems įrenginiams iOS operacinei sistemai; Rašto darbo parengimas; Grupinio darbo pristatymas
Sukurtos programos atnaujinimas ir tobulinimas, pateikimas galutiniam vartotojui. Šiuolaikiniai reikalavimai taikomosioms programoms, skirtoms išmaniesiems įrenginiams. Taikomųjų programų, skirtų Android, iOS operacinėms sistemoms, talpinimas „Google Play“, „iOS App“, „Amazon Appstore“ parduotuvėse.	4				4		8	10	Literatūros paieška ir analizė; Savarankiškų praktinių užduočių atlikimas; Komandinis projekto vykdymas.
Pasiruošimas egzaminui ir egzamino laikymas.								6	Literatūros kartojimas
Iš viso	24				24		48	83	

Vertinimo strategija	Svoris proc.	Atsiskaitymo laikas	Vertinimo kriterijai
Projektinis darbas (pirmoji dalis)	30	semestro metu	Studentai pasirenka projekto temą savarankiškai arba iš pasiūlytų temų sąrašo. Vieną temą renka vienas arba keli studentai, bet maksimalus vieno projekto studentų skaičius yra 2. Temą privaloma rezervuoti ir apie tai pranešti dėstytojui. Temos pasirinkimas vyksta 1-2 sav. Privaloma iki semestro vidurio (8 sav.) pateikti projekto idėją, išsamų planą, pasiūlyti projektinius sprendimus, naudojamus algoritmus, duomenų modelius. Privaloma pateikti apie 1/3 veikiančio projekto kodo (vidutiniškai 400-600 eil. asmeninio, nesugeneruoto kodo vienam komandos nariui). Vertinimo kriterijai: teisingai veikiantis išmaniajame įrenginyje projekto kodas, programavimo standartų ir gerų praktikų naudojimas, algoritmų, šablonų, metodų taikymas.
Projektinis darbas (antroji dalis)	30	semestro metu	Studentai užbaigia projektą ir pristato jį grupėje. Studentai pristato užbaigtą projektą, projektinius sprendimus, testavimo metodiką. Vyksta semestro pabaigoje. Privaloma pristatyti pilnai veikiantį išmaniajame įrenginyje projektą su visu programiniu kodu (vidutiniškai 800-1200 eil. asmeninio, nesugeneruoto kodo vienam komandos nariui). Vertinama: teisingai veikiantis projekto programinis kodas, programavimo standartų ir gerų praktikų naudojimas, algoritmų, šablonų, metodų taikymas, kokybiški programiniai ir testavimo sprendimai, naudotojo sąsaja, duomenų modelis, grupinis darbas (užduočių atlikimas laiku). Norint laikyti egzaminą, privaloma atlikti nemažiau nei pusę projekto reikalavimų.
Pasirinkto mobiliojo karkaso ar technologijos pristatymas	10	semestro metu	0,5 balo už technologijos/karkaso pristatymą. 0,5 balo už pasirinktos technologijos/karkaso kodo pavyzdį.
Egzaminas	30	Egzaminų sesijos metu	Testą sudaro 30 atvirojo ir uždarojo tipo klausimų. 3 balai – labai geros žinios ir gebėjimai. 25–30 teisingų atsakymų, 2 balai – vidutinės žinios ir gebėjimai, yra klaidų. 19–24 teisingi atsakymai; 1 balas – žinios ir gebėjimai nesiekia vidutinių, yra esminių klaidų. 13–18 teisingi atsakymai; 0,5 balo – žinios ir gebėjimai dar patenkina minimalius reikalavimus, tačiau daug klaidų. 7–12 teisingų atsakymų; 0 balų – nepatenkinami minimalūs reikalavimai. 0–6 teisingi atsakymai. Perlaikymo atveju studentas privalo pristatyti projekto rezultatus pagal pirmos ir antros dalies vertinimo kriterijus bei išlaikyti egzaminą. Dalyko negalima laikyti eksternu.

Autorius	Leidimo metai	Pavadinimas	Periodinio leidinio Nr. ar leidinio tomas	Leidimo vieta ir leidykla ar internetinė nuoroda
Privaloma literatūra				
Neil Smyth	2024	Android Studio Iguana Essentials - Kotlin Edition: Developing Android Apps Using Android Studio 2023.2.1 and Kotlin	ISBN-10 1951442857 ISBN-13 978- 951442859	Payload Media
Sufyan bin Uzayr	2022	Mastering Android Studio: A Beginner's Guide (Mastering Computer Science) 1st Edition	ISBN-10 1032134127 ISBN-13 978- 1032134123	CRC Press; 1st edition (February 24, 2022)
Google	2024	Android developer documentation		https://developer.android.com/guide
Papildoma literatūra				
Thomas Bailey, Alessandro Biessek, Trevor Wills	2021	Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter 2.5 and Dart		Packt Publishing
Roger Ye	2024	.NET MAUI Cross-Platform Application Development - Second Edition: Build high-performance apps for Android, iOS, macOS, and Windows using XAML and Blazor with .NET 8	ISBN-10 1835080596 ISBN-13 978- 1835080597	Packt Publishing
Ahmad Sahar	2023	iOS 17 Programming for Beginners - Eighth Edition: Unlock the world of iOS Development with Swift 5.9, Xcode 15, and iOS 17 - Your Path to App Store Success 8th ed. Edition	ISBN-10 1837630569 ISBN-13 978- 1837630561	Packt Publishing



COURSE UNIT (MODULE) DESCRIPTION

Course unit (module) title	Code
Smart Device Programming	

Lecturer(s)	Department(s) where the course unit (module) is delivered
Coordinator: Dr. Igor Katin Other(s): Dr. Viktor Medvedev	Faculty of Mathematics and Informatics Institute of Data Science and Digital Technologies

Study cycle	Type of the course unit (module)
First	Optional

Mode of delivery	Period when the course unit (module) is delivered	Language(s) of instruction
face-to-face	6 th semester	Lithuanian / English

Requirements for students	
Prerequisites: User Interface Design, Algorithms and Data Structures, Procedural and Object-Oriented Programming, Operating Systems	Additional requirements (if any):

Course (module) volume in credits	Total student's workload	Contact hours	Self-study hours
5	131	48	83

Purpose of the course unit (module): programme competences to be developed

The purpose of the course unit is to provide students with the knowledge on smart device programming, familiar with the mobile device platform, the development of applications for smart mobile devices, to develop practical skills to apply the programming knowledge for developing applications for smart devices.

Learning outcomes of the course unit (module)	Teaching and learning methods	Assessment methods
Ability to use groupware systems developing applications for smart mobile devices. Ability to analyse and apply the programming and design methods for developing applications for smart mobile devices, and to identify needs of mobile applications development and update, and prepare the detailed documentation. Ability to design and implement software solutions for Android and iOS operating systems and to choose the optimal programming tools. Ability to apply reverse engineering methods for improving and modifying developed mobile applications for smart devices. Ability to select and apply promising technologies, new programming languages (Objective-C, Swift) to solve smart devices programming challenges. Ability to apply the acquired knowledge in the domains of smart devices programming by developing and deploying cloud computing solutions.	Literature search and analysis; Concept maps; Self-dependent and group work; Lecture-discussion; Practical tasks; Example demonstration and analysis; Written works; Report preparation and presentation.	Tests; Groupwork; Independent practical tasks (project); Laboratory work; Project presentation and defence; Exam.

Content: breakdown of the topics	Contact hours							Self-study work: time and assignments	
	Lectures	Tutorials	Seminars	Exercises	Laboratory work	Internship/work placement	Contact hours	Self-study hours	Assignments
1. Introduction. Smart device programming: fundamentals and opportunities. Mobile operating systems (mobile OS) Android, Windows Mobile, iOS: review, comparative analysis. Advanced integrated development environments for iOS and Android: Xamarin, Android Studio, Eclipse, Xcode.	4				3		7	7	Individual reading.

2. Development of Android applications for smart devices. The structure and features of the Android operating system platform. General principles of mobile applications design. The structure and components of mobile applications. Software development tools for Android applications. Android applications design. Supporting different devices. Mobile application life cycle. User interface components. Application testing and debugging. Open source libraries and frameworks.	8				9		17	30	Individual reading; Laboratory assignments; Practical group work; Preparation of written work
3. Development of iOS applications for smart devices. The structure and features of the iOS platform. General principles of mobile applications design. The structure and components of mobile applications. Design of iOS applications. iOS application lifecycle. iOS user interface components. Objective-C and Swift programming languages.	8				8		16	30	Individual reading; Laboratory assignments; Practical work (project); Preparation of written work.
4. Developing an End-User application: requirements, updating, improvement. Modern software engineering methodologies for mobile applications. Uploading and publishing applications for smart devices to Google Play, iOS App, Amazon Appstore.	4				4		8	10	Individual reading; Laboratory assignments; Practical work (project); Preparation of written work.
5. Exam.								6	Literature review; preparation for the exam.
Total	24				24		48	83	

Assessment strategy	Weight,%	Deadline	Assessment criteria
Project work (part 1)	30	During the semester	Students choose a project topic independently or from a list of suggested topics. One or several students choose one topic, but the maximum number of students per project is 2. The topic must be reserved and the teacher informed about it. The selection of the topic should be done in Weeks 1-2. It is mandatory to submit a project idea, a detailed plan, propose project solutions, used algorithms, data models by the middle of the semester (in Weeks 8-9). It is mandatory to submit about 1/3 of the working project code (on average 400-600 lines of personal, non-generated code per team member). Evaluation criteria: correctly functioning project code on a smart device, use of programming standards and good practices, application of algorithms, templates, methods.
Project work (part 2)	30	End the semester	Students complete the project and present it to the group. Students present the completed project, project solutions, testing methodology. Should be done at the end of the semester. It is mandatory to present a project fully functional on a smart device with all software code (on average 800-1200 lines of personal, non-generated code per team member). Evaluated: correctly functioning project code, use of programming standards and good practices, application of algorithms, templates, methods, quality software and testing solutions, user interface, data model, group work (timely completion of tasks). In order to take the exam, at least half of the project requirements must be completed.
Presentation of selected mobile framework or technology	10	During the semester	0.5 points for technology/framework presentation. 0.5 points for sample code of selected technology/framework.
Final exam	30	Exam session	The test consists of 30 open and closed type questions. • 3 points - very good knowledge and skills. 25-30 correct answers, • 2 points - average knowledge and abilities, there are mistakes. 19-24 correct answers; • 1 point – knowledge and abilities are below average, there are significant errors. 13-18 correct answers; • 0.5 points - knowledge and skills still meet the minimum requirements, but there are many errors. 7-12 correct answers; • 0 points – the minimum requirements are not met. 0-6 correct answers. In case of a retake, the student must present the results of the project according to the evaluation criteria of the first and second part and pass the exam. Exam cannot be taken as an external student.

Author	Year of publication	Title	Issue of a periodical or volume of a publication	Publishing place and house or web link
Compulsory reading				
Neil Smyth	2024	Android Studio Iguana Essentials - Kotlin Edition: Developing Android Apps Using Android Studio 2023.2.1 and Kotlin	ISBN-10 1951442857 ISBN-13 978-951442859	Payload Media
Sufyan bin Uzayr	2022	Mastering Android Studio: A Beginner's Guide (Mastering Computer Science) 1st Edition	ISBN-10 1032134127 ISBN-13 978-1032134123	CRC Press; 1st edition (February 24, 2022)
Google	2024	Android developer documentation		https://developer.android.com/guide
Optional reading				
Thomas Bailey, Alessandro Biessek , Trevor Wills	2021	Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter 2.5 and Dart		Packt Publishing
Roger Ye	2024	.NET MAUI Cross-Platform Application Development - Second Edition: Build high-performance apps for Android, iOS, macOS, and Windows using XAML and Blazor with .NET 8	ISBN-10 1835080596 ISBN-13 978-1835080597	Packt Publishing
Ahmad Sahar	2023	iOS 17 Programming for Beginners - Eighth Edition: Unlock the world of iOS Development with Swift 5.9, Xcode 15, and iOS 17 - Your Path to App Store Success 8th ed. Edition	ISBN-10 1837630569 ISBN-13 978-1837630561	Packt Publishing