

JAUNESNIOJO PROGRAMUOTOJO MODULINĖ PROFESINIO MOKYMO PROGRAMA

(Programos pavadinimas)

Programos valstybinis kodas, išsilavinimo reikalavimai stojančiajam ir apimtis mokymosi kreditais:

P42061304 – programa, skirta pirminiam profesiniam mokymui, asmenims, turintiems pagrindinį išsilavinimą ir kurie kartu mokosi vidurinio ugdymo programą, 60 mokymosi kreditų (jei taikoma)

P43061306 – programa, skirta pirminiam profesiniam mokymui, asmenims, turintiems vidurinį išsilavinimą, 60 mokymosi kreditų

T43061309 – programa, skirta tęstiniam profesiniam mokymui, asmenims, turintiems vidurinį išsilavinimą, 50 mokymosi kreditų

Kvalifikacijos pavadinimas – jaunesnysis programuotojas

Kvalifikacijos lygis pagal Lietuvos kvalifikacijų sandarą (LTKS) – IV

Kiti reikalavimai stojančiajam – laikomas testas stojančiojo pasirengimui mokytis nustatyti

1. PROGRAMOS APIBŪDINIMAS

Programos paskirtis. Jaunesniojo programuotojo modulinė profesinio mokymo programa skirta kvalifikuotam darbuotojui parengti, kuris gebėtų projektuoti ir programuoti vartotojo sąsajas, projektuoti ir kurti nesudėtingas duomenų bazines, valdyti programavimo aplinkos ir programinės įrangos kūrimo procesą, projektuoti ir kurti informacines sistemas viena iš programavimo kalbų (Java, JavaScript, C#, Python, PHP) arba programuoti žaidimus, naudojant Unity platformą.

Būsimo darbo specifika. Įgijęs kvalifikaciją asmuo galės dirbti jaunesniuoju programuotoju informacinių ir ryšių technologijų sektoriaus verslo įmonėse, kitų sektorių organizacijų informacinių technologijų padaliniuose.

Dirbama individualiai ir (arba) komandoje, galimas lankstus darbo grafikas, nuotolinis darbas. Jaunesnysis programuotojas paprastai dirba prižiūrimas vyresniojo specialisto ir vykdo jam iš anksto suformuluotas ir pateiktas užduotis, savarankiškai fiksuoja ir tikrina savo darbo rezultatus, dirba ir bendrauja su kolegomis (sistemų analitikais, testuotojais, priežiūros specialistais), keičiantis technologijoms, nuolat atnaujina ir gilina žinias, ir tobulina savo įgūdžius. Jo darbo rezultatai tikrinami prieš integruojant į bendrą komandos darbo rezultata. Augant specialisto patirčiai jo savarankiškumo lygmuo didėja.

Jaunesniajam programuotojui svarbios šios asmeninės savybės: kruopštumas, atsakingumas, savarankiškumas, dėmesio koncentracija.

2. PROGRAMOS PARAMETRAI

Valstybinis kodas	Modulio pavadinimas	LTKS lygis	Apimtis mokymosi kreditais	Kompetencijos	Kompetencijų pasiekimą iliustruojantys mokymosi rezultatai
Įvadinis modulis (iš viso 1 mokymosi kreditai)*					
4000005	Įvadas į profesiją	IV	1	Pažinti profesiją.	Išmanyti jaunesniojo programuotojo profesiją ir jos teikiamas galimybes darbo rinkoje. Suprasti jaunesniojo programuotojo profesinę veiklą, veiklos procesus, funkcijas ir uždavinius. Demonstruoti jau turimus, neformaliuotu ir (arba) savaiminiu būdu įgytus jaunesniojo programuotojo kvalifikacijai būdingus gebėjimus.
Bendrieji moduliai (iš viso 4 mokymosi kreditai)*					
4102201	Saugus elgesys ekstremaliose situacijose	IV	1	Saugiai elgtis ekstremaliose situacijose.	Išmanyti ekstremalių situacijų tipus, galimus pavojus. Išmanyti saugaus elgesio ekstremaliose situacijose reikalavimus ir instrukcijas, garsinius civilinės saugos signalus.
4102105	Sąmoningas fizinio aktyvumo reguliavimas	IV	1	Reguluoti fizinį aktyvumą.	Išmanyti fizinio aktyvumo formas. Demonstruoti asmeninį fizinį aktyvumą. Taikyti fizinio aktyvumo formas, atsižvelgiant į darbo specifiką.
4102203	Darbuotojų sauga ir sveikata	IV	2	Tausoti sveikatą ir saugiai dirbti.	Išmanyti darbuotojų saugos ir sveikatos reikalavimus, keliamus darbo vietai.
Kvalifikaciją sudarančioms kompetencijoms įgyti skirti moduliai (iš viso 25 mokymosi kreditai)					
<i>Privalomieji (iš viso 25 mokymosi kreditai)</i>					
406131343	Vartotojo sąsajos projektavimas ir programavimas	IV	10	Taikyti programinės įrangos kūrimui naudojamus informatikos principus ir metodus.	Apibūdinti įvairias skaičiavimo sistemas. Parinkti duomenų struktūras, pagrindinius algoritmus taikant logikos principus. Sukurti algoritmų schemas problemoms išspręsti.
				Projektuoti ir programuoti žiniatinklio puslapių vartotojo sąsajas.	Paašškinti interneto veikimo principus. Pateikti internetinio puslapio turinį, naudojant kompiuterinę žymėjimo kalbą. Apipavidalinti internetinį puslapį, naudojant pakopinius stilių šablonus ir karkasus.

					<p>Valdyti JavaScript programavimo kalbos įrankius ir sintaksę.</p> <p>Kurti nesudėtingą programinį kodą JavaScript programavimo kalba.</p> <p>Valdyti HTML elementų medį, naudojant JavaScript programavimo kalbą (DOM).</p> <p>Kurti vartotojo sąsajas, naudojant JavaScript programavimo kalbos karkasus.</p>
406121239	Nesudėtingų duomenų bazių projektavimas ir kūrimas	IV	5	Projektuoti tipines reliacines ir nereliacines (NoSQL) duomenų bases.	<p>Apibūdinti duomenų bazių tipus, skirtumus ir taikymo galimybes.</p> <p>Kurti reliacinės duomenų bazės schemą, naudojant programinę įrangą.</p> <p>Kurti nereliacinės (NoSQL) duomenų bazės schemą, naudojant programinę įrangą.</p>
				Naudoti ir integruoti duomenų bases.	<p>Apibūdinti duomenų bazių sistemas, jų diegimą ir valdymą.</p> <p>Kurti užklausas, naudojant SQL ir NoSQL kalbas duomenų bazės užpildymui ir informacijos išrinkimui.</p> <p>Administruoti duomenų bases, naudojant gerąsias duomenų bazių administravimo praktikas.</p>
406131344	Programavimo aplinkos ir programinės įrangos kūrimo proceso valdymas	IV	10	Naudoti tarnybinių stočių operacines sistemas.	<p>Apibūdinti serverių veikimo principus ir panaudojimo galimybes.</p> <p>Administruoti Web serverio programinę įrangą bylų viešinimui HTTP protokolu.</p> <p>Pritaikyti Docker konteinerius bylų viešinimui HTTP protokolu.</p>
				Analizuoti skirtingų tipų reikalavimus, keliamus kuriamai programinei įrangai.	<p>Paašškinti AGILE filosofiją ir jos principus.</p> <p>Paašškinti SCRUM proceso dalis ir komandos narių atsakomybes.</p> <p>Analizuoti pateiktus techninius reikalavimus, naudojant projekto eigos valdymo principus ir įrankius.</p>
				Kurti programinę įrangą individualiai ir dirbant komandoje, taikant programinės įrangos kūrimo metodus.	<p>Paašškinti versijų kontrolės svarbą įrangos kūrimo procese.</p> <p>Derinti komandinį programavimo darbų vykdymą, naudojant versijavimo platformas.</p>

					Versijuoti programinį kodą, naudojant versijų kontrolės sistemas.
<i>Privalomai pasirenkamieji (iš viso 20 mokymosi kreditų)</i>					
406131345	Informacinių sistemų projektavimas ir kūrimas Java programavimo kalba	IV	20	Kurti tipinę programinę įrangą Java programavimo kalba.	Apžvelgti Java programavimo kalbos atsiradimo istoriją ir pagrindines taikymo sritis. Valdyti Java programavimo kalbos įrankius ir sintaksę. Kurti nesudėtingą programinį kodą Java programavimo kalba. Programuoti Java programavimo kalba, taikant objektinio programavimo principus. Testuoti programinę įrangą, naudojant su Java programavimo kalba suderinamus testavimo įrankius ir metodus. Naudoti dirbtinio intelekto įrankius programuojant.
				Naudoti Spring karkasą Java platformai.	Paašškinti programavimo karkasų privalumus ir trūkumus. Konfigūruoti Spring priklausomybių valdymo karkasą įvairiais būdais. Naudoti Spring Bean Java programose įvairiais būdais.
				Naudoti Java Persistence <i>API</i> duomenų valdymui Java programose.	Paašškinti objektų ir reliacinės duomenų bazės komunikavimą, taikant įrankius. Susieti objektus ir reliacines duomenų bazines naudojant Java Persistence API. Valdyti duomenis, naudojant Hibernate karkasą taikomuosiose Java programose.
406131346	Informacinių sistemų projektavimas ir kūrimas JavaScript programavimo kalba	IV	20	Kurti tipinę programinę įrangą JavaScript programavimo kalba.	Paašškinti objektinio programavimo principus JavaScript programavimo kalboje. Programuoti JavaScript programavimo kalba, taikant objektinio programavimo principus. Testuoti programinę įrangą, naudojant su JavaScript programavimo kalba suderinamus testavimo įrankius ir metodus. Naudoti dirbtinio intelekto įrankius programuojant.
				Kurti JavaScript taikomąsias programas, naudojant karkasus.	Paašškinti programavimo karkasų privalumus ir trūkumus. Kurti taikomąsias programas, naudojant JavaScript programavimo kalbos karkasus.

					Kurti taikomąsias žiniatinklio programas su grafine sąsaja, naudojant JavaScript programavimo kalbos karkasus.
406131347	Informacinių sistemų projektavimas ir kūrimas C# programavimo kalba	IV	20	Kurti tipinę programinę įrangą C# programavimo kalba.	Apžvelgti C# programavimo kalbos atsiradimo istoriją ir pagrindines taikymo sritis. Valdyti C# programavimo kalbos įrankius ir sintaksę. Kurti nesudėtingą programinį kodą C# programavimo kalba. Programuoti C# programavimo kalba, taikant objektinio programavimo principus. Testuoti programinę įrangą, naudojant su C# programavimo kalba suderinamus testavimo įrankius ir metodus. Naudoti dirbtinio intelekto įrankius programuojant.
				Kurti C# taikomąsias programas, naudojant karkasus.	Paašškinti programavimo karkasų privalumus ir trūkumus. Kurti taikomąsias žiniatinklio programas su grafine vartotojo sąsaja, naudojant .NET aplinką. Programuoti taikomąsias žiniatinklio programas su grafine vartotojo sąsaja, naudojant daugiagijinio ir asinchroninio programavimo principus.
406131348	Informacinių sistemų projektavimas ir kūrimas PHP programavimo kalba	IV	20	Kurti tipinę programinę įrangą PHP programavimo kalba.	Apžvelgti PHP programavimo kalbos atsiradimo istoriją ir pagrindines taikymo sritis. Valdyti PHP programavimo kalbos įrankius ir sintaksę. Kurti nesudėtingą programinį kodą PHP programavimo kalba. Programuoti PHP programavimo kalba, taikant objektinio programavimo principus. Testuoti programinę įrangą, naudojant su PHP programavimo kalba suderinamus testavimo įrankius ir metodus. Naudoti dirbtinio intelekto įrankius programuojant.
				Kurti taikomąsias PHP programas, naudojant <i>Laravel</i> karkasą.	Paašškinti programavimo karkasų privalumus ir trūkumus. Kurti taikomąsias žiniatinklio programas su grafine sąsaja, naudojant karkasą. Kurti žiniatinklio programas ir API servigus, naudojant karkasą.

406131349	Informacinių sistemų projektavimas ir kūrimas Python programavimo kalba	IV	20	Kurti tipinę programinę įrangą Python programavimo kalba.	Apžvelgti Python programavimo kalbos atsiradimo istoriją ir pagrindines taikymo sritis. Valdyti Python programavimo kalbos įrankius ir sintaksę. Kurti nesudėtingą programinį kodą Python programavimo kalba. Programuoti Python programavimo kalba, taikant objektinio programavimo principus. Testuoti programinę įrangą, naudojant su Python programavimo kalba suderinamus testavimo įrankius ir metodus. Naudoti dirbtinio intelekto įrankius programuojant.
				Kurti taikomąsias programas, naudojant Python karkasus.	Paašškinti programavimo karkasų privalumus ir trūkumus. Kurti taikomąsias žiniatinklio programas su grafine vartotojo sąsaja, naudojant Python programavimo kalbą. Kurti taikomąsias programas, naudojant duomenų analizės ir vizualizacijos karkasus. Kurti taikomąsias programas, naudojant mašininio mokymo modelius.
406131350	Žaidimų programavimas naudojant Unity platformą	IV	20	Naudoti Unity platformą nesudėtingiems žaidimams kurti.	Paašškinti žaidimų kūrimo procesą, nagrinėjant žaidimo specifikaciją. Valdyti C# programavimo kalbos įrankius ir sintaksę. Kurti nesudėtingą programinį kodą C# programavimo kalba. Programuoti C# programavimo kalba, taikant objektinio programavimo principus. Kurti žaidimus Unity aplinkoje, taikant žaidimų kūrimo principus. Naudoti dirbtinio intelekto įrankius programuojant.
				Programuoti sudėtingesnius žaidimus, naudojant Unity platformą.	Apibūdinti grafinių elementų komponavimo principus. Kurti dvimatės grafikos produktus, naudojant žaidimų kūrimo programinę įrangą. Kurti trimatčius grafikos produktus, naudojant žaidimų kūrimo programinę įrangą.
Pasirenkamieji moduliai (iš viso 5 mokymosi kreditai)*					

406131351	Vartotojo sąsajos kūrimas naudojant TypeScript programavimo kalbą	IV	5	Kurti tipinę programinę įrangą TypeScript programavimo kalba.	Apžvelgti TypeScript programavimo kalbos atsiradimo istoriją ir pagrindines taikymo sritis. Valdyti TypeScript programavimo kalbos įrankius ir sintaksę. Kurti nesudėtingą programinį kodą TypeScript programavimo kalba.
				Naudoti TypeScript programavimo kalbą vartotojo sąsajos kūrime.	Kurti žiniatinklio programas, naudojant TypeScript programavimo kalbą. Testuoti žiniatinklio programas, naudojant TypeScript programavimo kalbą.
406131352	Dirbtinio intelekto taikymas programinės įrangos kūrimo ir testavimo procesuose	IV	5	Taikyti bendruosius dirbtinio intelekto veikimo principus informacinių technologijų sektoriaus profesinėje veikloje.	Apibūdinti dirbtinio intelekto sampratą ir veikimo principus. Parinkti dirbtinio intelekto modulius pagal konkrečią situaciją ir sprendžiamą problemą. Taikyti dirbtinio intelekto modulius praktinėms problemoms spręsti.
				Kurti programinės įrangos sprendimus naudojant dirbtinį intelektą.	Paašškinti dirbtinio intelekto taikymo sritis programinės įrangos kūrimo procese. Taikyti dirbtinio intelekto įrankius programinės įrangos reikalavimų analizės ir projektavimo etapuose. Taikyti dirbtinio intelekto įrankius programavimo ir testavimo etapuose programinės įrangos kūrimo procese.
Baigiamasis modulis (iš viso 5 mokymosi kreditai)					
4000004	Įvadas į darbo rinką	IV	5	Formuoti darbinius įgūdžius realioje darbo vietoje.	Įsivertinti ir realioje darbo vietoje demonstruoti įgytas kompetencijas. Susipažinti su būsimo darbo specifika ir adaptuotis realioje darbo vietoje. Įsivertinti asmenines integracijos į darbo rinką galimybes.

* Šie moduliai vykdant tęstinį profesinį mokymą neįgyvendinami, o darbuotojų saugos ir sveikatos bei saugaus elgesio ekstremaliose situacijose mokymas integruojamas į kvalifikaciją sudarančioms kompetencijoms įgyti skirtus modulius.

3. REKOMENDUOJAMA MODULIŲ SEKA

Valstybinis kodas	Modulio pavadinimas	LTKS lygis	Apimtis mokymosi kreditais	Asmens pasirengimo mokytis modulyje reikalavimai
Įvadinis modulis (iš viso 1 mokymosi kreditai)*				
4000005	Įvadas į profesiją	IV	1	<i>Netaikoma</i>
Bendrieji moduliai (iš viso 4 mokymosi kreditai)*				
4102201	Saugus elgesys ekstremaliose situacijose	IV	1	<i>Netaikoma</i>
4102105	Sąmoningas fizinio aktyvumo reguliavimas	IV	1	<i>Netaikoma</i>
4102203	Darbuotojų sauga ir sveikata	IV	2	<i>Netaikoma</i>
Kvalifikaciją sudarančioms kompetencijoms įgyti skirti moduliai (iš viso 45 mokymosi kreditai)				
<i>Privalomieji (iš viso 25 mokymosi kreditai)</i>				
406131343	Vartotojo sąsajos projektavimas ir programavimas	IV	10	<i>Netaikoma</i>
406121239	Nesudėtingų duomenų bazių projektavimas ir kūrimas	IV	5	<i>Netaikoma</i>
406131344	Programavimo aplinkos ir programinės įrangos kūrimo proceso valdymas	IV	10	<i>Netaikoma</i>
<i>Privalomai pasirenkamieji (iš viso 20 mokymosi kreditų)</i>				
406131345	Informacinių sistemų projektavimas ir kūrimas Java programavimo kalba	IV	20	<i>Netaikoma</i>
406131346	Informacinių sistemų projektavimas ir kūrimas JavaScript programavimo kalba	IV	20	<i>Netaikoma</i>
406131347	Informacinių sistemų projektavimas ir kūrimas C# programavimo kalba	IV	20	<i>Netaikoma</i>
406131348	Informacinių sistemų projektavimas ir kūrimas PHP programavimo kalba	IV	20	<i>Netaikoma</i>
406131349	Informacinių sistemų projektavimas ir kūrimas Python programavimo kalba	IV	20	<i>Netaikoma</i>
406131350	Žaidimų programavimas naudojant Unity platformą	IV	20	<i>Netaikoma</i>
Pasirenkamieji moduliai (iš viso 5 mokymosi kreditai)*				
406131351	Vartotojo sąsajos kūrimas naudojant TypeScript programavimo kalbą	IV	5	<i>Baigtas šis modulis: Vartotojo sąsajos projektavimas ir programavimas</i>
406131352	Dirbtinio intelekto taikymas programinės įrangos kūrimo ir testavimo procesuose	IV	5	<i>Baigtas šis modulis: Vartotojo sąsajos projektavimas ir programavimas</i>

Baigiamasis modulis (iš viso 5 mokymosi kreditai)				
4000004	Įvadas į darbo rinką	IV	5	<i>Baigti visi jaunesniojo programuotojo kvalifikaciją sudarantys privalomieji moduliai.</i>

* Šie moduliai vykdant tęstinį profesinį mokymą neįgyvendinami, o darbuotojų saugos ir sveikatos bei saugaus elgesio ekstremaliose situacijose mokymas integruojamas į kvalifikaciją sudarančioms kompetencijoms įgyti skirtus modulius.

4. REKOMENDACIJOS DĖL PROFESINEI VEIKLAI REIKALINGŲ BENDRŪJŲ KOMPETENCIJŲ UGDYMO

Bendrosios kompetencijos	Bendrųjų kompetencijų pasiekimą iliustruojantys mokymosi rezultatai
Raštingumo kompetencija	Rašyti gyvenimo ir profesinės patirties aprašymą, motyvacinį laišką, prašymą, ataskaitą, elektroninį laišką. Bendrauti vartojant profesinius terminus.
Daugiakalbystės kompetencija	Vartoti pagrindines profesinės terminijos sąvokas užsienio kalba. Bendrauti profesine užsienio kalba darbinėje aplinkoje žodžiu ir raštu. Skaityti ir kurti profesinę dokumentaciją užsienio kalba.
Matematinė kompetencija ir gamtos mokslų, technologijų ir inžinerijos kompetencija	Suvokti skaičiavimo sistemas, matematinės logikos principus. Taikyti matematinę logiką grįstus algoritmus ir objektinio programavimo principus. Suvokti techninės užduoties reikalavimus ir įvertinti vartotojo reikalavimus. Parengti užduoties techninę specifikaciją. Išmanyti finansinį raštingumą. Suvokti žmogaus veiklos daromą poveikį gamtai ir piliečio asmeninę atsakomybę. Taikyti saugumo ir aplinkos tvarumo principus.
Skaitmeninė kompetencija	Suvokti kompiuterijos ir interneto veikimo principus. Naudotis interneto paieškos ir komunikavimo sistemomis, dokumentų kūrimo ir redagavimo programomis. Rinkti, apdoroti ir saugoti reikalingą darbui informaciją. Naudotis laiko planavimo ir elektroninio pašto programomis. Rengti paslaugos ir (arba) darbo pristatymą kompiuterinėmis programomis. Naudotis skaitmeninėmis technologijomis atsakingai ir saugiai, apsaugant informaciją, turinį, duomenis ir skaitmeninę tapatybę.
Asmeninė, socialinė ir mokymosi mokyti kompetencija	Įsivertinti turimas žinias ir gebėjimus. Kitiškai mąstyti ir priimti sprendimus. Tobulinti savo profesines ir bendrąsias kompetencijas. Organizuoti savo mokymąsi. Pritaikyti turimas žinias ir gebėjimus dirbant individualiai ir komandoje. Parengti profesinio tobulėjimo planą.
Pilietiškumo kompetencija	Gebėti bendrauti su klientais ir kolegomis. Valdyti savo psichologines būsenas, pojūčius ir savybes. Spręsti psichologines krizines situacijas. Gerbti save ir kitus, savo šalį ir jos tradicijas. Pristaisyti prie tarptautinės, daugiakultūrinės aplinkos.
Verslumo kompetencija	Suprasti įmonės veiklos tikslus ir principus, verslo aplinką. Panaudojant intuiciją, kūrybiškumą ir analitinius gebėjimus atpažinti su technologijomis susijusias naujas

	<p>rinkos ir vertės kūrimo galimybes. Suprasti socialiai atsakingo verslo principus. Prisiimti atsakomybę, dirbti savarankiškai ir komandoje, planuoti savo laiką.</p>
Kultūrinio sąmoningumo ir raiškos kompetencija	<p>Paaikškinti Europos ir kitų šalių kultūrinius skirtumus, darbo su kitų kultūrų kolegomis ir (arba) klientais ypatumus. Lavinti estetinį požiūrį į aplinką.</p>

5. PROGRAMOS STRUKTŪRA, VYKDANT PIRMINĮ IR TĘSTINĮ PROFESINĮ MOKYMĄ

Kvalifikacija – jaunesnysis programuotojas, LTKS lygis IV	
Programos, skirtos pirminiam profesiniam mokymui, struktūra	Programos, skirtos tęstiniam profesiniam mokymui, struktūra
<i>Ivadinis modulis (iš viso 1 mokymosi kreditas)</i> Įvadas į profesiją, 1 mokymosi kreditas	<i>Ivadinis modulis (0 mokymosi kreditų)</i> –
<i>Bendrieji moduliai (iš viso 4 mokymosi kreditai)</i> Saugus elgesys ekstremaliose situacijose, 1 mokymosi kreditas Sąmoningas fizinio aktyvumo reguliavimas, 1 mokymosi kreditas Darbuotojų sauga ir sveikata, 2 mokymosi kreditai	<i>Bendrieji moduliai (0 mokymosi kreditų)</i> –
<i>Kvalifikaciją sudarančioms kompetencijoms įgyti skirti moduliai (iš viso 45 mokymosi kreditai)</i> <i>Privalomieji moduliai (iš viso 25 mokymosi kreditai)</i> Vartotojo sąsajos projektavimas ir programavimas, 10 mokymosi kreditų Nesudėtingų duomenų bazių projektavimas ir kūrimas, 5 mokymosi kreditai Programavimo aplinkos ir programinės įrangos kūrimo proceso valdymas, 10 mokymosi kreditų <i>Privalomai pasirenkamieji moduliai (iš viso 20 mokymosi kreditų)</i> Informacinių sistemų projektavimas ir kūrimas Java programavimo kalba, 20 mokymosi kreditai Informacinių sistemų projektavimas ir kūrimas JavaScript programavimo kalba, 20 mokymosi kreditai Informacinių sistemų projektavimas ir kūrimas C# programavimo kalba, 20 mokymosi kreditai Informacinių sistemų projektavimas ir kūrimas PHP programavimo kalba, 20 mokymosi kreditai Informacinių sistemų projektavimas ir kūrimas Python programavimo kalba, 20 mokymosi kreditai Žaidimų programavimas naudojant Unity platformą, 20 mokymosi kreditai	<i>Kvalifikaciją sudarančioms kompetencijoms įgyti skirti moduliai (iš viso 45 mokymosi kreditai)</i> <i>Privalomieji moduliai (iš viso 25 mokymosi kreditai)</i> Vartotojo sąsajos projektavimas ir programavimas, 10 mokymosi kreditų Nesudėtingų duomenų bazių projektavimas ir kūrimas, 5 mokymosi kreditai Programavimo aplinkos ir programinės įrangos kūrimo proceso valdymas, 10 mokymosi kreditų <i>Privalomai pasirenkamieji moduliai (iš viso 20 mokymosi kreditų)</i> Informacinių sistemų projektavimas ir kūrimas Java programavimo kalba, 20 mokymosi kreditai Informacinių sistemų projektavimas ir kūrimas JavaScript programavimo kalba, 20 mokymosi kreditai Informacinių sistemų projektavimas ir kūrimas C# programavimo kalba, 20 mokymosi kreditai Informacinių sistemų projektavimas ir kūrimas PHP programavimo kalba, 20 mokymosi kreditai Informacinių sistemų projektavimas ir kūrimas Python programavimo kalba, 20 mokymosi kreditai Žaidimų programavimas naudojant Unity platformą, 20 mokymosi kreditai
<i>Pasirenkamieji moduliai (iš viso 5 mokymosi kreditai)</i> Vartotojo sąsajos kūrimas naudojant TypeScript programavimo kalbą, 5 mokymosi kreditai Dirbtinio intelekto taikymas programinės įrangos kūrimo ir testavimo procesuose, 5 mokymosi kreditai	<i>Pasirenkamieji moduliai (0 mokymosi kreditų)</i> –
<i>Baigiamasis modulis (iš viso 5 mokymosi kreditai)</i>	<i>Baigiamasis modulis (iš viso 5 mokymosi kreditai)</i>

Pastabos

- Vykdamas pirminį profesinį mokymą asmeniui, jaunesniam nei 16 metų ir neturinčiam pagrindinio išsilavinimo, turi būti sudaromos sąlygos mokytis pagal pagrindinio ugdymo programą (*jei taikoma*).
- Vykdamas pirminį profesinį mokymą asmeniui turi būti sudaromos sąlygos mokytis pagal vidurinio ugdymo programą (*jei taikoma*).
- Vykdamas tęstinį profesinį mokymą asmens ankstesnio mokymosi pasiekimai įskaitomi švietimo ir mokslo ministro nustatyta tvarka.
- Tęstinio profesinio mokymo programos modulius gali vesti mokytojai, įgiję andragogikos žinių ir turintys tai pagrindžiantį dokumentą arba turintys neformaliojo suaugusiųjų švietimo patirties.
- Saugaus elgesio ekstremaliose situacijose modulį vedantis mokytojas turi būti baigęs civilinės saugos mokymus pagal Priešgaisrinės apsaugos ir gelbėjimo departamento direktoriaus patvirtintą mokymo programą ir turėti tai pagrindžiantį dokumentą.
- Tęstinio profesinio mokymo programose darbuotojų saugos ir sveikatos mokymas integruojamas į kvalifikaciją sudarančioms kompetencijoms įgyti skirtus modulius. Darbuotojų saugos ir sveikatos mokoma pagal Mokinių, besimokančių pagal pagrindinio profesinio mokymo programas, darbuotojų saugos ir sveikatos programos aprašą, patvirtintą Lietuvos Respublikos švietimo, mokslo ir sporto ministro 2005 m. rugsėjo 28 d. įsakymu Nr. ISAK-1953 „Dėl Mokinių, besimokančių pagal pagrindinio profesinio mokymo programas, darbuotojų saugos ir sveikatos programos aprašo patvirtinimo“. Darbuotojų saugos ir sveikatos mokymą vedantis mokytojas turi būti baigęs darbuotojų saugos ir sveikatos mokymus ir turėti tai pagrindžiantį dokumentą.
- Tęstinio profesinio mokymo programose saugaus elgesio ekstremaliose situacijose mokymas integruojamas pagal poreikį į kvalifikaciją sudarančioms kompetencijoms įgyti skirtus modulius.

6. PROGRAMOS MODULIŲ APRAŠAI

6.1. ĮVADINIS MODULIS

Modulio pavadinimas – „Įvadas į profesiją“

Valstybinis kodas	4000005	
Modulio LTKS lygis	IV	
Apimtis mokymosi kreditais	1	
Kompetencijos	Mokymosi rezultatai	Rekomenduojamas turinys mokymosi rezultatams pasiekti
1. Pažinti profesiją.	1.1. Išmanyti Jaunesniojo programuotojo profesiją ir jos teikiamas galimybes darbo rinkoje.	Tema. <i>Programuotojo profesija, jos specifika ir galimybės darbo rinkoje</i> <ul style="list-style-type: none"> Programuotojo profesija, jos specifika Programuotojo profesijai svarbios kompetencijos ir veiklos sritys Programuotojo vaidmuo kuriant informacinių technologijų produktus (vykdant informacinių technologijų projektus) Programuotojo galimybės darbo rinkoje
	1.2. Suprasti Jaunesniojo programuotojo profesinę veiklą, veiklos procesus, funkcijas ir uždavinius.	Tema. <i>Jaunesniojo programuotojo profesinės veiklos aplinka, procesai ir tikslai</i> <ul style="list-style-type: none"> Jaunesniojo programuotojo profesinės veiklos, aplinka, pagrindiniai veiklos procesai ir tikslai Veiklos, uždaviniai, kuriuos programuotojas atlieka darbo vietoje Gerosios praktikos taisyklės ir jų svarba darbo procesui bei rezultatų kokybei
	1.3. Demonstruoti jau turimus, neformaliuotu ir (arba) savaiminiu būdu įgytus Jaunesniojo programuotojo kvalifikacijai būdingus gebėjimus.	Tema. <i>Jaunesniojo programuotojo modulinė profesinio mokymo programa</i> <ul style="list-style-type: none"> Modulinės Jaunesniojo programuotojo profesinio mokymo programos tikslai ir uždaviniai Mokymosi formos ir metodai, siekiami rezultatai (kompetencijos) ir jų patvirtinimas, mokymosi pasiekimų vertinimo kriterijai Mokymosi įgūdžių demonstravimo formos (metodai) Individualus mokymosi planas Tema. <i>Turimų gebėjimų, įgytų savaiminiu ar neformaliuotu būdu, vertinimas</i> <ul style="list-style-type: none"> Savaiminiu ir (arba) neformaliuotu būdu įgytų Java programuotojo kvalifikacijai būdingų gebėjimų demonstravimas Savaiminiu ir (arba) neformaliuotu būdu įgytų gebėjimų įsivertinimas
Mokymosi pasiekimų vertinimo kriterijai	Siūlomas įvadinio modulio įvertinimas – <i>įskaityta (neįskaityta)</i> .	
Reikalavimai mokymui skirtiems metodiniams ir	<i>Mokymo(si) medžiaga:</i> <ul style="list-style-type: none"> Jaunesniojo programuotojo modulinė profesinio mokymo programa 	

materialiesiems ištekliams	<ul style="list-style-type: none"> • Vadovėliai ir kita mokomoji medžiaga • Testas turimiems gebėjimams vertinti <p><i>Mokymo(si) priemonės:</i></p> <ul style="list-style-type: none"> • Techninės priemonės mokymo(si) medžiagai iliustruoti, vizualizuoti, pristatyti
Reikalavimai teorinio ir praktinio mokymo vietai	<p>Klasė ar kita mokymuisi pritaikyta patalpa su techninėmis priemonėmis mokymo(si) medžiagai pateikti (kompiuteriu ir prieiga prie interneto, vaizdo projektoriumi) ir kompiuteriais, skirtais mokinių darbui.</p> <p>Praktinio mokymo klasė (patalpa), aprūpinta kompiuteriais, prieiga prie interneto, programine įranga.</p>
Reikalavimai mokytojų dalykiniam pasirengimui (dalykinei kvalifikacijai)	<p>Modulį gali vesti mokytojas, turintis:</p> <ol style="list-style-type: none"> 1) Lietuvos Respublikos švietimo įstatyme ir Reikalavimų mokytojų kvalifikacijai apraše, patvirtintame Lietuvos Respublikos švietimo, mokslo ir sporto ministro 2014 m. rugpjūčio 29 d. įsakymu Nr. V-774 „Dėl Reikalavimų mokytojų kvalifikacijai aprašo patvirtinimo“, nustatytą išsilavinimą ir kvalifikaciją; 2) Informatikos mokslų studijų kryptių grupės ar lygiavertį išsilavinimą arba vidurinį išsilavinimą ir jaunesniojo programuotojo ar lygiavertę kvalifikaciją, ne mažesnę kaip 3 metų jaunesniojo programuotojo profesinės veiklos patirtį ir pedagoginių ir psichologinių žinių kurso baigimo pažymėjimą.

6.2. KVALIFIKACIJĄ SUDARANČIOMS KOMPETENCIJOMS ĮGYTI SKIRTI MODULIAI

6.2.1. Privalomieji moduliai

Modulio pavadinimas – „Vartotojo sąsajos projektavimas ir programavimas“

Valstybinis kodas	406131343	
Modulio LTKS lygis	IV	
Apimtis mokymosi kreditais	10	
Asmens pasirengimo mokytis modulyje reikalavimai	<i>Netaikoma</i>	
Kompetencijos	Mokymosi rezultatai	Rekomenduojamas turinys mokymosi rezultatams pasiekti
1. Taikyti programinės įrangos kūrimui naudojamus informatikos principus ir metodus.	1.1. Apibūdinti įvairias skaičiavimo sistemas.	<p>Tema. <i>Skaičiavimo sistemos</i></p> <ul style="list-style-type: none"> • Skaičiavimo sistemų reikšmė mokslui, technikai ir skaitmeninei elektronikai • Įvairios skaičiavimo sistemos • Skaičiavimo sistemų skaičių išreiškimas ir atvirkštinis perėjimas <p>Tema. <i>Informacijos matavimo vienetai</i></p> <ul style="list-style-type: none"> • Bitas, baitas • Informacijos kiekio apskaičiavimas • Informacijos perdavimo greitis
	1.2. Parinkti duomenų struktūras, pagrindinius algoritmus taikant logikos principus.	<p>Tema. <i>Pagrindinės duomenų struktūros</i></p> <ul style="list-style-type: none"> • Masyvai, sąrašai: jų privalumai, trūkumai ir taikymo sritys • Medžiai, grafai: jų privalumai, trūkumai ir taikymo sritys <p>Tema. <i>Duomenų struktūrų pasirinkimas</i></p> <ul style="list-style-type: none"> • Kriterijai, kuriais remiantis reikia pasirinkti tinkamą duomenų struktūrą: duomenų tipas, operacijos, atminties naudojimas ir kt. • Realūs pavyzdžiai: kokias duomenų struktūras pasirinkti kuriant konkrečias programas (pvz., duomenų bazes, paieškos sistemas, žaidimus) <p>Tema. <i>Teiginių logika ir pagrindiniai operatoriai</i></p> <ul style="list-style-type: none"> • Logikos operatoriai: NOT, AND, OR • Operatoriai ir jų veikimo prioritetas • Apskaičiavimo svarba loginių išraiškų suprantamumui <p>Tema. <i>Bazinės logikos taisyklės (pagal SWEBOK)</i></p> <ul style="list-style-type: none"> • Tapatumo, prieštaravimo ir trečiojo atmetimo dėsniai

		<ul style="list-style-type: none"> • De Morgano taisyklės • Loginių išraiškų supaprastinimas
	1.3. Sukurti algoritmų schemas problemoms išspręsti.	<p>Tema. Algoritmų kūrimas ir analizė</p> <ul style="list-style-type: none"> • Algoritmo kūrimo etapai: nuo problemos suformulavimo iki algoritmo testavimo • Algoritmo reprezentacijos būdai: pseudokodas, srauto diagramos <p>Tema. Algoritmų efektyvumas</p> <ul style="list-style-type: none"> • Algoritmo sudėtingumo analizė: laiko ir atminties sudėtingumas, didysis O žymėjimas • Algoritmų optimizavimas: būdai, kaip pagerinti algoritmo efektyvumą
2. Projektuoti ir programuoti žiniatinklio puslapių vartotojo sąsajas.	2.1. Paaiškinti interneto veikimo principus.	<p>Tema. Internetas</p> <ul style="list-style-type: none"> • Interneto veikimo principai • HTTP protokolo veikimo principai ir struktūra • Interneto naršyklė ir jų veikimas <p>Tema. Priegloba</p> <ul style="list-style-type: none"> • Domenų vardų sistema (angl. Domain Name System (DNS)) ir jos veikimo principas • Svetainių talpinimas (angl. hosting) ir jų veikimas
	2.2. Pateikti internetinio puslapio turinį, naudojant kompiuterinę žymėjimo kalbą.	<p>Tema. Hiperteksto ženklavimo kalba - HTML (angl. HyperText Markup Language)</p> <ul style="list-style-type: none"> • HTML istorija ir standartas • HTML elementas, žymė (angl. tag) ir atributas (angl. attribute) • Pagrindinė HTML dokumento struktūra: <!DOCTYPE html>, <html>, <head>, <body> • <head> dalies žymės ir jų svarba: <title>, <meta>, <link> • Pagrindiniai skirtumai tarp <head> ir <body> žymių • HTML kodo redaktoriai ir jų tipai (tekstiniai, WYSIWYG, internetiniai) • HTML kodo komentavimas <p>Tema. Pagrindinės HTML žymės</p> <ul style="list-style-type: none"> • Antraščių (<h1> ... <h6>) ir pastraipų (<p>, <pre>) žymės • Teksto formatavimo žymės (, , ir t. t.) • Ženklinčių, numeruotų ir sąvokų (angl. description) sąrašų sudarymo žymės (, , , <dl>, <dt>) • Kelių lygių sąrašai (angl. nested list) • Nuorodos žymė ir pagrindiniai atributai (<a>, href, target ir kt.) • Grafinių vaizdų žymė ir pagrindiniai atributai (, src, alt ir kt.) • Lentelių sudarymo žymės ir pagrindiniai atributai (<table>, <tr>, <td>, <th>, <thead>, <tbody>, <tfoot>, <colspan>, <rowspan> ir kt.) • Formos sudarymo žymės ir pagrindiniai atributai (<form>, <input>, <textarea>, <button>,

		<p><select>, <option>, <label>, name, value, type, placeholder, required ir kt.)</p> <ul style="list-style-type: none"> • Įvestų duomenų formose tikrinimas (anlg. validation), naudojant reguliariąsias išraiškas (anlg. regular expressions) • Pagrindiniai skirtumai tarp HTML blokinių (anlg. block) ir įterptinių (anlg. inline) elementų • Grupavimo žymės (<div>,) • HTML multimedijos žymės ir pagrindiniai atributai (<audio>, <video>, controls, <source> ir kt.) • Specialūs simboliai ir jų įterpimo būdai (&copy; &#169; ir pan.) • Naršyklės įrankiai, skirti tinklapių kūrimui ir derinimui (anlg. DevTools) • HTML kodo derinimo (anlg. validation) įrankiai <p>Tema. Tinklapis ir interneto svetainė</p> <ul style="list-style-type: none"> • Semantinės grupavimo žymės ir jų taikymas (<header>, <nav>, <main>, <footer>, <section>, <article>, <aside> ir kt.) • Svetainės (anlg. web site) sudarymas iš atskirų tinklapių (anlg. web pages) • Svetainės failų ir aplankų struktūra • HTML kodo optimizavimo (anlg. minification) tikslas ir įrankiai <p>Tema. Interneto svetainės prieinamumas (anlg. website accessibility)</p> <ul style="list-style-type: none"> • Kas yra skaitmeninis prieinamumas ir kodėl jis svarbus? • Vartotojų grupės (regėjimo, klausos, motorikos ir kt. negalios) ir jų poreikiai • ARIA (Accessible Rich Internet Applications, WAI-ARIA) - atributų rinkinio taikymas svetainių prieinamumui gerinti • WCAG (Web Content Accessibility Guidelines) standartas <p>Tema. Interneto svetainės optimizavimas paieškos sistemoms SEO (Search Engine Optimization)</p> <ul style="list-style-type: none"> • Paieškos sistemų veikimo principai • SEO tipai: On-page SEO, Off-Page SEO, Mobile SEO ir kt. <p>SEO įrankių naudojimas</p>
	<p>2.3. Apipavidalinti internetinį puslapį, naudojant pakopinius stilių šablonus ir karkasus.</p>	<p>Tema. CSS (anlg. Cascading Style Sheets) pagrindai</p> <ul style="list-style-type: none"> • CSS istorija ir standartas • Pagrindinė CSS sintaksė: taisyklė, selektorius, deklaracija, savybė, reikšmė • Numatytieji naršyklės stiliai (anlg. default browser styles) • CSS įterpimo (taikymo) būdai: išorinis (anlg. external stylesheets), vidinis (anlg. internal stylesheets), eilutinis (anlg. inline styles), @import

	<ul style="list-style-type: none"> • CSS taisyklių specifiškumas, taisyklių paveldėjimas ir konfliktai • CSS kodo komentavimas <p>Tema. CSS selektoriai</p> <ul style="list-style-type: none"> • Pagrindiniai selektoriai (angl. basic selectors): universalus (*), žymės (angl. tag), klasės, ID, pseudo klasės, pseudo elementų • Atributų selektoriai (angl. attribute selectors) • Kombinacijos (angl. combinators): palikuonių selektorius (angl. descendant selector), vaikų selektorius (angl. child selector), ~ selektorius (angl. general sibling selector), + selektorius (angl. adjacent sibling selector) • CSS įterpimo (angl. nesting) taikymas tvarkingai kodo struktūrai užrašyti <p>Tema. CSS dėžės modelis</p> <ul style="list-style-type: none"> • Skirtumai tarp blokinių ir eilutinių (angl. inline) elementų • Pagrindinės dėžės modelio savybės: aukštis, plotis, vidinė paraštė (angl. padding), išorinė paraštė (angl. margin), rėmelis (angl. border) ir jų skirtingi reikšmių užrašymo būdai • Savybė box-sizing, jos reikšmės ir įtaka atvaizduojamam elementui • Išorinių paraščių susiliejimas (angl. margin collapsing) • Pagrindinės savybės display reikšmės: block, inline, inline-block ir none <p>Tema. CSS matavimo vienetai</p> <ul style="list-style-type: none"> • Absoliutūs ilgio vienetai • Santykiniai ilgio vienetai • Savybių reikšmės: skaičiai, procentai, spalva, funkcijos, pozicijos, matematinės funkcijos • Vaizdinės srities vienetai (angl. viewport units) <p>Tema. Pagrindinės CSS savybės ir reikšmės</p> <ul style="list-style-type: none"> • Spalvos, jų užrašymo būdai kodais ir funkcijomis (rgb(), rgba(), hsl(), hsla() ir t. t.) • Sąrašų stiliaus savybės ir reikšmės • Tipografijos stilių savybės ir reikšmės • Fono stiliaus savybės ir reikšmės (spalvos, grafiniai vaizdai, gradientai) • Rėmelio stilių savybės ir reikšmės • Lentelių stilių savybės ir reikšmės • Formos ir jos elementų stilių savybės ir reikšmės • Grafinių vaizdų stilių savybės ir reikšmės • Perėjimų ir animacijos savybės ir reikšmės (angl. transitions and animations) • Blokinių (angl. block) ir eilutinių (angl. inline) elementų vertikalus ir horizontalus centravimas
--	---

		<ul style="list-style-type: none"> • Pasirinktinės savybės (kintamieji) (angl. custom properties) • Išorinių stilių įterpimas ir taikymas (šriftų rinkiniai) • CSS kodo derinimo įrankiai <p>Tema. Tinklapio maketavimo savybės ir jų rinkiniai</p> <ul style="list-style-type: none"> • Savybės float, clear ir jų reikšmės • Savybė position ir jos reikšmės • Flexbox modelis • Grid modelis • Media queries taikymas - užtikrinantis tinklapio pritaikomumą įvairiuose įrenginiuose • CSS karkasų (angl. frameworks) taikymas (Bootstrap, Tailwind) • CSS kodo optimizavimo (angl. minification) tikslas ir įrankiai <p>Tema. Išplėstinės CSS galimybės</p> <ul style="list-style-type: none"> • Preprocesorių (pvz. Sass, LESS) sintaksė, kintamieji, funkcijos • Įdėtinės taisyklės (angl. nesting) ir mixins naudojimas • Automatinis CSS generavimas ir optimizacija
	<p>2.4. Valdyti JavaScript programavimo kalbos įrankius ir sintaksę.</p>	<p>Tema. JavaScript programavimo aplinka</p> <ul style="list-style-type: none"> • JavaScript programavimo aplinka ir įrankiai • IDE (instaliavimas, konfigūravimas) <p>Tema. JavaScript programavimo kalba</p> <ul style="list-style-type: none"> • JavaScript programavimo kalbos istorija, standartas, versijos • Java ir JavaScript programavimo kalbų panašumai ir skirtumai • JavaScript programavimo kalbos sintaksė • Strict mode funkcija, jos paskirtis, taikymo pavyzdžiai • JavaScript programavimo kalbos kodo komentarai • JavaScript kodo derinimo įrankiai (IDE, naršyklėje) <p>Tema. JavaScript programavimo kalbos pagrindai</p> <ul style="list-style-type: none"> • Kintamieji, jų deklaravimas (let, var const), galiojimo sritis (angl. scope) ir savybė hoisting • Bendrosios kintamųjų vardų kūrimo taisyklės ir gerosios praktikos • Duomenų tipai: primityvus ir objektai • Duomenų tipo konvertavimas (angl. Data type conversion) • Operatoriai: priskyrimo, palyginimo, aritmetiniai, loginiai, sąlyginiai • Sąlyginiai sakiniai ir klaidų apdorojimas (angl. Control flow and error handling): if...else, switch, throw, try...catch

		<ul style="list-style-type: none"> • Ciklai ir iteracijos (angl. Loops and iteration): for, while, do-while, continue, break, for-in, for-of • Algoritmai. Minimalios reikšmės paieška, maksimalios reikšmės paieška, kiekio (angl. count) skaičiavimo algoritmas, paieškos algoritmai • Duomenų saugojimas naršyklėje (angl. Local Storage) ir pagrindiniai metodai darbui su duomenimis: setItem(key, value), getItem(key), removeItem(key), clear(), key(index) ir savybė length
	<p>2.5. Kurti nesudėtingą programinį kodą JavaScript programavimo kalba.</p>	<p>Tema. <i>Funkcijos</i></p> <ul style="list-style-type: none"> • Funkcijų užrašymas ir pagrindiniai skirtumai: function declaration, function expressions • Vardinės ir anoniminės funkcijos (angl. named and anonymous functions) • Bendrosios funkcijų vardų kūrimo taisyklės ir gerosios praktikos • Funkcijų parametrai ir gražintinos reikšmės (angl. parameters and return values) • Funkcijų iškvietimas ir funkcijų savybė hoisting • Funkcijų galiojimo sritis (angl. function scope) • Funkcijų vykdymo eiliškumas (angl. function stack) • Įterptinės ir išorinės funkcijos ir jų veikimo principai (angl. inner and outer functions). Closures savybė • IIFE funkcijos (angl. Immediately Invoked Function Expression) • Callback funkcijos: asinchroninės operacijos, aukštesniosios eilės funkcijos (angl. Higher-Order Functions), neblokuojantis vykdymas (angl. Non-Blocking Execution) • Funkcijų argumentai (arguments object) • Funkcijų parametrai (default parameters, rest parameters) • Rodyklinės funkcijos (angl. Arrow functions) • JavaScript moduliai <p>Tema. <i>Integruotas objektas Math</i></p> <ul style="list-style-type: none"> • Statiniai metodai: Math.abs(), Math.ceil(), Math.floor(), Math.max(), Math.min(), Math.pow(), Math.random(), Math.round(), Math.sqrt(), Math.trunc() ir kt. • Statinės savybės: Math.PI, Math.E ir kt. <p>Tema. <i>Integruotas objektas Date</i></p> <ul style="list-style-type: none"> • Date objekto skirtingi sukūrimo būdai: new Date(), new Date(milliseconds), new Date(year, month, day, hour, minute, second, millisecond) ir kt. • Statiniai metodai: Date.now(), Date.parse() ir kt. • Metodai: getDate(), getDay(), getFullYear(), getHours(), getMilliseconds(), getMinutes(), getMonth(), getSeconds(), setDate(), setFullYear(), setHours(), setMilliseconds(),

		<p>setMinutes(), setMonth(), setSeconds() ir kt.</p> <p>Tema. Masyvai</p> <ul style="list-style-type: none"> • Masyvų apibrėžimas, svarba ir taikymas JavaScript programavimo kalboje • Įvairūs masyvų kūrimo būdai: masyvo raidinis užrašas (angl. Array Literal Notation), konstruktorius, masyvo savybė length • Masyvo elementų pridėjimas ir pašalinimas: push(), unshift(), pop(), shift(), splice(), slice() • Masyvo elementų paieška: indexOf(), lastIndexOf(), includes(), find(), findIndex() • Transformacija ir filtravimas: map(), filter(), reduce(), reduceRight(), every(), some(), forEach(), concat() • Pažangiosios masyvų technikos: masyvo destrukūrizavimas (angl. Array Destructuring), operatorius rest <p>Tema. Tekstinė eilutė</p> <ul style="list-style-type: none"> • Tekstinės eilutės (angl. String) apibrėžimas ir skirtingi sukūrimo būdai, naudojant dvigubas, viengubas kabutes ir backtick kabutes • Tekstinės eilutės savybė length • Tekstinės eilutės paieškos metodai: indexOf(), lastIndexOf(), includes(), startsWith(), endsWith(), search() • Tekstinės eilutės modifikavimo metodai: replace(), replaceAll(), trim(), slice(), substring(), split() • Tekstinės eilutės suliejimo metodai: concat() ir kt. • Tekstinės eilutės metodai: toUpperCase(), toLowerCase(), charAt(), charCodeAt(), repeat() <p>Tema. Objektai</p> <ul style="list-style-type: none"> • Objektų apibrėžimas ir jų svarba JavaScript programavimo kalboje • Įvairūs objektų kūrimo būdai: objekto raidinis užrašas (angl. Object Literal Notation), raktažodis new, konstruktorius • Objekto savybių pridėjimas ir keitimas • Prieiga prie objektų savybių: taškinis ir skliaustinis užrašas (angl. Dot notation, bracket notation) • Objekto savybių šalinimas • Metodai objektuose: funkcijos kaip objekto savybės • Raktažodis this, elgsenos skirtumai, naudojant raktažodį this skirtinguose kontekstuose • Objekto metodai: Object.keys(), Object.values(), Object.entries(), Object.assign()
--	--	--

		<ul style="list-style-type: none"> • Pažangiosios objektų technikos: objekto destrukūrizavimas (angl. Object Destructuring), paskirstymo sintaksė (angl. Spread Syntax) • Prototipo grandinė (angl. prototype chain): savybių ir metodų paveldėjimas iš kitų objektų • JavaScript Object Notation (JSON): struktūra, sintaksė, palyginimas su kitais formatais • JSON.parse() ir JSON.stringify() funkcijos <p>Tema. <i>Asinchroninis programavimas</i></p> <ul style="list-style-type: none"> • Asinchroninio programavimo apibrėžimas, palyginimas su sinchroniniu programavimu, taikymo pavyzdžiai • Callback funkcijos ir Callback hell situacijos sprendimo būdai • Promises, Async/Await • Įvykių ciklo supratimas (angl. Event Loop) <p>Tema. <i>API (Application Programming Interface) pagrindai</i></p> <ul style="list-style-type: none"> • API apibrėžimas ir taikymo pavyzdžiai • API tipai: RESTful, SOAP, GraphQL, Websocket • HTTP protokolas: GET, POST, PUT, DELETE metodai • Representational State Transfer (REST) architektūra: pagrindiniai principai • URI struktūra: efektyvių URL kūrimas • Duomenų perdavimas, naudojant JSON • Fetch API - užklausų siuntimas ir gavimas • Axios biblioteka - alternatyva Fetch API • Duomenų Manipuliavimas su API: duomenų siuntimas, gavimas, klaidų apdorojimas • Viešų API integracija (pvz.: orų, knygų, filmų ir pan.)
	<p>2.6. Valdyti HTML elementų medį, naudojant JavaScript programavimo kalbą (DOM).</p>	<p>Tema. <i>Dokumento objekto modelis (DOM)</i></p> <ul style="list-style-type: none"> • DOM apibrėžimas, tinklapio atvaizdavimas DOM struktūroje • DOM mazgai, elementai, atributai ir teksto mazgai • Elementų išrinkimo metodai DOM medyje: getElementById(), getElementsByClassName(), getElementsByTagName(), querySelector(), querySelectorAll() ir kt. • Navigavimas DOM medžiu: prieiga prie tėvų, vaikų ir brolių ir seserų mazgų, naudojant tokias savybes kaip parentNode, childNodes, nextSibling ir previousSibling • Naujų DOM elementų sukūrimas ir pridėjimas į DOM medį: createElement(), appendChild(), insertBefore(), replaceChild() • DOM elementų šalinimas: remove(), removeChild() • Turinio įterpimas į DOM elementus: textContent, innerHTML

		<ul style="list-style-type: none"> • Stiliaus ir atributų įterpimas: style savybė, setAttribute(), getAttribute() <p>Tema. Įvykiai</p> <ul style="list-style-type: none"> • Įvykiai: Click, Hover, Submit, Load, Keypress • Pagrindiniai formų įvykiai: submit, focus, change, blur, event.preventDefault() • Įvykių vykdymo fazės (pvz.: angl. Event Bubbling)
	<p>2.7. Kurti vartotojo sąsajas, naudojant JavaScript programavimo kalbos karkasus.</p>	<p>Tema. Vartotojo sąsajos kūrimo karkasai</p> <ul style="list-style-type: none"> • Vartotojo sąsajos kūrimo karkasų apžvalga, privalumai ir trūkumai (pvz.: React, Vue, Angular ir kt.) • Integruota programavimo (kūrimo) aplinka, jos diegimas ir konfigūravimas (angl. IDE) • Statinio kodo analizės įrankiai (pvz. ESLint) • Kodo formatavimo įrankiai (pvz. Prettier) • Greitas naujo projekto kūrimas naudojant įrankius (pvz.: npx, create-react-app) ir skirtingus šablonus (pvz.: typescript) • Projekto paruošimas (angl. build) vykdymui ir aplikacijos paleidimas naršyklėje (pvz.: npm start) • Pagalbiniai naršyklės įrankiai, skirti programinio kodo derinimui (pvz.: Chrome React Dev Tools) • Projekto failų struktūra (pvz.: node_modules, public, src, package.json, package-lock.json, README.md) • Bibliotekų versijavimas (package.json ir package-lock.json) • Dokumentavimas (README.md) • Individualizuoti aplinkos kintamieji (angl. Environment Variables) • Komponento sąvoka (angl. component), logika ir išvaizda • Funkciniai ir klasės komponentai, jų paskirtis ir pagrindiniai skirtumai • JavaScript sintaksės praplėtimas JSX (JavaScript XML) • Stiliaus taikymas: stiliaus įterpimas į programinį kodą (angl. inline), stiliaus importavimas iš failo, paveikslėlių, šriftų ir failų importavimas (angl. static assets), bibliotekų naudojimas (pvz. Bootstrap, Material UI) <p>Tema. Komponentų panaudojimas</p> <ul style="list-style-type: none"> • Parametrų (angl. props) perdavimas tarp komponentų • Perteklinis sudėtingas duomenų perdavimas (angl. prop drilling) ir jo vengimo būdai • Sąlyginis komponentų generavimas (angl. conditional rendering), gerosios praktikos, nesugeneruoto komponento gražinimas (angl. return NULL) • Komponentų generavimas (angl. render) iš duomenų rinkinio (pvz. masyvo)

		<ul style="list-style-type: none"> • Komponentų būsenos ir jų valdymas (pvz. React useState) • Konteksto kūrimas ir panaudojimas duomenų perdavimui tarp komponentų (pvz. React useContext) • Duomenų išlaikymas tarp komponentų pergeneravimo (pvz. React useRef) • Komponentų sujungimas ir sinchronizavimas su išorinėmis sistemomis (pvz. React useEffect) • Komponentų pergeneravimo optimizavimas (pvz. React useMemo) • Individualizuotų komponentų kūrimas (angl. custom hooks) <p>Tema. Vartotojo interakcijos su aplikacija</p> <ul style="list-style-type: none"> • Vartotojo sąveikos su aplikacija veiksmai (pvz.: onClick, onSubmit, onChange) • Darbas su įvesties formomis (pvz. React-hook-form) • Navigacija (angl. multiple pages routes) • Dinaminiai ir statiniai keliai (angl. routes) <p>Tema. Darbas su duomenimis</p> <ul style="list-style-type: none"> • Duomenų gavimas iš API (angl. Application Programming Interface) • CRUD (create, read, update, delete) metodų panaudojimas • Duomenų filtravimas ir rikiavimas, puslapiavimas <p>Tema. Vartotojo sąsajos testavimas</p> <ul style="list-style-type: none"> • Unit testų rašymas ir paleidimas • Testavimo karkasas (pvz. Jest)
Mokymosi pasiekimų vertinimo kriterijai	<p>Apibūdintos įvairios skaičiavimo sistemos. Parengtos tinkamos duomenų struktūros, pagrindiniai algoritmai taikant logikos principus. Sukurtos algoritmų schemas problemoms išspręsti. Paaiškinti interneto veikimo principai. Pateiktas internetinio puslapio turinys, naudojant kompiuterinę žymėjimo kalbą. Apipavidalintas internetinis puslapis, naudojant pakopinius stilių šablonus ir karkasus. Pademonstruotas JavaScript programavimo kalbos sintaksės ir įrankių valdymas. Sukurtas nesudėtingas programinis kodas JavaScript programavimo kalba. Pademonstruotas HTML elementų medžio valdymas (DOM) JavaScript programavimo kalba. Sukurta vartotojo sąsaja, naudojant JavaScript programavimo kalbos karkasą. Atliekant darbus laikomasi ergonomikos, saugaus darbo, priešgaisrinės saugos, darbo higienos ir aplinkosaugos reikalavimų, užtikrinančių IT specialistų sveikatą, produktyvumą ir saugią darbo aplinką.</p>	
Reikalavimai mokymui skirtiems metodiniams ir materialiesiems ištekliams	<p><i>Mokymo(si) medžiaga:</i></p> <ul style="list-style-type: none"> • Vadovėliai ir kita mokomoji medžiaga • Testas turimiems gebėjimams vertinti <p><i>Mokymo(si) priemonės:</i></p> <ul style="list-style-type: none"> • Techninės priemonės mokymo(si) medžiagai iliustruoti, vizualizuoti, pristatyti 	
Reikalavimai teorinio ir	<p>Klasė ar kita mokymuisi pritaikyta patalpa su techninėmis priemonėmis mokymo(si) medžiagai pateikti (kompiuteriu ir prieiga prie</p>	

praktinio mokymo vietai	interneto, vaizdo projektoriumi) ir kompiuteriais, skirtais mokinių darbui. Praktinio mokymo klasė (patalpa), aprūpinta kompiuteriais, prieiga prie interneto, programine įranga.
Reikalavimai mokytojų dalykiniam pasirengimui (dalykinei kvalifikacijai)	Modulį gali vesti mokytojas, turintis: 1) Lietuvos Respublikos švietimo įstatyme ir Reikalavimų mokytojų kvalifikacijai apraše, patvirtintame Lietuvos Respublikos švietimo, mokslo ir sporto ministro 2014 m. rugpjūčio 29 d. įsakymu Nr. V-774 „Dėl Reikalavimų mokytojų kvalifikacijai aprašo patvirtinimo“, nustatytą išsilavinimą ir kvalifikaciją; 2) Informatikos mokslų studijų kryptių grupės ar lygiavertį išsilavinimą arba vidurinį išsilavinimą ir jaunesniojo programuotojo ar lygiavertę kvalifikaciją, ne mažesnę kaip 3 metų jaunesniojo programuotojo profesinės veiklos patirtį ir pedagoginių ir psichologinių žinių kurso baigimo pažymėjimą.

Modulio pavadinimas – „Nesudėtingų duomenų bazių projektavimas ir kūrimas“

Valstybinis kodas	406121239	
Modulio LTKS lygis	IV	
Apimtis mokymosi kreditais	5	
Asmens pasirengimo mokyti modulyje reikalavimai	<i>Netaikoma</i>	
Kompetencijos	Mokymosi rezultatai	Rekomenduojamas turinys mokymosi rezultatams pasiekti
1. Projektuoti tipines reliacines ir nereliacines (NoSQL) duomenų bazes.	1.1. Apibūdinti duomenų bazių tipus, skirtumus ir taikymo galimybes.	Tema. Duomenų bazės <ul style="list-style-type: none"> • Duomenų bazės sąvoka • Vartotojui prieinamos standartinės operacijos, DBS ir DBVS sąvokos • DBVS (duomenų bazių valdymo sistema) paskirtis, vaidmuo • DBVS modeliai – sąryšinis, hierarchinis, tinklinis • RDBVS ir pagrindinės sistemos sąvokos – lentelė, stulpelis, eilutė ir laukas • Reliacinių sistemų principai Tema. Duomenų bazės pasirinkimas <ul style="list-style-type: none"> • Populiarios DBVS: atvirojo ir uždarojo kodo, reliacinės ir nereliacinės • SQL kalbos apibrėžimas, sakinių grupės – DDL, DML, DCL ir TCL
	1.2. Kurti reliacinės duomenų bazės schemą, naudojant programinę įrangą.	Tema. Duomenų bazių kūrimas <ul style="list-style-type: none"> • Duomenų tipai: tekstas, skaičiai, dvejetainiai duomenys, data ir laikas. ISO standartai • Lentelių kūrimas, naudojant CREATE sakinį, struktūros bei duomenų apribojimo sąvokos • ALTER table sakiny • Pirminis raktas

		<ul style="list-style-type: none"> • Antrinis raktas ir jo prasmė įvedus pirminį raktą • Išorinis raktas • Indeksai <p>Tema. Duomenų bazių projektavimas</p> <ul style="list-style-type: none"> • Projektavimo prasmė ir proceso žingsniai (paskirties numatymas, lentelių apibrėžimas, stulpelių apibrėžimas, pirminių ir antrinių raktų nustatymas) • Normalinių formų prasmė, pirma, antra bei trečia normalinės formos • Ryšių tarp lentelių tipai, daug:daug ryšio pašalinimas reliacinėse sistemose • Projekto tobulinimas: galimos struktūros klaidos
	1.3. Kurti nereliacinės (NoSQL) duomenų bazės schemą, naudojant programinę įrangą.	<p>Tema. Įvadas į nereliacines duomenų bazes</p> <ul style="list-style-type: none"> • Nereliacinės duomenų bazės sąvoka, privalumai ir trūkumai palyginus su RDB • Dokumentinės duomenų bazės • Rakto – reikšmės duomenų bazės • Stulpelinės (šeimos) duomenų bazės • Grafų (angl. graph) duomenų bazės <p>Tema. Nereliacinių duomenų bazių kūrimas ir projektavimas</p> <ul style="list-style-type: none"> • Nereliacinės duomenų bazės struktūros kūrimas • Sukurtos schemos ypatumai (unikalaus įrašo apibrėžimas, galimos struktūros klaidos)
2. Naudoti ir integruoti duomenų bazes.	2.1. Apibūdinti duomenų bazių sistemas, jų diegimą ir valdymą.	<p>Tema. Duomenų bazių diegimas</p> <ul style="list-style-type: none"> • DBVS (MySQL, H2, T-SQL, Postgre, SQLite ir pan.) • DBVS sistemos paleidimo ypatumai (URL, užimti prievadai, reikalingos teisės) • Naujos duomenų bazės sukūrimas, ypatumai (raidynas, jautra), fizinė vieta failų sistemoje <p>Tema. Duomenų bazių eksportavimas ir importavimas</p> <ul style="list-style-type: none"> • Sukurtos duomenų bazės kopijos darymas, eksportavimas • Duomenų bazės importavimas, naudojant SQL failus
	2.2. Kurti užklausas, naudojant SQL ir NoSQL kalbas duomenų bazės užpildymui ir informacijos išrinkimui.	<p>Tema. Baziniai SQL sakiniai</p> <ul style="list-style-type: none"> • SELECT sakiny • INSERT sakiny • UPDATE sakiny • DELETE sakiny <p>Tema. Sudėtingesni SQL sakiniai</p> <ul style="list-style-type: none"> • Platesnis WHERE sąlygos panaudojimas (<>, dviejų stulpelių palyginimas, BETWEEN..AND, IN, LIKE, NULL) • NOT, AND/OR sakiniai, jų pirmenybės ir apskliaudimas

		<ul style="list-style-type: none"> • SQL rikiavimo sakinyss ORDER BY • Papildomos SQL funkcijos (DISTINCT, ILIKE, LOWER, UPPER, CONCAT, matematiniai veiksmai) • SQL grupavimo sakinyss GROUP BY (grupinės (agregatinės) funkcijos, COUNT(*) žvaigždutės prasmė, sąlygos sakinyss HAVING) • Bendra SQL SELECT sintaksė, apdorojamų duomenų apribojimo sakinyss • Lentelių jungimo prasmė, reliacinio modelio skaidymo naudingumas • Lentelių jungimo sąvokos, jungimas, naudojant WHERE sąlygą ir jo trūkumai • Lentelių jungimas, naudojant JOIN sąlygą, trijų lentelių jungimo sintaksė • LEFT ir RIGHT JOIN jungimo tipų prasmė, FULL OUTER JOIN • Lentelių jungimas pačių su savimi • Struktūrinės užklausos ir jų panaudojimas • Standartinis duomenų bazių optimizavimas: JOIN vs. SubQuery <p>Tema. CRUD sakinių panaudojimas nereliacinėse duomenų bazėse</p> <ul style="list-style-type: none"> • Duomenų išrinkimo metodas (pvz.: MongoDB find metodas) • Duomenų įterpimo metodas (pvz.: MongoDB insert metodas) • Duomenų atnaujinimo metodas (pvz.: MongoDB update metodas) • Duomenų šalinimo metodas (pvz.: MongoDB delete metodas) <p>Tema. Transakcijos</p> <ul style="list-style-type: none"> • Transakcijos sąvoka • COMMIT ir ROLLBACK sakiniai • Izoliacijos sąvoka • Veikimo režiai • Paskirstytos (angl. distributed) transakcijos (pvz. 2PC, 3PC, XA)
	2.3.Administruoti duomenų bazes, naudojant gerąsias duomenų bazių administravimo praktikas.	<p>Tema. Duomenų bazių vartotojai</p> <ul style="list-style-type: none"> • Naujo vartotojo kūrimas • GRANT sakinyss, jo lygiai <p>Tema. Duomenų bazių administravimas</p> <ul style="list-style-type: none"> • Administratoriaus teisės • Sistemos rolės
Mokymosi pasiekimų vertinimo kriterijai	Apibūdinti duomenų bazių tipai, skirtumai ir taikymo galimybės. Sukurtos reliacinės duomenų bazės schemas, naudojant programinę įrangą. Sukurtos nereliacinės duomenų bazės schemas, naudojant programinę įrangą. Apibūdintos duomenų bazių sistemos, jų diegimas ir valdymas. Sukurtos užklausos duomenų bazės užpildymui ir informacijos išrinkimui, naudojant SQL ir NoSQL kalbas. Pademonstruotas duomenų bazių administravimas. Atliekant darbus laikomasi ergonomikos, saugaus darbo,	

	priešgaisrinės saugos, darbo higienos ir aplinkosaugos reikalavimų, užtikrinančių IT specialistų sveikatą, produktyvumą ir saugią darbo aplinką.
Reikalavimai mokymui skirtiems metodiniams ir materialiesiems ištekliams	<p><i>Mokymo(si) medžiaga:</i></p> <ul style="list-style-type: none"> • Vadovėliai ir kita mokomoji medžiaga • Testas turimiems gebėjimams vertinti <p><i>Mokymo(si) priemonės:</i></p> <ul style="list-style-type: none"> • Techninės priemonės mokymo(si) medžiagai iliustruoti, vizualizuoti, pristatyti
Reikalavimai teorinio ir praktinio mokymo vietai	Klasė ar kita mokymuisi pritaikyta patalpa su techninėmis priemonėmis mokymo(si) medžiagai pateikti (kompiuteriu ir prieiga prie interneto, vaizdo projektoriumi) ir kompiuteriais, skirtais mokinių darbui. Praktinio mokymo klasė (patalpa), aprūpinta kompiuteriais, prieiga prie interneto, programine įranga.
Reikalavimai mokytojų dalykiniam pasirengimui (dalykinei kvalifikacijai)	Modulį gali vesti mokytojas, turintis: 1) Lietuvos Respublikos švietimo įstatyme ir Reikalavimų mokytojų kvalifikacijai apraše, patvirtintame Lietuvos Respublikos švietimo, mokslo ir sporto ministro 2014 m. rugpjūčio 29 d. įsakymu Nr. V-774 „Dėl Reikalavimų mokytojų kvalifikacijai aprašo patvirtinimo“, nustatytą išsilavinimą ir kvalifikaciją; 2) Informatikos mokslų studijų krypties grupės ar lygiavertį išsilavinimą arba vidurinį išsilavinimą ir jaunesniojo programuotojo ar lygiavertę kvalifikaciją, ne mažesnę kaip 3 metų jaunesniojo programuotojo profesinės veiklos patirtį ir pedagoginių ir psichologinių žinių kurso baigimo pažymėjimą.

Modulio pavadinimas – „Programavimo aplinkos ir programinės įrangos kūrimo proceso valdymas“

Valstybinis kodas	406131344	
Modulio LTKS lygis	IV	
Apimtis mokymosi kreditais	10	
Asmens pasirengimo mokytis modulyje reikalavimai	<i>netaikoma</i>	
Kompetencijos	Mokymosi rezultatai	Rekomenduojamas turinys mokymosi rezultatams pasiekti
1. Naudoti tarnybinių stočių operacines sistemas.	1.1. Apibūdinti serverių veikimo principus ir panaudojimo galimybes.	<p>Tema. Serveriai ir jų operacinės sistemos</p> <ul style="list-style-type: none"> • Serveris ir jo paskirtis • Serverio struktūriniai komponentai • Serverių operacinės sistemos • GNU / Linux • Atviroji programinė įranga <p>Tema. Kompiuterių tinklų pagrindai</p>

		<ul style="list-style-type: none"> • Kompiuterių tinklai • Kompiuterių tinklų charakteristikos (tipai, topologija) • Kompiuterių tinklų modeliai (OSI, TCP/IP) • Kompiuterių tinklo protokolai (HTTP, HTTPS, SSL) • Kliento-serverio architektūra <p>Tema. <i>Serverio OS diegimas</i></p> <ul style="list-style-type: none"> • Virtualizacija • Virtualizacijos privalumai, tipai, technologijos • Hipervizorius, virtualizacijos įrankiai • Debesų kompiuterija (SaaS, PaaS, IaaS) • Virtualbox (diegimas, konfigūracija) • Linux OS virtualizavimas • Programuotojo aplinkos diegimas operacinėje sistemoje • Integruotos programavimo terpės (aplinkos) diegimas ir konfigūravimas <p>Tema. <i>Serverio tekstinės aplinkos funkcionalumas</i></p> <ul style="list-style-type: none"> • Tekstinė vartotojo sąsaja (CLI) • Komandų interpretatoriai (sh, bash ir t. t.) • Komandų eilutės pagalba (man, apropos, info ir t. t.) • Darbas su failų sistema (pwd, ls, cd, stat, file, touch, mkdir, rm, rmdir, cp, mv, ln ir t. t.) • Darbas su failų turiniu (more, less, cat, tac, head, tail, sort, uniq, cut, wc, grep, sed ir t. t.) • Darbas su resursais (whereis, locate, find, du, df, vmstat, free, time, timedatectl, uptime, date, cal, lshw, lspci, lsblk, lsusb ir t. t.) • Darbas su vartotojais, grupėmis (whoami, who, w, useradd, userdel, groupadd, groupdel, passwd, gpasswd, usermod, groups, id ir t. t.) • Darbas su failų teisėmis (chmod, chown, chgrp ir t. t.) • Darbas su procesais (ps, pstree, pgrep, jobs, fg, bg, kill, systemctl, top ir t. t.) • CLI teksto redaktoriai, archyvavimo programos (vi, vim, nano, tar, gzip, zip ir t. t.) <p>Tema. <i>Bash scenarijai, darbų automatizacija</i></p> <ul style="list-style-type: none"> • Bash scenarijus (bash aplinka, kintamieji, scenarijų klaidų kodai) • Bash sąlyginiai sakiniai (if-then, if-then-else, skaičių, eilučių, failų palyginimas) • Bash ciklai (for, while) • Darbų automatizavimas, naudojant bash <p>Tema. <i>Programiniai paketai, paketų repozitorijos</i></p> <ul style="list-style-type: none"> • Programos paketas (deb, rpm ir t. t.)
--	--	--

		<ul style="list-style-type: none"> • Programų paketų diegimas (atnaujinimo procesas, įrankiai) • Programų paketų valdymo sistema • Debian paketų valdymo įrankiai (apt, dpkg) • Programų surinkimas iš išeities kodo
	1.2. Administruoti Web serverio programinę įrangą bylų viešinimui HTTP protokolu.	<p>Tema. <i>WEB serveris ir Aplikacijos serveris</i></p> <ul style="list-style-type: none"> • WEB serveris (nginx, IIS, Apache HTTP Server, ir pan.) • Aplikacijos serveris ir konfigūravimas (Tomcat ir kt.) • Statinių ir dinaminių tinklapių viešinimas <p>Tema. <i>Nuotolinis serverio administravimas</i></p> <ul style="list-style-type: none"> • Nuotolinis administravimas: principai, įrankiai • Nuotolinio administravimo protokolai (telnet, ssh ir t. t.) • OpenSSH: diegimas, konfigūravimas, prisijungimas prie serverio, saugumas • SSH raktai: generavimas, naudojimas
	1.3. Pritaikyti Docker konteinerius bylų viešinimui HTTP protokolu.	<p>Tema. <i>Docker pagrindai</i></p> <ul style="list-style-type: none"> • Docker platforma • Skirtumai tarp virtualizacijos ir konteinerizavimo • Docker: pagrindinės sąvokos, architektūra (bazinės technologijos: Linux containers, Linux namespaces, cgroups ir t. t.) • Docker konteineris: konteinerio gyvavimo ciklas, cli komandos • Docker image: cli komandos, atvaizdo kūrimo pagrindai • Docker network: cli komandos • Docker Desktop (GUI) • Docker WEB (Portainer) <p>Tema. <i>Debesijos (angl. Cloud) pagrindai</i></p> <ul style="list-style-type: none"> • Debesijos platformos • Didžiausių debesijos tiekėjų (AWS, Azure, Google Cloud) apžvalga ir palyginimas • CLI komandos • GUI apžvalga • Tinklo konfigūravimas • Docker image įkėlimas ir paleidimas • FaaS (angl. Function as a Service) principo apžvalga
2. Analizuoti skirtingų tipų reikalavimus, keliamus kuriamai programinei	2.1. Paaiškinti AGILE filosofiją ir jos principus.	<p>Tema. <i>Įvadas į Agile</i></p> <ul style="list-style-type: none"> • Agile manifestas ir principai • Agile ir tradiciniai projekto valdymo metodai, palyginimas, privalumai ir trūkumai

įrangai.		<p>Tema. <i>Projektų valdymo metodikos</i></p> <ul style="list-style-type: none"> • Scrum metodikos privalumai ir trūkumai • Kanban metodikos privalumai ir trūkumai
	2.2.Paaiškinti SCRUM proceso dalis ir komandos narių atsakomybes.	<p>Tema. <i>Scrum pagrindai</i></p> <ul style="list-style-type: none"> • Scrum terminologija (sprint, product backlog, sprint backlog, daily scrum) • Scrum komandos vaidmenys (product owner, scrum master, development team, product owner proxy) • Scrum renginiai (sprint planning, daily scrum, sprint review, sprint retrospective, backlog grooming) <p>Tema. <i>Projekto vykdymas pagal Scrum</i></p> <ul style="list-style-type: none"> • Produkto Backlog sudarymas (užduočių kūrimas, prioretizavimas) • Sprinto planavimas (tikslų nustatymas, užduočių pasiskirstymas) • Sprinto eiga (kasdieniai susitikimai) • Sprinto peržiūra (angl. review), sprinto diagramos (Velocity, Burndown diagrama) • Sprinto retrospektyva (komandinio darbo analizė, sprendimų priėmimas proceso gerinimui)
	2.3.Analizuoti pateiktus techninius reikalavimus, naudojant projekto eigos valdymo principus ir įrankius.	<p>Tema. <i>Programinės įrangos reikalavimų analizė</i></p> <ul style="list-style-type: none"> • Vartotojo poreikių nustatymas ir analizė • Vartotojo reikalavimų analizė kuriant iteracijos užduotis • Vartotojo poreikių ir reikalavimų analizavimas taikant baigtumo kriterijus <p>Tema. <i>Programinės įrangos atitikimas reikalavimams</i></p> <ul style="list-style-type: none"> • Programinės įrangos atitikimo reikalavimams nustatymas • Programinės įrangos atitikimo reikalavimams vertinimas
3.Kurti programinę įrangą individualiai ir dirbant komandoje, taikant programinės įrangos kūrimo metodus.	3.1.Paaiškinti versijų kontrolės svarbą įrangos kūrimo procese.	<p>Tema. <i>Kodo saugojimo ir versijavimo platformos</i></p> <ul style="list-style-type: none"> • Kodo saugojimo ir versijavimo platformų (GitHub, GitLab, Bitbucket ir pan.) pagrindinės funkcijos ir panaudojimo galimybės • Pagrindiniai skirtumai tarp kodo saugojimo ir versijavimo platformos ir versijavimo sistemos Git • Kodo saugojimo ir versijavimo platformos konfigūravimas <p>Tema. <i>Versijavimo sistema Git</i></p> <ul style="list-style-type: none"> • Lokalūs Git sistemos diegimas ir konfigūravimas • .gitignore failas paskirtis ir panaudojimo būdai • Git sistemos veikimo principai • Pagrindinės Git komandos (git status, git add, git commit, git log)

<p>3.2. Derinti komandinį programavimo darbų vykdymą, naudojant versijavimo platformas.</p>	<ul style="list-style-type: none"> • Git commit žinutė: struktūra ir turinys, geroji praktika, pavyzdžiai <p>Tema. Šakojimosi (angl. branching) pagrindai</p> <ul style="list-style-type: none"> • Git šakų kūrimo privalumai ir trūkumai • Pagrindinės šakų kūrimo komandos (git branch, git switch -c, git checkout -b) • Git šakų pavadinimų gerosios praktikos • Git šakų pavadinimų keitimas • Git šakų trynimasis • Peršokimas į kitą šaką (git checkout, git switch) <p>Tema. Lokalus šakų suliejimas ir konfliktai</p> <ul style="list-style-type: none"> • Šakų liejimo (angl. merging) pagrindai • Konfliktų tipai ir sprendimo būdai <p>Tema. Git istorijos keitimas ir peržiūra</p> <ul style="list-style-type: none"> • Pakeitimų atšaukimas (git reset, --soft, --hard, --mixed) • Istorijos perrašymas (git commit --amend, git rebase, git filter-branch, git push --force) • Skirtumų peržiūra: tarp šakų, tarp versijos užfiksavimų (angl. commit) <p>Tema. Bendradarbiavimas kodo saugojimo ir versijavimo platformoje</p> <ul style="list-style-type: none"> • Repozitorijos kūrimas ir konfigūravimas • Klonavimas (angl. cloning) • Šakų kūrimas (angl. branching) kodo saugojimo ir versijavimo platformoje • Šakų liejimas (angl. merging) kodo saugojimo ir versijavimo platformoje • Konfliktai ir jų sprendimas kodo saugojimo ir versijavimo platformoje <p>Tema. Komandinis darbas ir bendravimas</p> <ul style="list-style-type: none"> • Komandos narių pridėjimas prie kodo saugojimo platformos repozitorijos • Darbas su užklausomis (angl. Pull requests) – kūrimas, peržiūra ir patvirtinimas • Bendravimas per kodo saugojimo ir versijavimo platformas: komentarai, žymės • Dokumentacija, projekto README failas paskirtis ir jo kūrimas
<p>3.3. Versijuoti programinį kodą, naudojant versijų kontrolės sistemas.</p>	<p>Tema. Versijavimo strategijos</p> <ul style="list-style-type: none"> • Išleidimo (naujos versijos) šakos kūrimo strategija (angl. Release branching) • Funkcijos šakų kūrimo strategija (angl. Feature branching strategy) • Užduoties šakų kūrimo strategija (angl. Task Branching strategy) • GitFlow srauto šakų strategija (angl. GitFlow branching strategy) • GitLab srauto šakų strategija (angl. GitLab Flow branching strategy) • GitHub srauto šakų strategija (angl. GitHub Flow branching strategy) <p>Tema. Grafinė vartotojo sąsaja</p>

		<ul style="list-style-type: none"> • Repozitorijos klonavimas, komandų commit, push, pull ir t. t. naudojimas, naudojant grafinę vartotojo sąsają (GitKraken, Fork, Git Extensions ir t. t.) • Konfliktų sprendimas, naudojant integruota arba papildoma grafinę vartotojo sąsają (Meld, Sublime Merge, kdiff3 ir t. t.)
Mokymosi pasiekimų vertinimo kriterijai	<p>Apibūdinti serverių veikimo principai ir panaudojimo galimybės. Pademonstruotas Web serverio programinės įrangos valdymas bylų viešinimui HTTP protokolu. Pritaikytas Docker konteineris bylų viešinimui HTTP protokolu. Paašškinta AGILE filosofija ir jos principai. Paašškintos SCRUM proceso dalys ir komandos narių atsakomybės. Išanalizuoti pateikti techniniai reikalavimai, naudojant projekto eigos valdymo principus ir įrankius. Paašškinta versijų kontrolės svarba įrangos kūrimo procese. Suplanuotas komandinis programavimo darbų vykdymas, naudojant versijavimo platformas. Versijuojamas programinis kodas, naudojant versijų kontrolės sistemas. Atliekant darbus laikomasi ergonomikos, saugaus darbo, priešgaisrinės saugos, darbo higienos ir aplinkosaugos reikalavimų, užtikrinančių IT specialistų sveikatą, produktyvumą ir saugią darbo aplinką.</p>	
Reikalavimai mokymui skirtiems metodiniams ir materialiesiems ištekliams	<p><i>Mokymo(si) medžiaga:</i></p> <ul style="list-style-type: none"> • Vadovėliai ir kita mokomoji medžiaga • Testas turimiems gebėjimams vertinti <p><i>Mokymo(si) priemonės:</i></p> <ul style="list-style-type: none"> • Techninės priemonės mokymo(si) medžiagai iliustruoti, vizualizuoti, pristatyti 	
Reikalavimai teorinio ir praktinio mokymo vietai	<p>Klasė ar kita mokymuisi pritaikyta patalpa su techninėmis priemonėmis mokymo(si) medžiagai pateikti (kompiuteriu ir prieiga prie interneto, vaizdo projektoriumi) ir kompiuteriais, skirtais mokinių darbui.</p> <p>Praktinio mokymo klasė (patalpa), aprūpinta kompiuteriais, prieiga prie interneto, programine įranga.</p>	
Reikalavimai mokytojų dalykiniam pasirengimui (dalykinei kvalifikacijai)	<p>Modulį gali vesti mokytojas, turintis:</p> <ol style="list-style-type: none"> 1) Lietuvos Respublikos švietimo įstatyme ir Reikalavimų mokytojų kvalifikacijai apraše, patvirtintame Lietuvos Respublikos švietimo, mokslo ir sporto ministro 2014 m. rugpjūčio 29 d. įsakymu Nr. V-774 „Dėl Reikalavimų mokytojų kvalifikacijai aprašo patvirtinimo“, nustatytą išsilavinimą ir kvalifikaciją; 2) Informatikos mokslų studijų krypties grupės ar lygiavertį išsilavinimą arba vidurinį išsilavinimą ir jaunesniojo programuotojo ar lygiavertę kvalifikaciją, ne mažesnę kaip 3 metų jaunesniojo programuotojo profesinės veiklos patirtį ir pedagoginių ir psichologinių žinių kurso baigimo pažymėjimą. 	

6.2.2. Privalomai pasirenkamieji moduliai

Modulio pavadinimas – „Informacinių sistemų projektavimas ir kūrimas Java programavimo kalba“

Valstybinis kodas	406131345	
Modulio LTKS lygis	IV	
Apimtis mokymosi kreditais	20	
Asmens pasirengimo mokyti modulyje reikalavimai	<i>netaikoma</i>	
Kompetencijos	Mokymosi rezultatai	Rekomenduojamas turinys mokymosi rezultatams pasiekti
1. Kurti tipinę programinę įrangą Java programavimo kalba.	1.1. Apžvelgti Java programavimo kalbos atsiradimo istoriją ir pagrindines taikymo sritis.	<p>Tema. Įvadas į Java programavimo kalbą</p> <ul style="list-style-type: none"> • Programavimo kalbos istorija • Java versijavimas ir licencija • Java kalbos konceptas <p>Tema. Programavimo kalbos taikymo sritys</p> <ul style="list-style-type: none"> • Enterprise programos • Android programos • Web programos
	1.2. Valdyti Java programavimo kalbos įrankius ir sintaksę.	<p>Tema. Java aplinka</p> <ul style="list-style-type: none"> • Java programavimo aplinka ir įrankiai • Sąvokos ir jų paskirtis: Source code, Bytecode, JVM, JRE, JDK • Komandinės eilutės komandos (javac, javadoc, jar..) • Integruota kūrimo aplinka (angl. IDE): instaliavimas, konfigūravimas <p>Tema. Java programavimo pagrindai ir sintaksė</p> <ul style="list-style-type: none"> • Kintamieji (deklaravimas, inicializavimas, literalai) • Bendrosios kintamųjų vardų kūrimo taisyklės ir gerosios praktikos • Primityvūs duomenų tipai. Tipų konvertavimas • Primityvių tipų masyvai (angl. array) • Operatoriai su primitiviais tipais: priskyrimo, palyginimo, aritmetiniai, loginiai • Sąlyginiai sakiniai: If, if-else, switch • Ciklai: while, do-while, for, break, continue • Statiniai metodai: Math.abs(), Math.ceil(), Math.floor(), Math.max(), Math.min(), Math.pow(), Math.random(), Math.round(), Math.sqrt() ir kt.
	1.3. Kurti nesudėtingą programinį	Tema. Metodai

	kodą Java programavimo kalba.	<ul style="list-style-type: none"> • Metodo parametrai ir gražinamas tipas • Metodų perkrova (angl. overloading) • Kintamieji argumentai • Rekursija (angl. recursion) • Algoritmai: minimalios reikšmės paieška, maksimalios reikšmės paieška, kiekio (angl. count) skaičiavimo algoritmas, paieškos algoritmai • Nekartokite savęs (angl. Do not Repeat Yourself (DRY)) principas <p>Tema. Tekstinė eilutė</p> <ul style="list-style-type: none"> • Tekstinės eilutės (angl. String) apibrėžimas ir skirtingi sukūrimo būdai • Tekstinės eilutės metodas length() • Tekstinės eilutės paieškos metodai: indexOf(), lastIndexOf(), contains(), startsWith(), endsWith(), search() • Tekstinės eilutės modifikavimo metodai: substring(), split(), replace(), replaceAll(), trim() • Tekstinės eilutės suliejimo metodai: concat() ir kt. • Kiti tekstinės eilutės metodai: toUpperCase(), toLowerCase(), charCodeAt(), repeat() <p>Tema. Specialūs simboliai</p> <ul style="list-style-type: none"> • Specialių simbolių įterpimas (angl. escape character), naudojant pasvirąjį kairinį brūkšnį (angl. backslash) • Naujos eilutės simboliai: \r, \n, \r\n. System.lineSeparator() <p>Tema. Kodavimo standartai</p> <ul style="list-style-type: none"> • Java Oracle standartai • Dokumentacijos kūrimas naudojant JavaDoc standartą
	1.4. Programuoti Java programavimo kalba, taikant objektinio programavimo principus.	<p>Tema. Objektinis programavimas</p> <ul style="list-style-type: none"> • Funkcinio, procedūrinio ir objektinio programavimo principai • Objektinio programavimo samprata ir abstrakcijos principas • Klasės ir objekto sąvokos • Objekto kintamieji ir objekto metodai • Objektų kūrimas ir panaudojimas • Konstruktoriai • Objektų palyginimas, naudojant equals • Statiniai kintamieji, konstantos. Pasiekiamumo kontrolė • Inkapsuliacijos samprata • Paketai. Importo sakiniai • Matomumo modifikatoriai

- Paveldėjimo samprata
- Metodų užklotis (angl. overriding)
- Nuorodos super ir this
- java.lang.Object metodai toString(), equals() ir hashCode()
- Polimorfizmo samprata
- Abstrakčiosios klasės ir metodai
- Sąsajos (angl. interface)
- Funkcinės sąsajos (angl. functional interfaces)
- Tipo keitimas (angl. casting) ir instanceof operatorius
- Gija (angl. thread)
- Virtuali gija (angl. virtual thread)
- Programavimo principai: SOLID, Separation of Concerns
- UML diagramos, pagrindiniai elementai
- **Tema. Žurnalizavimas (angl. logging)**
- Žurnalizavimo lygiai: klaidos (angl. Error), informacijos (angl. Info), derinimo (angl. Debug), įspėjimo (angl. Warning)
- Gerosios žurnalizavimo praktikos
- Populiarūs Java žurnalizavimo įrankiai (pvz.: Log4j)
- **Tema. Dizaino šablonai**
- Dizaino šablonų nauda kuriant programinį kodą
- Kūrybiniai (angl. Creational) šablonai: Singleton, Builder, Abstract factory, Factory method
- Struktūriniai (angl. Structural) šablonai: Adapter
- Elgesio (angl. Behavioral) šablonai: Observer
- **Tema. Algoritmai**
- Rikiavimo algoritmai (bubble, selection, insertion, merge, quicksort)
- Algoritmų našumas ir kompleksiskumas (angl. Big-O notation)
- **Tema. Duomenų struktūros**
- Kolekcijų sąsajos (angl. Collection interfaces)
- Sąrašo (angl. List) sąsaja ir pagrindinės ją įgyvendinančios klasės
- Aibės (angl. Set) sąsaja ir pagrindinės ją įgyvendinančios klasės
- Map sąsaja ir pagrindinės ją įgyvendinančios klasės
- Eilė (angl. Queue)
- Iterable ir Iterator skirtumai ir panaudojimas

		<ul style="list-style-type: none"> • Duomenų srautai (angl. data streams), metodai map ir reduce • Comparable ir Comparator skirtumai, ir jų panaudojimas • Specialus duomenų tipas enum • Įrašas (angl. record) <p>Tema. Parametrizuoti tipai (angl. Generics)</p> <ul style="list-style-type: none"> • Bendrųjų metodų kūrimas • Bendrųjų klasių kūrimas <p>Tema. Įterptinės klasės (angl. Nested Classes)</p> <ul style="list-style-type: none"> • Vidinės klasės (angl. Inner Classes) • Lokalios klasės (angl. Local Classes) • Anoniminės klasės (angl. Anonymous Classes) • Rodyklinė funkcija (angl. Lambdas) <p>Tema. Išimčių (angl. Exception) gaudymas ir metimas</p> <ul style="list-style-type: none"> • Exception bazinės klasės sąvoką ir ypatumai • Išimčių gaudymas (angl. Catch) • Išimčių metimas (angl. Throw)
	<p>1.5. Testuoti programinę įrangą, naudojant su Java programavimo kalba suderinamus testavimo įrankius ir metodus.</p>	<p>Tema. Įvadas į programų testavimą</p> <ul style="list-style-type: none"> • Test-driven development (TDD) filosofija • Testų tipai: modulių (angl. unit), integraciniai (angl. integration), rankinis (angl. manual) testavimas ir t. t. • Priklausomumo nuo abstrakcijų svarba naudojant modulių (angl. unit) testus ir testines realizacijas <p>Tema. Programų testavimo įrankiai</p> <ul style="list-style-type: none"> • Testavimo įrankių apžvalga ir palyginimas (JUnit ir pan.) • Netikrų duomenų (angl. mock) naudojimas vienetų (angl. unit) testuose • Dirbtinio intelekto įrankiai testų generavimui
	<p>1.6. Naudoti dirbtinio intelekto įrankius programuojant.</p>	<p>Tema. Dirbtinio intelekto (DI) (angl. Artificial intelligence (AI)) pagrindai</p> <ul style="list-style-type: none"> • Mašininio mokymo apžvalga: algoritmai, taikymai ir tendencijos • Kalbos modeliai (angl. large language model (LLM)) ir jų alternatyvos (pvz.: GPT, Gemini, Claude, Grok, Phi4 ir kt.) • Modelių ypatumai (modelio dydis, haliucinacijos, konteksto lango dydis) • Užklausų dizainas (angl. prompt engineering): užklausos dalys, esamo kodo pateikimas, instrukcijų metodai <p>Tema. Dirbtinio intelekto (DI) (angl. Artificial intelligence (AI)) taikymas</p>

		<ul style="list-style-type: none"> • Dirbtinio intelekto taikymas programinio kodo rašymui • Dirbtinio intelekto taikymas testų generavimui • Dirbtinio intelekto taikymas dokumentuojant kodą
2. Naudoti Spring karkasą Java platformai.	2.1. Paašškinti programavimo karkasų privalumus ir trūkumus.	<p>Tema. <i>Karkasas</i></p> <ul style="list-style-type: none"> • Apibrėžimas ir paskirtis • Palyginimas su bibliotekomis • Architektūriniai šablonai (pvz.: MVC, MVVM) <p>Tema. <i>Karkasų apžvalga</i></p> <ul style="list-style-type: none"> • Populiariausi karkasai skirti Java programavimo kalbai • Kriterijai, pagal kuriuos renkamės karkasą
	2.2. Konfigūruoti <i>Spring</i> priklausomybių valdymo karkasą įvairiais būdais.	<p>Tema. <i>Objektų tipai</i></p> <ul style="list-style-type: none"> • Paprastasis Java objektas (angl. plain old Java object (POJO)) • Java Bean objektas • Spring Bean objektas • POJO, Java Bean ir Spring Bean palyginimas <p>Tema. <i>Spring karkasas</i></p> <ul style="list-style-type: none"> • Spring panaudojimas ir privalumai • SpringBoot projektas • Spring ryšių valdymo karkaso naudojimo tikslai • Spring ryšių valdymo karkaso moduliai ir jų funkcijos • Spring ryšių valdymo karkaso panaudojimo galimybės • Spring migracija, jos paskirtis ir iššūkiai <p>Tema. <i>MVC (angl. Model–view–controller) architektūra</i></p> <ul style="list-style-type: none"> • MVC pagrindai: koncepcijos ir principai • MVC įgyvendinimas įvairiose technologijose • WEB MVC architektūros modelis ir jo taikymas kuriant programinę įrangą <p>Tema. <i>Komunikacija tarp aplikacijų</i></p> <ul style="list-style-type: none"> • API (angl. Application Programming Interface) apibrėžimas ir paskirtis • SOAP (angl. Simple Object Access Protocol) apibrėžimas ir paskirtis • REST (angl. representational state transfer) apibrėžimas ir paskirtis • API dokumentacijos kūrimas (pvz. Swagger, OpenAPI)
	2.3. Naudoti <i>Spring Bean</i> Java programose įvairiais būdais.	<p>Tema. <i>Spring karkaso konteinerio elementas</i></p> <ul style="list-style-type: none"> • Spring Bean koncepcija, palyginimas su paprastais Java objektais • Spring konteinerio vaidmuo

		<p>Tema. <i>Spring bean</i></p> <ul style="list-style-type: none"> • Spring Bean aprašas • Spring Bean galiojimo sritis • Spring Bean gyvavimo ciklai • Spring Bean baigiamoji doroklė • Spring Bean konfigūravimas (priklausomybių, kolekcijų injekcijos, anotacijos) • Spring Bean susiejimas pagal tipą, vardą • Automatinio susiejimo naudojimas (angl. Autowiring)
3. Naudoti Java Persistence API duomenų valdymui Java programose.	3.1. Paaiškinti objektų ir reliacinės duomenų bazės komunikavimą, taikant įrankius.	<p>Tema. <i>Objektų santykių kūrimas (angl. ORM)</i></p> <ul style="list-style-type: none"> • Apibrėžimas ir paskirtis • ORM privalumai ir trūkumai • ORM įrankiai <p>Tema. <i>Java Persistence API</i></p> <ul style="list-style-type: none"> • Apibrėžimas ir paskirtis • Populiariausios Java Persistence API alternatyvos <p>Tema. <i>Hibernate</i></p> <ul style="list-style-type: none"> • Apibrėžimas ir paskirtis • Java Persistence API ir Hibernate palyginimas
	3.2. Susieti objektus ir reliacines duomenų bazes naudojant Java Persistence API.	<p>Tema. <i>Reliacinių duomenų manipuliavimas Java programiniame kode</i></p> <ul style="list-style-type: none"> • Objektų ir reliacinių duomenų bazių susiejimo principai • JPA (Java Persistence API) projekto konfigūravimas • Esybių klasių naudojimas • Esybių gyvavimo ciklas <p>Tema. <i>JPA naudojimas taikomiosiose programose</i></p> <ul style="list-style-type: none"> • Sąryšių tarp esybių valdymas • JPA aplikacijos užklausų kūrimas • CRUD operacijų vykdymas
	3.3. Valdyti duomenis, naudojant Hibernate karkasą taikomiosiose Java programose.	<p>Tema. <i>Java programinio kodo sujungimas su duomenų baze</i></p> <ul style="list-style-type: none"> • Komunikacija su duomenų baze, naudojant JDBC (angl. Java Database Connectivity) • Komunikacija su duomenų baze, naudojant Hibernate karkasą • JDBC ir Hibernate palyginimas, privalumai ir trūkumai <p>Tema. <i>Hibernate karkaso pagrindai</i></p> <ul style="list-style-type: none"> • Hibernate įrankiai ir aplinkos sąranka • Hibernate karkaso konfigūracija

	<ul style="list-style-type: none"> • Hibernate Persistence • Hibernate ORM • Hibernate užklausų kalba (Hibernate Query Language - HQL)
Mokymosi pasiekimų vertinimo kriterijai	Išsamiai papasakota apie Java programavimo kalbos atsiradimo istoriją ir pagrindines taikymo sritis. Pademonstruotas Java programavimo kalbos sintaksės ir įrankių valdymas. Sukurtas nesudėtingas programinis kodas Java programavimo kalba. Pademonstruoti objektinio programavimo principai programuojant Java programavimo kalba. Ištestuota programinė įranga, parašyta Java programavimo kalba, naudojant suderintus testavimo įrankius ir metodus. Pademonstruotas dirbtinio intelekto įrankių naudojimas programuojant. Paaiškinti programavimo karkasų privalumai ir trūkumai. Įvairiais būdais sukonfigūruotas Spring priklausomybių valdymo karkasas. Įvairiais būdais programose naudojamas Spring Bean. Išsamiai papasakota apie objektų ir reliacinių duomenų bazių komunikavimą, taikant įrankius. Susieti objektai ir reliacinės duomenų bazės, naudojant Java Persistence API. Pademonstruotas duomenų valdymas, naudojant Hibernate karkasą. Atliekant darbus laikomasi ergonomikos, saugaus darbo, priešgaisrinės saugos, darbo higienos ir aplinkosaugos reikalavimų, užtikrinančių IT specialistų sveikatą, produktyvumą ir saugią darbo aplinką.
Reikalavimai mokymui skirtiems metodiniams ir materialiesiems ištekliams	<p><i>Mokymo(si) medžiaga:</i></p> <ul style="list-style-type: none"> • Vadovėliai ir kita mokomoji medžiaga • Testas turimiems gebėjimams vertinti <p><i>Mokymo(si) priemonės:</i></p> <ul style="list-style-type: none"> • Techninės priemonės mokymo(si) medžiagai iliustruoti, vizualizuoti, pristatyti
Reikalavimai teorinio ir praktinio mokymo vietai	Klasė ar kita mokymuisi pritaikyta patalpa su techninėmis priemonėmis mokymo(si) medžiagai pateikti (kompiuteriu ir prieiga prie interneto, vaizdo projektoriumi) ir kompiuteriais, skirtais mokinių darbui. Praktinio mokymo klasė (patalpa), aprūpinta kompiuteriais, prieiga prie interneto, programine įranga.
Reikalavimai mokytojų dalykiniam pasirengimui (dalykinei kvalifikacijai)	Modulį gali vesti mokytojas, turintis: 1) Lietuvos Respublikos švietimo įstatyme ir Reikalavimų mokytojų kvalifikacijai apraše, patvirtintame Lietuvos Respublikos švietimo, mokslo ir sporto ministro 2014 m. rugpjūčio 29 d. įsakymu Nr. V-774 „Dėl Reikalavimų mokytojų kvalifikacijai aprašo patvirtinimo“, nustatytą išsilavinimą ir kvalifikaciją; 2) Informatikos mokslų studijų krypties grupės ar lygiavertį išsilavinimą arba vidurinį išsilavinimą ir jaunesniojo programuotojo ar lygiavertę kvalifikaciją, ne mažesnę kaip 3 metų jaunesniojo programuotojo profesinės veiklos patirtį ir pedagoginių ir psichologinių žinių kurso baigimo pažymėjimą.

Modulio pavadinimas – „Informacinių sistemų projektavimas ir kūrimas JavaScript programavimo kalba“

Valstybinis kodas	406131346
Modulio LTKS lygis	IV
Apimtis mokymosi kreditais	20
Asmens pasirengimo	<i>netaikoma</i>

mokyti modulyje reikalavimai		
Kompetencijos	Mokymosi rezultatai	Rekomenduojamas turinys mokymosi rezultatams pasiekti
1. Kurti tipinę programinę įrangą JavaScript programavimo kalba.	1.1. Paaiškinti objektnio programavimo principus JavaScript programavimo kalboje.	<p>Tema. <i>Objektnio programavimo principai</i></p> <ul style="list-style-type: none"> • Funkcinio, procedūrinio ir objektnio programavimo principai • Objektnio programavimo samprata ir abstrakcijos principas <p>Tema. <i>Objektnio programavimo taikymas</i></p> <ul style="list-style-type: none"> • Abstrakcijos taikymas objektniame programavime • Paveldėjimo ir polimorfizmo taikymo pavyzdžiai
	1.2. Programuoti JavaScript programavimo kalba, taikant objektnio programavimo principus.	<p>Tema. <i>Objektnio programavimo paradigma</i></p> <ul style="list-style-type: none"> • Klasės ir objekto sąvokos • Objekto kintamieji ir objekto metodai • Objektų kūrimas ir panaudojimas. Konstruktoriai. Objektų palyginimas • Inkapsuliacijos samprata • Paveldėjimo samprata • Polimorfizmo samprata • Kodo struktūravimas ir organizavimas • Prototipinis paveldėjimas • Prototipų grandinė • Asinchroninis programavimas su objektais <p>Tema. <i>Duomenų struktūros</i></p> <ul style="list-style-type: none"> • Rinkiniai (angl. Sets) • Žemėlapiai (angl. Maps) <p>Tema. <i>Žurnalizavimas (angl. logging)</i></p> <ul style="list-style-type: none"> • Žurnalizavimo lygiai: klaidos (angl. Error), informacijos (angl. Info), derinimo (angl. Debug), įspėjimo (angl. Warning) • Gerosios žurnalizavimo praktikos • JavaScript žurnalizavimo įrankiai <p>Tema. <i>Dizaino šablonai</i></p> <ul style="list-style-type: none"> • Dizaino šablonų nauda kuriant programinę kodą • Kūrybiniai (angl. Creational) šablonai: Singleton, Builder, Abstract factory, Factory method • Struktūriniai (angl. Structural) šablonai: Adapter • Elgesio (angl. Behavioral) šablonai: Observer

		<p>Tema. Algoritmai</p> <ul style="list-style-type: none"> • Rikiavimo algoritmai (bubble, selection, insertion, merge, quicksort) • Algoritmų našumas ir kompleksiskumas (angl. Big-O notation) <p>Tema. Gerosios praktikos</p> <ul style="list-style-type: none"> • Keep it simple, stupid (KISS) principas • SOLID projektavimo principai ir pritaikymas <p>Tema: Išimčių (Exceptions) gaudymas ir metimas JavaScript</p> <ul style="list-style-type: none"> • Error klasės sąvoka ir ypatumai • Išimčių metimas naudojant throw raktinį žodį • Naujos klaidos (Exception) klasės kūrimas paveldint Error klasę • try/catch/finally blokų naudojimas • try, catch ir finally blokų veikimas • catch bloko veikimas ir paskirtis • Skirtingų klaidų tipų gaudymas ir apdorojimas (Error, TypeError, SyntaxError ir kt.) • Išimčių filtravimas catch bloke naudojant sąlyginius patikrinimus <p>Tema: Kodo transformavimo ir pakavimo įrankių naudojimas JavaScript</p> <ul style="list-style-type: none"> • Modulinių sistemų naudojimas JavaScript: ES Modules (ESM), CommonJS, AMD, UMD • Babel – modernios JavaScript sintaksės transpiliacija į senesnes naršyklių versijas • Web bundler'iai: Webpack, Parcel, Vite, Rollup – jų paskirtis ir naudojimas • Minifikavimas ir optimizavimas (pvz. Terser, UglifyJS) • Nebereikalingo kodo pašalinimas (angl. Tree Shaking) naudojant ES modulius (ES Modules) ir Webpack • Automatizuotas priklausomybių valdymas naudojant npm ir pnpm
	<p>1.3. Testuoti programinę įrangą, naudojant su JavaScript programavimo kalba suderinamus testavimo įrankius ir metodus.</p>	<p>Tema. Įvadas į programų testavimą</p> <ul style="list-style-type: none"> • Test-driven development (TDD) filosofija • Testų tipai: modulių (angl. unit), integraciniai (angl. integration), rankiniai (angl. manual) testai ir t. t. • Priklausymo nuo abstrakcijų svarba naudojant modulių (angl. unit) testus ir testines realizacijas • Programinio kodo padengimas testais <p>Tema. Programų testavimo įrankiai</p> <ul style="list-style-type: none"> • Testavimo įrankių apžvalga ir palyginimas (Jest) • Netikrų duomenų (angl. mock) naudojimas vienetu (angl. unit) testuose, naudojant įrankius

	1.4. Naudoti dirbtinio intelekto įrankius programuojant.	<p>Tema. <i>Dirbtinio intelekto (DI) (angl. Artificial intelligence (AI)) pagrindai</i></p> <ul style="list-style-type: none"> • Mašininio mokymo apžvalga: algoritmai, taikymai ir tendencijos • Kalbos modeliai (angl. large language model (LLM)) ir jų alternatyvos (pvz.: GPT, Gemini, Claude, Grok, Phi4 ir kt.) • Modelių ypatumai (modelio dydis, haliucinacijos, konteksto lango dydis) • Užklausų dizainas (angl. prompt engineering): užklausos dalys, esamo kodo pateikimas, instrukcijų metodai <p>Tema. <i>Dirbtinio intelekto (DI) (angl. Artificial intelligence (AI)) taikymas</i></p> <ul style="list-style-type: none"> • Dirbtinio intelekto taikymas programinio kodo rašymui • Dirbtinio intelekto taikymas testų generavimui • Dirbtinio intelekto taikymas dokumentuojant kodą
2. Kurti JavaScript taikomąsias programas, naudojant karkasus.	2.1. Paaiškinti programavimo karkasų privalumus ir trūkumus.	<p>Tema. <i>Karkasas</i></p> <ul style="list-style-type: none"> • Apibrėžimas ir paskirtis • Palyginimas su bibliotekomis • Architektūriniai šablonai (pvz.: MVC, MVVM) <p>Tema. <i>Karkasų apžvalga</i></p> <ul style="list-style-type: none"> • Populiariausi karkasai skirti JavaScript programavimo kalbai • Kriterijai, pagal kuriuos renkamės karkasą
	2.2. Kurti taikomąsias programas, naudojant JavaScript programavimo kalbos karkasus.	<p>Tema. <i>Node.js pagrindai</i></p> <ul style="list-style-type: none"> • Node.js paskirtis ir pagrindinės savybės • Node.js architektūra (V8 variklis) • Asinchroninis I/O (neblokuojančios operacijos ir callback funkcijos) • Node.js diegimas • Node Package Manager (NPM) paskirtis ir diegimas • NPM paketo valdymas: ieškoti, diegti ir atnaujinti paketus (npm install, npm update, npm uninstall) • package.json failo struktūra ir reikšmė projektui • Projekto priklausomybių valdymas ir jų versijos <p>Tema. <i>Node.js modulių sistema</i></p> <ul style="list-style-type: none"> • Pagrindiniai Node.js globalūs objektai (global, process, console, module, require(), __dirname, __filename) • Modulio sukūrimas ir taikymas • Modulių kapsuliavimo funkcija (angl. Module wrapper function) • Node.js path modulis darbui su failų ir katalogų keliais. Pagrindinės modulio funkcijos:

		<p>path.join([...paths]), path.resolve([...paths]), path.basename(path, [ext]), path.dirname(path) ir kt.</p> <ul style="list-style-type: none"> • Node.js OS modulis informacijos gavimui apie operacinę sistemą • Node.js File System (fs) modulis darbui su failų sistema. Pagrindinės funkcijos: skaitymas, rašymas, pridėjimas, šalinimas • Node.js HTTP modulis skirtas kurti HTTP serverius ir valdyti užklausas bei atsakymus • HTTP modulis: serverio kūrimas, užklausų apdorojimas, HTTP antraščių nustatymas <p>Tema. <i>Node Package Manager (NPM)</i></p> <ul style="list-style-type: none"> • Paketų (angl. package) paskirtis, diegimas ir naudojimas • Standartinė versijų valdymo sistema (angl. Semantic Versioning (SemVer)) • Pagrindinės SemVer taisyklės NPM: MAJOR, MINOR, PATCH • Priklausomybių valdymas: Patch releases, Minor releases, Major releases • Paketo informacijos peržiūra (angl. Viewing Package Information). Pagrindinės komandos: npm view <paketo-pavadinimas>, npm view <paketo-pavadinimas>@<versija> dependencies, npm view <paketo-pavadinimas> repository.url • Specifinės paketo versijos diegimas, patikrinimas, informacija apie naują versiją: npm install <paketo-pavadinimas>@<versijos-numeris>, npm list <paketo-pavadinimas>, npm view <paketo-pavadinimas> version • Lokalus paketų atnaujinimas: npm outdated, npm update, npm update <paketo-pavadinimas>, npm install <paketo-pavadinimas>@latest • DevDependencies paketo tipas, paskirtis, diegimas • Paketų šalinimas • Globalių paketų diegimas ir peržiūra, privalumai ir trūkumai • Paketų viešinimas. NPM paskyros sukūrimas. Pagrindiniai viešinimo žingsniai <p>Tema. <i>Darbas su JavaScript karkasais: pagrindiniai įrankiai ir metodai</i></p> <ul style="list-style-type: none"> • Maršrutizavimas (angl. Routing) • Dinaminiai maršrutai ir apsaugoti maršrutai • Duomenų susiejimas (angl. Data Binding) • Vienpusis ir dvipusis duomenų susiejimas • Formų kūrimas ir tvarkymas • Kliento pusės ir serverio pusės duomenų validacija • Priklausomybių injekcija ir komponentų apimtis
	2.3. Kurti taikomasias žiniatinklio programas su grafine sąsaja,	<p>Tema. <i>API kūrimas</i></p> <ul style="list-style-type: none"> • Pagrindiniai RESTful API principai: kliento-serverio, Stateless

	<p>naudojant JavaScript programavimo kalbos karkasus.</p>	<ul style="list-style-type: none"> • RESTful API veikimo principai • Express.js web aplikacijų kūrimo karkasas, diegimas • Request-Response ciklas • Paprasčiausio web serverio sukūrimas ir paleidimas • Nodemon įrankis, jo privalumai, diegimas, konfigūravimas • Aplinkos kintamieji (angl. environment variables), naudojimo būdai (.env failas) ir prieiga (paketas dotenv) • Dinamiški URL segmentai Route parameters, req.params objektas • HTTP užklausos: GET, POST, PUT, DELETE • Įrankiai, skirti REST API interfeisų kūrimui, testavimui, dokumentavimui ir vizualizavimui (pvz.: Postman, Swagger) • Middleware – jų paskirtis ir naudojimo pavyzdžiai • Middleware moduliai (pvz.: body-parser, cors, express-validator) • Router naudojimas • Autentifikavimas ir autorizavimas (angl. Authentication and Authorization) • JSON Web Token (JWT) standartas • API raktų generavimas ir naudojimas • API dokumentacijos kūrimas, gerosios praktikos • API versijų valdymas <p>Tema. Kodavimo standartai</p> <ul style="list-style-type: none"> • Aiški ir nuosekli kodo struktūra (pvz. MVC, Clean Architecture, Hexagonal Architecture) • Kintamųjų, funkcijų ir klasės pavadinimų standartai • Asinchroninio kodo tvarkymas (async/await, Promises) • Automatinės kodo analizės įrankiai (pvz. SonarQube, Codacy) • Statinio kodo analizės įrankiai Node.js projektuose (pvz. ESLint) • Kodo formatavimo įrankiai Node.js projektuose (pvz. Prettier) • API užklausų (angl. API endpoint) pavadinimų ir struktūros geriausios praktikos • HTTP statuso kodų naudojimas (200 OK, 400 Bad Request, 500 Internal Server Error) <p>Tema. Darbas su duomenimis</p> <ul style="list-style-type: none"> • CRUD operacijos. Create (sukurti), Read (skaityti), Update (atnaujinti) ir Delete (ištrinti) • Node.js duomenų bazių palaikymas (Reliacinė ir nereliacinė duomenų bazės) • Reliacinių duomenų bazių integracija (pvz. PostgreSQL): diegimas, konfigūravimas • Pagrindinės užklausos duomenis gauti, įterpti, atnaujinti, ištrinti, filtruoti, rūšiuoti, grupuoti, funkcijos in ir between
--	---	---

		<ul style="list-style-type: none"> • Nereliacinių duomenų bazių integracija (pvz. MongoDB): diegimas, konfigūravimas • Nereliacinių duomenų bazių modeliai ir schemas • Dokumentų užklausos, naudojant find() metodą. Užklausos operatoriai, AND ir OR sąlygų naudojimas
Mokymosi pasiekimų vertinimo kriterijai		<p>Paaiškinti objektinio programavimo principai JavaScript programavimo kalboje. Pademonstruoti objektinio programavimo principai programuojant JavaScript programavimo kalba. Ištestuota programinė įranga, parašyta JavaScript programavimo kalba, naudojant suderintus testavimo įrankius ir metodus. Pademonstruotas dirbtinio intelekto įrankių naudojimas programuojant. Paaiškinti programavimo karkasų privalumai ir trūkumai. Sukurtos taikomosios programos, naudojant JavaScript programavimo kalbos karkasus. Sukurtos taikomosios žiniatinklio programos su grafine sąsaja, naudojant JavaScript programavimo kalbos karkasus. Atliekant darbus laikomasi ergonomikos, saugaus darbo, priešgaisrinės saugos, darbo higienos ir aplinkosaugos reikalavimų, užtikrinančių IT specialistų sveikatą, produktyvumą ir saugią darbo aplinką.</p>
Reikalavimai mokymui skirtiems metodiniams ir materialiesiems ištekliams		<p><i>Mokymo(si) medžiaga:</i></p> <ul style="list-style-type: none"> • Vadovėliai ir kita mokomoji medžiaga • Testas turimiems gebėjimams vertinti <p><i>Mokymo(si) priemonės:</i></p> <ul style="list-style-type: none"> • Techninės priemonės mokymo(si) medžiagai iliustruoti, vizualizuoti, pristatyti
Reikalavimai teorinio ir praktinio mokymo vietai		<p>Klasė ar kita mokymuisi pritaikyta patalpa su techninėmis priemonėmis mokymo(si) medžiagai pateikti (kompiuteriu ir prieiga prie interneto, vaizdo projektoriumi) ir kompiuteriais, skirtais mokinių darbui.</p> <p>Praktinio mokymo klasė (patalpa), aprūpinta kompiuteriais, prieiga prie interneto, programine įranga.</p>
Reikalavimai mokytojų dalykiniam pasirengimui (dalykinei kvalifikacijai)		<p>Modulį gali vesti mokytojas, turintis:</p> <ol style="list-style-type: none"> 1) Lietuvos Respublikos švietimo įstatyme ir Reikalavimų mokytojų kvalifikacijai apraše, patvirtintame Lietuvos Respublikos švietimo, mokslo ir sporto ministro 2014 m. rugpjūčio 29 d. įsakymu Nr. V-774 „Dėl Reikalavimų mokytojų kvalifikacijai aprašo patvirtinimo“, nustatytą išsilavinimą ir kvalifikaciją; 2) Informatikos mokslų studijų krypties grupės ar lygiavertį išsilavinimą arba vidurinį išsilavinimą ir jaunesniojo programuotojo ar lygiavertę kvalifikaciją, ne mažesnę kaip 3 metų jaunesniojo programuotojo profesinės veiklos patirtį ir pedagoginių ir psichologinių žinių kurso baigimo pažymėjimą.

Modulio pavadinimas – „Informacinių sistemų projektavimas ir kūrimas C# programavimo kalba“

Valstybinis kodas	406131347
Modulio LTKS lygis	IV
Apimtis mokymosi kreditais	20
Asmens pasirengimo mokyti modulyje reikalavimai	<i>netaikoma</i>

Kompetencijos	Mokymosi rezultatai	Rekomenduojamas turinys mokymosi rezultatams pasiekti
1. Kurti tipinę programinę įrangą C# programavimo kalba.	1.1. Apžvelgti C# programavimo kalbos atsiradimo istoriją ir pagrindines taikymo sritis.	<p>Tema. Įvadas į C# programavimo kalbą</p> <ul style="list-style-type: none"> • Programavimo kalbos istorija • C# versijavimas ir licencija • C# kalbos konceptas <p>Tema. Programavimo kalbos taikymo sritys</p> <ul style="list-style-type: none"> • Windows darbalaukio programos • Web programos • Mobiliosios programos
	1.2. Valdyti C# programavimo kalbos įrankius ir sintaksę.	<p>Tema. C# .NET aplinka</p> <ul style="list-style-type: none"> • .NET istorija ir evoliucija • .NET įvairių platformų (angl. cross-platform) palaikymas • .NET SDK diegimas • Grafinė programavimo aplinkos sąsaja (angl. Integrated Development Environment (IDE)) diegimas ir konfigūravimas. Pvz.: Visual Studio, Rider • Common Language Runtime (CLR) sąvoka ir reikšmė programos paleidime • just-in-time (JIT) compiler sąvoka ir reikšmė programos kompiliavime • Common Intermediate Language (CIL) sąvoka ir reikšmė programos kompiliavime <p>Tema. C# projekto kūrimas ir struktūra</p> <ul style="list-style-type: none"> • C# identifikatorių pavadinimų taisyklės ir susitarimai • Konsolinės aplikacijos (angl. Console application) projekto kūrimas • Programos paleidimas, naudojant .NET CLI komandą dotnet • Projektų rinkinio (angl. solution) struktūros apžvalga (projektų rinkinys (angl. solution), projektas (angl. project), projekto rinkinio failas (angl. solution file)) • Projekto struktūros apžvalga (projekto failas (angl. project file), vardų rinkinys (angl. namespace), klasė (angl. class)) • Programos pradžios taškas (Main metodas) <p>Tema. C# programavimo kalbos pagrindai ir sintaksė</p> <ul style="list-style-type: none"> • Blokinė schema • Integruoti duomenų tipai (angl. Built-in types). Pvz.: skaitiniai, loginis, tekstiniai • Aritmetinės operacijos (+, -, /, % ir t. t.) • Kintamojo deklaravimas ir inicijavimas. var raktinis žodis • Sąlyginis sakiny (if, if-else) • Palyginimo operatoriai (==, !=, <, >, <=, >=)

		<ul style="list-style-type: none"> • Loginiai operatoriai (&&, , !) • Trinaris sąlyginis sakiny (?:) • switch konstrukcija • Ciklai (for, while, do) • break ir continue raktinių žodžių paskirtis ir panaudojimas cikluose • Masyvai, daugiamačiai masyvai
	<p>1.3. Kurti nesudėtingą programinį kodą C# programavimo kalba.</p>	<p>Tema. Metodai</p> <ul style="list-style-type: none"> • Statinių metodų aprašymas • Kintamųjų perdavimas į metodą • Reikšmės gražinimas iš metodo • Rekursija (angl. recursion) • Algoritmai: minimalios reikšmės paieška, maksimalios reikšmės paieška, kiekio (angl. count) skaičiavimo algoritmas, paieškos algoritmai • Nekartokite savęs (angl. Do not Repeat Yourself (DRY)) principas • Kintamųjų galiojimo sritis (angl. scope) <p>Tema. Tekstinė eilutė</p> <ul style="list-style-type: none"> • Tekstinės eilutės (angl. String) apibrėžimas ir skirtingi sukūrimo būdai • Tekstinės eilutės savybė Length • Tekstinės eilutės paieškos metodai: IndexOf(), LastIndexOf(), StartsWith(), EndsWith(), Contains() • Tekstinės eilutės modifikavimo metodai: Substring(), Replace(), Trim() • Tekstinės eilutės suliejimo metodai: Concat() ir kt. <p>Tema. Specialūs simboliai</p> <ul style="list-style-type: none"> • Specialių simbolių įterpimas (angl. escape character), naudojant pasvirąjį kairinį brūkšnį (angl. backslash) • Naujos eilutės simboliai: \r, \n, \r\n. Environment.NewLine laukas • \$ (angl. string interpolation) ir @ (angl. verbatim text) specialių simbolių panaudojimas kartu su tekstinėmis eilutėmis
	<p>1.4. Programuoti C# programavimo kalba, taikant objektinio programavimo principus.</p>	<p>Tema. Įvadas į objektinį programavimą</p> <ul style="list-style-type: none"> • Tipų sistema (angl. Common type system) • Klasės sąvoka ir sintaksė • Objekto sąvoka. Objekto kūrimas, naudojant raktinį žodį new • Tipų pagal nuorodą (angl. reference type) ir tipų pagal reikšmę (angl. value type) ypatumai ir skirtumai

		<ul style="list-style-type: none"> • Klasės nariai (savybės (properties), laukai (fields), konstantos (constants)) • Metodai. Metodų deklavimas (angl. method signature) • Metodų parametrų ir argumentų sąvokos ir skirtumai • Parametrų perdavimas į metodą pagal reikšmę (angl. value type) ar pagal nuorodą (angl. reference type) • ref, out, in, ref readonly raktiniai žodžiai • Metodų perkrovimas (angl. overloading) • Operatorių perkrovimas (angl. operator overloading) • Lambda išraiškos • Klasės konstruktoriai • Klasės destruktoriai (angl. finalizers) • Prieigos modifikatoriai (angl. access modifiers). Pvz.: private, public, protected, internal ir t. t. <p>Tema. Objektinio programavimo paradigma (angl. Object-Oriented Programming (OOP))</p> <ul style="list-style-type: none"> • Inkapsuliacijos samprata (angl. Encapsulation) • Paveldėjimo samprata (angl. Inheritance) • Metodų perrašymas (angl. overriding). virtual/override raktiniai žodžiai • Klasės narių slėpimas, naudojant new raktinį žodį • Lokalios funkcijos (angl. local functions) • Polimorfizmo samprata (angl. Polymorphism) • Įrašo tipas (angl. record) sąvoka ir ypatumai • Abstrakčios klasės ir klasės nariai. Jų ypatumai ir veikimas • Interfeisai • IDisposable interfeiso sąvoka ir įgyvendinimas • using raktinio žodžio sąvoka ir panaudojimas • foreach ciklas, IEnumerable interfeiso įgyvendinimas ir panaudojimas • Kiti tipai (angl. Custom types). Pvz.: struct, enum • Statinės klasės • Extension metodai • Events sąvoka ir naudojimas • Delegate sąvoka ir naudojimas <p>Tema. Gerosios praktikos</p> <ul style="list-style-type: none"> • You Ain't Gonna Need It (YAGNI) konceptas • Keep it simple, stupid (KISS) principas
--	--	---

		<ul style="list-style-type: none"> • SOLID projektavimo principai ir pritaikymas C# aplinkoje <p>Tema. Išimčių (angl. Exception) gaudymas ir metimas</p> <ul style="list-style-type: none"> • Exception bazinės klasės sąvoką ir ypatumus • Išimčių metimas panaudojant throw raktinį žodį • Naujos išimties klasės aprašymas paveldint Exception klasę • try/catch/finally blokas • try, catch ir finally dalies veikimas • catch dalies veikimas ir paskirtis • Skirtingų tipų išimčių gaudymas ir apdorojimas • Išimčių filtravimas catch dalyje <p>Tema. Parametrizuoti tipai (angl. Generics)</p> <ul style="list-style-type: none"> • Bendrųjų metodų kūrimas • Bendrųjų klasių kūrimas • Parametrizuotų tipų apribojimas su where raktiniu žodžiu <p>Tema. Kolekcijos, duomenų struktūros ir algoritmai</p> <ul style="list-style-type: none"> • System.Collections vardų rinkinio reikšmė kolekcijoms ir duomenų struktūroms • Kolekcijos (Array, List, LinkedList ir t. t.) • IEnumerable interfeiso sąvoka ir svarba kolekcijose • Extension metodai (Select, Where, GroupBy, Max, Min ir t. t.) iš System.Linq vardų rinkinio darbui su IEnumerable • Žodyno tipo kolekcijos (Dictionary, HashSet) • Duomenų struktūros (Stack, Queue) <p>Tema. Darbas su duomenimis</p> <ul style="list-style-type: none"> • Duomenų skaitymas/rašymas į failą • Stream bazinės klasės sąvoką ir panaudojimas skaitant/rašant duomenis į failą (FileStream), atmintį (MemoryStream) ir t. t. • Object-Relational Mapping (ORM) sąvoka ir palyginimas (Entity Framework Core, Dapper ir t. t.) • Duomenų traukimas, rašymas ir modifikavimas duomenų bazėje, naudojant ORM (CRUD)
	<p>1.5. Testuoti programinę įrangą, naudojant su C# programavimo kalba suderintus testavimo įrankius ir metodus.</p>	<p>Tema. Įvadas į programų testavimą</p> <ul style="list-style-type: none"> • Test-driven development (TDD) filosofija • Testų tipai. Pvz.: modulių (angl. unit), integraciniai (angl. integration), rankinis (angl. manual) testai ir t. t.

		<ul style="list-style-type: none"> • Priklausymo nuo abstrakcijų svarba naudojant modulių (angl. unit) testus ir testines realizacijas <p>Tema. Programų testavimo įrankiai</p> <ul style="list-style-type: none"> • Testavimo įrankių apžvalga ir palyginimas (MSTest, xUnit, NUnit ir t. t.) • Netikrų duomenų (angl. mock) naudojimas modulių (angl. unit) testuose, naudojant įrankius (Moq, NSubstitute ir t. t.)
	1.6. Naudoti dirbtinio intelekto įrankius programuojant.	<p>Tema. Dirbtinio intelekto (DI) (angl. Artificial intelligence (AI)) pagrindai</p> <ul style="list-style-type: none"> • Mašininio mokymo apžvalga: algoritmai, taikymai ir tendencijos • Kalbos modeliai (angl. large language model (LLM)) ir jų alternatyvos (pvz.: GPT, Gemini, Claude, Grok, Phi4 ir kt.) • Modelių ypatumai (modelio dydis, haliucinacijos, konteksto lango dydis) • Užklausų dizainas (angl. prompt engineering): užklausos dalys, esamo kodo pateikimas, instrukcijų metodai <p>Tema. Dirbtinio intelekto (DI) (angl. Artificial intelligence (AI)) taikymas</p> <ul style="list-style-type: none"> • Dirbtinio intelekto taikymas programinio kodo rašymui • Dirbtinio intelekto taikymas testų generavimui • Dirbtinio intelekto taikymas dokumentuojant kodą
2. Kurti C# taikomąsias programas, naudojant karkasus.	2.1. Paaiškinti programavimo karkasų privalumus ir trūkumus.	<p>Tema. Karkasas</p> <ul style="list-style-type: none"> • Apibrėžimas ir paskirtis • Palyginimas su bibliotekomis • Architektūriniai šablonai (pvz.: MVC, MVVM) <p>Tema. Karkasų apžvalga</p> <ul style="list-style-type: none"> • Populiariausi karkasai skirti C# programavimo kalbai (ASP.NET Core, Entity Framework Core, ASP.NET MVC) • Kriterijai, pagal kuriuos renkamas karkasą • NuGet paketų tvarkyklės panaudojimas
	2.2. Kurti taikomąsias žiniatinklio programas su grafine vartotojo sąsaja, naudojant .NET aplinką.	<p>Tema. Įvadas į API kūrimą</p> <ul style="list-style-type: none"> • REST API sąvoka • API projekto kūrimas ir konfigūravimas • Kontrolerių (angl. Controller) aprašymas • Kontrolerių veiksmų (angl. Action) aprašymas • Route atributas ir kelio nustatymas • HttpGet, HttpPost ir t. t. atributai • Dependency Injection (DI) sąvoka ir svarba

		<ul style="list-style-type: none"> • Gyvavimo ciklo (angl. lifetime) nustatymas servisams, naudojant Dependency Injection (DI) <p>Tema. REST API kūrimas .NET aplinkoje</p> <ul style="list-style-type: none"> • API užklausos rezultato gražinimas JSON formatu • API paleidimas, naudojant dotnet komandą ir paduodant parametrus • Autentifikavimas ir autorizavimas (angl. Authentication and Authorization)
	2.3. Programuoti taikomasias žiniatinklio programas su grafine vartotojo sąsaja, naudojant daugiagijinio ir asinchroninio programavimo principus.	<p>Tema. Daugiagijis (angl. Multithreaded) programavimas</p> <ul style="list-style-type: none"> • Gijos sąvoka ir reikšmė aplikacijai • Gijos sukūrimas, paleidimas ir laukimas, naudojant Thread klasę • Procesų sinchronizavimas (lock, monitor, semaphore ir t. t.) • Apsaugotų kolekcijų (angl. thread safe collections) naudojimas gijose. Pvz.: ConcurrentDictionary, ConcurrentStack, ConcurrentQueue ir t. t. <p>Tema. Asinchroninis programavimas</p> <ul style="list-style-type: none"> • Task parallel library (TPL) paskirtis • Veiksmo paleidimas, laukimas ir t. t., naudojant Task klasę • async/await naudojimas
Mokymosi pasiekimų vertinimo kriterijai	Išsamiai papasakota apie C# programavimo kalbos atsiradimo istoriją ir pagrindines taikymo sritis. Pademonstruotas C# programavimo kalbos sintaksės ir įrankių valdymas. Sukurtas nesudėtingas programinis kodas C# programavimo kalba. Pademonstruoti objektinio programavimo principai programuojant C# programavimo kalba. Ištestuota programinė įranga, parašyta C# programavimo kalba, naudojant suderintus testavimo įrankius ir metodus. Pademonstruotas dirbtinio intelekto įrankių naudojimas programuojant. Paaiškinti programavimo karkasų privalumai ir trūkumai. Sukurtos taikomosios žiniatinklio programos su grafine vartotojo sąsaja, naudojant .NET aplinką. Suprogramuotos taikomosios žiniatinklio programos su grafine vartotojo sąsaja, naudojant daugiagijinio ir asinchroninio programavimo principus. Atliekant darbus laikomasi ergonomikos, saugaus darbo, priešgaisrinės saugos, darbo higienos ir aplinkosaugos reikalavimų, užtikrinančių IT specialistų sveikatą, produktyvumą ir saugią darbo aplinką.	
Reikalavimai mokymui skirtiems metodiniams ir materialiesiems ištekliams	<p><i>Mokymo(si) medžiaga:</i></p> <ul style="list-style-type: none"> • Vadovėliai ir kita mokomoji medžiaga • Testas turimiems gebėjimams vertinti <p><i>Mokymo(si) priemonės:</i></p> <ul style="list-style-type: none"> • Techninės priemonės mokymo(si) medžiagai iliustruoti, vizualizuoti, pristatyti 	
Reikalavimai teorinio ir praktinio mokymo vietai	Klasė ar kita mokymuisi pritaikyta patalpa su techninėmis priemonėmis mokymo(si) medžiagai pateikti (kompiuteriu ir prieiga prie interneto, vaizdo projektoriumi) ir kompiuteriais, skirtais mokinių darbui. Praktinio mokymo klasė (patalpa), aprūpinta kompiuteriais, prieiga prie interneto, programine įranga.	
Reikalavimai mokytojų	Modulį gali vesti mokytojas, turintis:	

dalykiniam pasirengimui (dalykinei kvalifikacijai)	<p>1) Lietuvos Respublikos švietimo įstatyme ir Reikalavimų mokytojų kvalifikacijai apraše, patvirtintame Lietuvos Respublikos švietimo, mokslo ir sporto ministro 2014 m. rugpjūčio 29 d. įsakymu Nr. V-774 „Dėl Reikalavimų mokytojų kvalifikacijai aprašo patvirtinimo“, nustatytą išsilavinimą ir kvalifikaciją;</p> <p>2) Informatikos mokslų studijų krypties grupės ar lygiavertį išsilavinimą arba vidurinį išsilavinimą ir jaunesniojo programuotojo ar lygiavertę kvalifikaciją, ne mažesnę kaip 3 metų jaunesniojo programuotojo profesinės veiklos patirtį ir pedagoginių ir psichologinių žinių kurso baigimo pažymėjimą.</p>
--	--

Modulio pavadinimas – „Informacinių sistemų projektavimas ir kūrimas PHP programavimo kalba“

Valstybinis kodas	406131348	
Modulio LTKS lygis	IV	
Apimtis mokymosi kreditais	20	
Asmens pasirengimo mokytiis modulyje reikalavimai	<i>netaikoma</i>	
Kompetencijos	Mokymosi rezultatai	Rekomenduojamas turinys mokymosi rezultatams pasiekti
1. Kurti tipinę programinę įrangą PHP programavimo kalba.	1.1. Apžvelgti PHP programavimo kalbos atsiradimo istoriją ir pagrindines taikymo sritis.	<p>Tema. Įvadas į PHP programavimo kalbą</p> <ul style="list-style-type: none"> • Programavimo kalbos istorija • PHP versijavimas ir licencija • PHP kalbos konceptas <p>Tema. Programavimo kalbos taikymo sritys</p> <ul style="list-style-type: none"> • Dinamiškų svetainių kūrimas • Turinio valdymo sistemos • Web aplikacijos
	1.2. Valdyti PHP programavimo kalbos įrankius ir sintaksę.	<p>Tema. PHP projekto kūrimas ir struktūra</p> <ul style="list-style-type: none"> • Php aplinka, diegimas ir nustatymas • CLI naudojimas • Skriptų paleidimas komandinėje eilutėje • Dažniausiai naudojamos PHP CLI komandos • Projekto kūrimas ir nustatymas <p>Tema. PHP programavimo pagrindai ir sintaksė</p> <ul style="list-style-type: none"> • Sakiniai, išraiškos ir kintamieji • Duomenų tipai • Sąlygos sakiniai: If, if-else, Switch , Ternary operator • Ciklai (for, while, do-while, foreach)

		<ul style="list-style-type: none"> • Vidiniai ciklai, break, continue • Iteracijų kontrolė, naudojant generatorius (yield) • Masyvai (indeksuoti, asociatyvūs), daugiamačiai • Funkcijos ir filtrai (array_map, array_filter) • Tipų deklaracijos (strict_types)
	<p>1.3. Kurti nesudėtingą programinį kodą PHP programavimo kalba.</p>	<p>Tema. Darbas su duomenimis</p> <ul style="list-style-type: none"> • Duomenų savybės (Properties) • Duomenų tipai, jų konvertavimas • Union Types, Nullable • Enum panaudojimas • Duomenų validavimas (filter_var) • Duomenų validavimas ir sanitarizavimas (filter_var) • Duomenų įvestis ir išvestis, naudojant specifines funkcijas (io streams, buffers) • Duomenų skaitymas/rašymas į failą <p>Tema. Standartinės PHP bibliotekos</p> <ul style="list-style-type: none"> • DateTime, PDO, Iterator • Darbas su JSON (json_encode, json_decode) • Monolog (logų valdymas) • Guzzle (HTTP klientas) • Composer (priklausomybių valdymas) <p>Tema. PHP kodavimo standartai</p> <ul style="list-style-type: none"> • PSR-1, PSR-12: kodo formatavimas ir stilistikos gairės • Pavadinimų konvencijos: kintamieji, funkcijos, klasės • Tema: Programinio kodo dokumentavimas • PHPDoc standartas. Anotacijos (@param, @return, @throws) • Dokumentacijos generavimo įrankiai (phpDocumentor) <p>Tema. PHP kodavimo standartai ir jų taikymas programuojant</p> <ul style="list-style-type: none"> • PSR-1, PSR-12: kodo formatavimas ir stilistikos gairės • Pavadinimų konvencijos: kintamieji, funkcijos, klasės • Tema: Programinio kodo dokumentavimas • PHPDoc standartas. Anotacijos (@param, @return, @throws) • Dokumentacijos generavimo įrankiai (phpDocumentor)
	<p>1.4. Programuoti PHP programavimo kalba, taikant</p>	<p>Tema. Objektinis programavimas</p>

	objektinio programavimo principus.	<ul style="list-style-type: none"> • Klasės sąvoka ir sintaksė • Objekto sąvoka • Objekto kūrimas, naudojant raktinį žodį new • Tipų pagal nuorodą (angl. reference type) ir tipų pagal reikšmę (angl. value type) ypatumai ir skirtumai • Klasės nariai (ypatybės (properties), laukai (fields), konstantos (constants)) • Metodai ir jų deklaravimas (angl. method signature) • Metodų parametrų ir argumentų sąvokos ir skirtumai • Parametrų perdavimas į metodą pagal reikšmę (angl. value type) ar pagal nuorodą (angl. reference type) • Konstruktoriai ir destruktoriai • Standartinės PHP klasės (stdClass) • stdClass naudojimo scenarijai • Dinamiškų savybių pridėjimas ir pašalinimas • Prieigos modifikatoriai (angl. access modifiers). Pvz.: private, public, protected • Enkapsuliacija (angl. Encapsulation) • Paveldėjimas (angl. Inheritance). Paveldimumo išplėtimas su parent • Kompozicijos naudojimas • Abstrakcija (abstrakčios klasės ir interfeisai). Trait naudojimas • Statiniai metodai, jų kūrimas ir naudojimas <p>Tema. Išimčių (angl. Exception) gaudymas ir metimas</p> <ul style="list-style-type: none"> • Exception bazinės klasės sąvoką ir ypatumai • Išimčių metimas panaudojant throw raktinį žodį • Naujos išimties klasės aprašymas paveldint Exception klasę • try/catch/finally blokas. try, catch ir finally dalies veikimas. catch dalies veikimas ir paskirtis • Skirtingu tipų išimčių gaudymas ir apdorojimas • Išimčių filtravimas catch dalyje <p>Tema. UML klasių diagramos</p> <ul style="list-style-type: none"> • Klasių ir komponentų sekų diagramos • Klasių kūrimas UML kalba <p>Tema. Dizaino šablonai</p> <ul style="list-style-type: none"> • Dizaino šablonų nauda kuriant programinį kodą • Kūrybiniai (angl. Creational) šablonai. Singleton, Builder, Abstract factory, Factory
--	------------------------------------	---

		<p>method</p> <ul style="list-style-type: none"> • Struktūriniai (angl. Structural) šablonai. Adapter • Elgesio (angl. Behavioral) šablonai. Observer <p>Tema. Algoritmai</p> <ul style="list-style-type: none"> • Rikiavimo algoritmai (bubble, selection, insertion, merge, quicksort) • Algoritmų našumas ir kompleksiskumas (angl. Big-O notation) <p>Tema. Kešavimas</p> <ul style="list-style-type: none"> • Memcached naudojimas • OPcache, Redis naudojimas <p>Tema. Profiliavimas ir derinimas (angl. profiling ir debugging)</p> <ul style="list-style-type: none"> • Profiliavimo samprata • Xdebug ar Blackfire naudojimas <p>Tema. Saugumo ir autentifikacijos metodai</p> <ul style="list-style-type: none"> • JWT, OAuth, token paremta autentifikacija • Slapukais (angl. cookies) paremta autentifikacija • Duomenų šifravimo algoritmai • Įvesties sanitarizavimas ir validavimas: apsauga nuo SQL injekcijų ir XSS atakų • OWASP top 10
	<p>1.5. Testuoti programinę įrangą, naudojant su PHP programavimo kalba suderinamus testavimo įrankius ir metodus.</p>	<p>Tema. Programų testavimas</p> <ul style="list-style-type: none"> • Programų testavimo principai • PHPUnit biblioteka • Programų kūrimo, naudojant Test Driven Development principus pagrindai • Programų derinimas • Žurnalizavimas (Logging) • Programinio kodo derinimas <p>Tema. Programų testavimo įrankiai</p> <ul style="list-style-type: none"> • Testavimo įrankių apžvalga ir palyginimas • Netikrų duomenų (angl. mock) naudojimas vienetu (angl. unit) testuose, naudojant įrankius
	<p>1.6. Naudoti dirbtinio intelekto įrankius programuojant.</p>	<p>Tema. Dirbtinio intelekto (DI) (angl. Artificial intelligence (AI)) pagrindai</p> <ul style="list-style-type: none"> • Mašininio mokymo apžvalga: algoritmai, taikymai ir tendencijos • Kalbos modeliai (angl. large language model (LLM)) ir jų alternatyvos (pvz.: GPT, Gemini, Claude, Grok, Phi4 ir kt.) • Modelių ypatumai (modelio dydis, haliucinacijos, konteksto lango dydis)

		<ul style="list-style-type: none"> • Užklausų dizainas (angl. prompt engineering): užklausos dalys, esamo kodo pateikimas, instrukcijų metodai <p>Tema. Dirbtinio intelekto (DI) (angl. Artificial intelligence (AI)) taikymas</p> <ul style="list-style-type: none"> • Dirbtinio intelekto taikymas programinio kodo rašymui • Dirbtinio intelekto taikymas testų generavimui • Dirbtinio intelekto taikymas dokumentuojant kodą
2. Kurti taikomąsias PHP programas, naudojant <i>Laravel</i> karkasą.	2.1. Paaiškinti programavimo karkasų privalumus ir trūkumus.	<p>Tema. Karkasas</p> <ul style="list-style-type: none"> • Apibrėžimas ir paskirtis • Palyginimas su bibliotekomis • Karkasų tipai (pvz., MVC) <p>Tema. Karkasų apžvalga</p> <ul style="list-style-type: none"> • Populiariausi karkasai skirti PHP programavimo kalbai • Kriterijai, pagal kuriuos renkamos karkasą
	2.2. Kurti taikomąsias žiniatinklio programas su grafine sąsaja, naudojant karkasą.	<p>Tema. <i>Laravel</i> pagrindai</p> <ul style="list-style-type: none"> • URL maršrutų valdymas (angl. Routing) • Attribute-Based Routing • Implicit Route Binding • Tarpinės programos (angl. Middleware) • Apsauga nuo Cross-site request forgery (CSRF) • Kontroleriai • Užklausų objektai (angl. Requests) ir atsakymų objektai (angl. Responses) • Vaizdai (angl. Views) • Dinaminiai URL • Sesijos • Validacija • Klaidų valdymas (angl. Error Handling) • Žurnalizavimas (angl. Logging) <p>Tema. Vartotojo sąsajos kūrimas</p> <ul style="list-style-type: none"> • Šablonų kūrimas • Vite naudojimas resursų kompiliavimas • Lokalizacija • Vartotojo sąsajos architektūra • Resursų kompiliavimas (angl. Compileing Assets) <p>Tema. <i>Laravel</i> ir duomenų bazės</p>

		<ul style="list-style-type: none"> • Užklausų formavimas (angl. Query Builder) • Puslapiavimas (angl. Pagination) • Migracijos (angl. Migrations) • Duomenų pildymas (angl. Seeding) <p>Tema. Objektinio – reliacinio atvaizdavimo (ORM) taikymas</p> <ul style="list-style-type: none"> • Įvadas į ORM • Ryšiai (angl. Relationships) • Kolekcijos (angl. Collections). Lazy Collections • Mutatoriai (angl. Mutators) • API resursai (angl. API Resources) • Serializacija (angl. Serialization)
	<p>2.3. Kurti žiniatinklio programas ir API servisus, naudojant karkasą.</p>	<p>Tema. Aplikacijų kuriamų, naudojant karkasą saugumas</p> <ul style="list-style-type: none"> • Autentifikavimas (Authentication) • API autentifikavimas (API Authentication) • Leidimai (Authorization) • El. Pašto patvirtinimas (Email Verification) • Šifravimas (Encryption, Hashing) • Aplinkos failų šifravimas (File Encryption) • Slaptažodžio atstatymas (Password Reset) <p>Tema. Programų testavimas</p> <ul style="list-style-type: none"> • Testavimo principai • HTTP testavimas • Testavimas per konsolę • Naršyklės testavimas • Duomenų bazių testavimas • Duomenų įterpimas (angl. Mocking)
<p>Mokymosi pasiekimų vertinimo kriterijai</p>	<p>Išsamiai papasakota apie PHP programavimo kalbos atsiradimo istoriją ir pagrindines taikymo sritis. Pademonstruotas PHP programavimo kalbos sintaksės ir įrankių valdymas. Sukurtas nesudėtingas programinis kodas PHP programavimo kalba. Pademonstruoti objektinio programavimo principai programuojant PHP programavimo kalba. Ištestuota programinė įranga, parašyta PHP programavimo kalba, naudojant suderintus testavimo įrankius ir metodus. Pademonstruotas dirbtinio intelekto įrankių naudojimas programuojant. Paaiškinti programavimo karkasų privalumai ir trūkumai. Sukurtos taikomosios žiniatinklio programos su grafine vartotojo sąsaja, naudojant karkasą. Sukurti žiniatinklio programos ir API servisai, naudojant karkasą. Atliekant darbus laikomasi ergonomikos, saugaus darbo, priešgaisrinės saugos, darbo higienos ir aplinkosaugos reikalavimų, užtikrinančių IT specialistų sveikatą, produktyvumą ir saugią darbo aplinką.</p>	

Reikalavimai mokymui skirtiems metodiniams ir materialiesiems ištekliams	<p><i>Mokymo(si) medžiaga:</i></p> <ul style="list-style-type: none"> • Vadovėliai ir kita mokomoji medžiaga • Testas turimiems gebėjimams vertinti <p><i>Mokymo(si) priemonės:</i></p> <ul style="list-style-type: none"> • Techninės priemonės mokymo(si) medžiagai iliustruoti, vizualizuoti, pristatyti
Reikalavimai teorinio ir praktinio mokymo vietai	<p>Klasė ar kita mokymuisi pritaikyta patalpa su techninėmis priemonėmis mokymo(si) medžiagai pateikti (kompiuteriu ir prieiga prie interneto, vaizdo projektoriumi) ir kompiuteriais, skirtais mokinių darbui.</p> <p>Praktinio mokymo klasė (patalpa), aprūpinta kompiuteriais, prieiga prie interneto, programine įranga.</p>
Reikalavimai mokytojų dalykiniam pasirengimui (dalykinei kvalifikacijai)	<p>Modulį gali vesti mokytojas, turintis:</p> <ol style="list-style-type: none"> 1) Lietuvos Respublikos švietimo įstatyme ir Reikalavimų mokytojų kvalifikacijai apraše, patvirtintame Lietuvos Respublikos švietimo, mokslo ir sporto ministro 2014 m. rugpjūčio 29 d. įsakymu Nr. V-774 „Dėl Reikalavimų mokytojų kvalifikacijai aprašo patvirtinimo“, nustatytą išsilavinimą ir kvalifikaciją; 2) Informatikos mokslų studijų kryptių grupės ar lygiavertį išsilavinimą arba vidurinį išsilavinimą ir jaunesniojo programuotojo ar lygiavertę kvalifikaciją, ne mažesnę kaip 3 metų jaunesniojo programuotojo profesinės veiklos patirtį ir pedagoginių ir psichologinių žinių kurso baigimo pažymėjimą.

Modulio pavadinimas – „Informacinių sistemų projektavimas ir kūrimas Python programavimo kalba“

Valstybinis kodas	406131349	
Modulio LTKS lygis	IV	
Apimtis mokymosi kreditais	20	
Asmens pasirengimo mokyti modulyje reikalavimai	<i>netaikoma</i>	
Kompetencijos	Mokymosi rezultatai	Rekomenduojamas turinys mokymosi rezultatams pasiekti
1. Kurti tipinę programinę įrangą Python programavimo kalba.	1.1. Apžvelgti Python programavimo kalbos atsiradimo istoriją ir pagrindines taikymo sritis.	<p>Tema. <i>Susipažinimas su Python programavimo kalba</i></p> <ul style="list-style-type: none"> • Programavimo kalbos istorija • Python versijavimas ir licencija • Python kalbos konceptas <p>Tema. <i>Programavimo kalbos taikymo sritys</i></p> <ul style="list-style-type: none"> • Web aplikacijos • Duomenų analizė ir mokslas • Automatizavimas
	1.2. Valdyti Python programavimo kalbos įrankius ir	<p>Tema. <i>Įvadas į Python programavimo kalbą</i></p> <ul style="list-style-type: none"> • Python panaudojimas

	sintaksę.	<ul style="list-style-type: none"> • Python diegimas ir programavimo aplinkos (VSCode, PyCharm, Jupyter Notebook/Google Colab, IDLE ar pan.) • Virtualios Python aplinkos (pvz.: virtualenv, Anaconda / conda) <p>Tema. Python programavimo kalbos pagrindai ir sintaksė</p> <ul style="list-style-type: none"> • Darbas terminale (pvz. Jupyter Notebook) • Loginiai duomenys (bool) ir jų veiksmi • Skaičių tipai (int, float) ir veiksmi (ypatimai: kėlimas laipsniu **, sveikoji dalyba // , liekana %) <p>Tema. Tekstinė eilutė (str)</p> <ul style="list-style-type: none"> • Darbas su tekstu: len, upper, title, startswith, replace, strip, count, + • Teksto palyginimas: ==, >, <, in • Specialus naujos eilutės simbolis \n • Baitų tipas (byte) (pvz. paveiksluko informacijos nuskaitymui) • Kintamieji: pavadinimo taisyklės, priskyrimas • Įvedimas/išvedimas, naudojant komandinę eilutę (print, input) • Skaičiaus atpažinimas iš teksto (pvz. float('3.14')) • Komentarai • Moduliai, funkcionalumo importavimas (Modulis math ir dažniausiai naudojamos modulio funkcijos ir konstantos) • Python ideologijos gairės: import this • Komanda help • Python modulis kaip konsolės komanda (pvz. python -m calendar 2025) <p>Tema. Python programavimo kalbos valdymo struktūros</p> <ul style="list-style-type: none"> • Sąlygos sakiniai (if/elif/else) • Vizualus programos sekimas, naudojant (pvz. https://pythontutor.com) • Ciklai: while (pradinė situacija, sąlyga, pokytis, amžinas ciklas), for (su range komanda, bei teksto eilutės perrinkimas) • Ciklo valdymas, naudojant break ir continue komandas • Algoritmai: minimalios reikšmės paieška, maksimalios reikšmės paieška, kiekio (angl. count) skaičiavimo algoritmas, paieškos algoritmai
	1.3. Kurti nesudėtingą programinį kodą Python programavimo kalba.	<p>Tema. Funkcijų kūrimas ir naudojimas</p> <ul style="list-style-type: none"> • Deklaravimas ir iškvietimas • Parametrai: paprasti, vardiniai, numatytieji • Gražinama reikšmė

- Funkcijų iškvietimo seka (call-stack)
 - Globalūs ir lokalūs kintamieji
 - Funkcijos parametrų tipų aprašai (angl. type-annotation)
 - Dokumentacija (docstring) ir testiniai atvejai (doctest)
 - Sudėtingesnės galimybės: dekoratoriai (pvz, lru_cache), lambda-funkcijos
 - Nekartokite savęs (angl. Do not Repeat Yourself (DRY)) principas
- Tema. Klaidos ir programos derinimas (angl. debug)**
- Klaidų tipai (SyntaxError, ImportError, NameError/AttributeError, ValueError, ZeroDivisionError, ir kt.)
 - Klaidos pranešimo analizė (traceback, jo analogija call-stack)
 - Programų derinimas, naudojant print komandą arba logging modulį
 - Programų vykdymas pažingsniui, naudojant programavimo aplinką (arba pdb priemonę)
 - Sustojimo taškai breakpoint, vykdymas po vieną eilutę, reikšmių sekimas watch values
 - Klaidų valdymas (raise, try/except)
- Tema. Duomenų struktūros**
- Sąrašai (list, tuple). Deklaravimas, papildymas (append), skaitymas/rašymas pagal indeksą, dalies ištraukimas
 - Nekintantis sąrašas (tuple)
 - Žodynai (dict)
 - Aibės (set/frozenset)
- Tema. Bendri kolekcijų tvarkymo principai**
- Elemento CRUD: įdėjimas, skaitymas, pakeitimas, ištrynimasis
 - Kolekcijų palyginimas, sujungimas
 - Kitos funkcijos: len, sum, max, min, sorted ir pan.
 - Perrinkimas (angl. comprehension), jo panašumai į SQL užklausą
- Tema. Sudėtingi veiksmai su tekstu**
- split, join, indeksavimas, fragmento ištraukimas
 - Teksto formatavimas, naudojant f-string, format
 - Reguliarios išraiškos (re modulis): search, findall, sub, ir pan.
 - JSON modulis (žodyno pavertimas JSON formato tekstu, ir atvirkščiai)
- Tema. Datos ir laiko tipai**
- date ir datetime moduliai
 - now, today komandos
 - Laiko skirtumo tvarkymas (timedelta)

		<ul style="list-style-type: none"> • Datos formatavimas ir atpažinimas iš teksto • Datos vertimas JSON formatu standartiškai nenumatytas (nes nėra visuotinai sutarto formato) <p>Tema. Failų sistema ir duomenų įvedimas/išvedimas</p> <ul style="list-style-type: none"> • Tekstinių failų nuskaitymas su open, read, ir readlines. Rašymo funkcijos su write ir writelines • Fail uždarymo ypatumai (failo uždarymui rekomenduojama naudoti with (context manager)) • CSV failų apdorojimas: skaitymas, rašymas (DictReader, DictWriter) • pathlib modulis Path klasė darbu su failais (cwd , / bei joinpath, glob)
	<p>1.4. Programuoti Python programavimo kalba, taikant objektinio programavimo principus.</p>	<p>Tema. Objektinio programavimo pagrindai</p> <ul style="list-style-type: none"> • Klasė – kompleksinis duomenų tipas • Atributų deklaravimas, naudojant dataclass • Metodų aprašai, self parametras • Objektų kūrimas, konstruktorius • Paveldėjimas, polimorfizmas, inkapsuliacija. Privatūs ir vieši metodai • Išvestinės klasės • UML klasių diagrama • classmethod, staticmethod, property dekoratoriai • Programavimo principai: SOLID, Separation of Concerns <p>Tema. Modulių ir paketų naudojimas</p> <ul style="list-style-type: none"> • Standartinė Python biblioteka • Savo paketų kūrimas ir valdymas, __init__.py failo kūrimas (pvz. bazinių ir išvestinių klasių saugojimas atskiruose failuose) • Trečių šalių paketai. pip naudojimas paketų įdiegimui, bei esamos aplinkos įsiminimui (freeze) <p>Tema. Objektų introspekcija (angl. introspection) bei serializavimas</p> <ul style="list-style-type: none"> • Objektų savybių kolekcija/žodynas, naudojant __dict__ dataclasses.asdict • JSON serializacija (naudojant standartinės priemonės, arba 3-čių šalių priedus, pvz. dataclasses-json), • Objekto duomenų išsaugojimas JSON faile • Objekto inicializavimas pagal duomenis esančius JSON faile
	<p>1.5. Testuoti programinę įrangą, naudojant su Python</p>	<p>Tema. Testavimo principai</p> <ul style="list-style-type: none"> • TDD principas, naudojant doctest

	<p>programavimo kalba suderinamus testavimo įrankius ir metodus.</p>	<ul style="list-style-type: none"> • Pirmiau testai, paskui – funkcijų logikos kūrimas • Testavimas, naudojant unittest, pytest ar pan. • Testavimo priemonės programavimo aplinkoje (IDE) <p>Tema. Kodo kokybė ir standartai</p> <ul style="list-style-type: none"> • PEP8 kodo stiliaus standartas, kodo analizė su pylint, flake8 • black – kodo automatinio formatavimo įrankis • pre-commit hooks • Markdown arba restructured teksto formatas, skirtas dokumentacijai
	<p>1.6. Naudoti dirbtinio intelekto įrankius programuojant.</p>	<p>Tema. Dirbtinio intelekto (DI) (angl. Artificial intelligence (AI)) pagrindai</p> <ul style="list-style-type: none"> • Mašininio mokymo apžvalga: algoritmai, taikymai ir tendencijos • Kalbos modeliai (angl. large language model (LLM)) ir jų alternatyvos (pvz.: GPT, Gemini, Claude, Grok, Phi4 ir kt.) • Modelių ypatumai (modelio dydis, haliucinacijos, konteksto lango dydis) • Užklausų dizainas (angl. prompt engineering): užklausos dalys, esamo kodo pateikimas, instrukcijų metodai <p>Tema. Dirbtinio intelekto (DI) (angl. Artificial intelligence (AI)) taikymas</p> <ul style="list-style-type: none"> • Dirbtinio intelekto taikymas programinio kodo rašymui • Dirbtinio intelekto taikymas testų generavimui • Dirbtinio intelekto taikymas dokumentuojant kodą
<p>2. Kurti taikomąsias programas, naudojant Python karkasus.</p>	<p>2.1. Paaiškinti programavimo karkasų privalumus ir trūkumus.</p>	<p>Tema. Karkasas</p> <ul style="list-style-type: none"> • Apibrėžimas ir paskirtis • Palyginimas su bibliotekomis • Architektūriniai šablonai (pvz.: MVC, MVVM) <p>Tema. Karkasų apžvalga</p> <ul style="list-style-type: none"> • Populiariausi karkasai skirti Python programavimo kalbai • Kriterijai, pagal kuriuos renkamės karkasą
	<p>2.2. Kurti taikomąsias žiniatinklio programas su grafine vartotojo sąsaja, naudojant Python programavimo kalbą.</p>	<p>Tema. Terminalo (CLI - Command Line Interface) programų kūrimas</p> <ul style="list-style-type: none"> • Parametrų nuskaitymas, naudojant sys.argv • Įvedamų duomenų išsaugojimas faile. Išsaugotų duomenų pateikimas <p>Tema. Web programų kūrimas</p> <ul style="list-style-type: none"> • Naršyklės web-developer tools naudojimas • Python Http kliento priemonės: requests modulis • Web programų pagrindai su Flask, FastAPI ar pan. • CRUD API pavyzdys

		<ul style="list-style-type: none"> • async/await naudojimas • html generavimas, naudojant html-šablonus (pvz. jinja2) • html formos ir jų duomenys • http užklausų kvietimas, naudojant curl, postman ar pan. <p>Tema. Darbas su duomenų bazėmis</p> <ul style="list-style-type: none"> • Duomenų bazių (sqlite ar pan.) naudojimo pagrindai • Duomenų bazės prijungimas • SQL užklausos vykdymas, parametrų perdavimas bei saugumo niuansai • SELECT užklausos rezultatų iteravimas • Eilutė kaip sąrašas arba žodynas (pvz. sqlite, žodyno gražinimą reguliuoja row_factory parametras) • ORM pagrindai (SQLAlchemy, SQLAlchemyModel ar pan.) • Duomenų modelio aprašymas (atitikmuo duomenų bazės lentelei) • CRUD veiksmai: įrašo sukūrimas, nuskaitymas, pakeitimas, ištrynimasis • Ryšiai tarp lentelių (angl. relations)
	<p>2.3. Kurti taikomąsias programas, naudojant duomenų analizės ir vizualizacijos karkasus.</p>	<p>Tema. Duomenų analizė ir vizualizacija</p> <ul style="list-style-type: none"> • NumPy duomenų ir skaičiavimų biblioteka • Pandas duomenų analizės biblioteka: • Duomenų masyvas (Series) ir lentelė (dataframe) • Duomenų skaitymas/rašymas iš/į failus CSV, XLSX ir pan. • Duomenų apžvalga: kelios pradinės eilutės (head), duomenų dydis (shape), stulpelių apžvalga ir tipai (info), didžiausios-mažiausios-vidutinės reikšmės ir pan (describe) • Duomenų atrinkimas/filtravimas pagal stulpelius ir eilutes (sąlygos laužtiniuose skliaustuose arba loc, iloc) • Duomenų valymas (replace, map ir pan) • Duomenų lentelės stiliai • Naujų stulpelių skaičiavimas pagal senus • Duomenų indeksai • Duomenų rikiavimas (sort_values, sort_index) • Grupavimas ir agregavimas, (pivot_table, groupby, agg, mean, min, max ir pan.) • Duomenų struktūros pertvarkymas (reshaping), pvz melt, stack, unstack, • Duomenų transformacijos: duomenų apjungimas (merge, join ir pan.) • Duomenų tyrinėjimas (Exploratory Data Analysis): pagrindinės statistikos bei grafikai (pvz, AutoViz)

		<p>Tema. Vizualizacija</p> <ul style="list-style-type: none"> • Vizualizacijos privalumai ir trūkumai • Vizualizacijos bibliotekos: Matplotlib, Plotly, Seaborn ir pan. <p>Tema. Grafikų tipai ir parametrai</p> <ul style="list-style-type: none"> • Linijiniai grafikai (Line plots) • Sklaidos diagramos (Scatter plots) • Stulpeliniai grafikai (Bar charts) • Histogramos (Histograms), Boxplot diagramos • UI kūrimas duomenų analizei (Jupyter interact dekoratorius, Streamlit priemonė ir pan.)
	<p>2.4. Kurti taikomasias programas, naudojant mašininio mokymo modelius.</p>	<p>Tema. Skaitinių modelių naudojimas</p> <ul style="list-style-type: none"> • Korelacių lentelė naudojant Pandas • Baziniai statistiniai mašininio mokymosi metodai su scikit-learn biblioteka: tiesinė regresija, klasterizavimas (k-means ar pan.), duomenų anomalijų nustatymas (z-score principas) <p>Tema. Kalbinių modelių naudojimas</p> <ul style="list-style-type: none"> • LLM (angl. large language model) modelio iškvietimas naudojant API (pvz.: ollama) • RAG (angl. Retrieval-Augmented Generation) pagrindai • LLM naudojimo karkasai (pvz.: LangChain) <p>Tema. AI modelių ir programų ekosistema</p> <ul style="list-style-type: none"> • HuggingFace platforma • GUI bibliotekos mašininio mokymo programoms (pvz.: GradIO)
<p>Mokymosi pasiekimų vertinimo kriterijai</p>	<p>Išsamiai papasakota apie Python programavimo kalbos atsiradimo istoriją ir pagrindines taikymo sritis. Pademonstruotas Python programavimo kalbos sintaksės ir įrankių valdymas. Sukurtas nesudėtingas programinis kodas Python programavimo kalba. Pademonstruoti objektinio programavimo principai programuojant Python programavimo kalba. Ištestuota programinė įranga, parašyta Python programavimo kalba, naudojant suderintus testavimo įrankius ir metodus. Pademonstruotas dirbtinio intelekto įrankių naudojimas programuojant. Paaiškinti programavimo karkasų privalumai ir trūkumai. Sukurtos taikomosios žiniatinklio programos su grafine vartotojo sąsaja, naudojant Python programavimo kalbą. Sukurtos taikomosios programos, naudojant duomenų analizės ir vizualizacijos karkasus. Sukurtos taikomosios programos, naudojant mašininio mokymo modelius. Atliekant darbus laikomasi ergonomikos, saugaus darbo, priešgaisrinės saugos, darbo higienos ir aplinkosaugos reikalavimų, užtikrinančių IT specialistų sveikatą, produktyvumą ir saugią darbo aplinką.</p>	
<p>Reikalavimai mokymui skirtiems metodiniams ir materialiesiems ištekliams</p>	<p><i>Mokymo(si) medžiaga:</i></p> <ul style="list-style-type: none"> • Vadovėliai ir kita mokomoji medžiaga • Testas turimiems gebėjimams vertinti <p><i>Mokymo(si) priemonės:</i></p>	

	<ul style="list-style-type: none"> • Techninės priemonės mokymo(si) medžiagai iliustruoti, vizualizuoti, pristatyti
Reikalavimai teorinio ir praktinio mokymo vietai	<p>Klasė ar kita mokymuisi pritaikyta patalpa su techninėmis priemonėmis mokymo(si) medžiagai pateikti (kompiuteriu ir prieiga prie interneto, vaizdo projektoriumi) ir kompiuteriais, skirtais mokinių darbui.</p> <p>Praktinio mokymo klasė (patalpa), aprūpinta kompiuteriais, prieiga prie interneto, programine įranga.</p>
Reikalavimai mokytojų dalykiniam pasirengimui (dalykinei kvalifikacijai)	<p>Modulį gali vesti mokytojas, turintis:</p> <ol style="list-style-type: none"> 1) Lietuvos Respublikos švietimo įstatyme ir Reikalavimų mokytojų kvalifikacijai apraše, patvirtintame Lietuvos Respublikos švietimo, mokslo ir sporto ministro 2014 m. rugpjūčio 29 d. įsakymu Nr. V-774 „Dėl Reikalavimų mokytojų kvalifikacijai aprašo patvirtinimo“, nustatytą išsilavinimą ir kvalifikaciją; 2) Informatikos mokslų studijų krypties grupės ar lygiavertį išsilavinimą arba vidurinį išsilavinimą ir jaunesniojo programuotojo ar lygiavertę kvalifikaciją, ne mažesnę kaip 3 metų jaunesniojo programuotojo profesinės veiklos patirtį ir pedagoginių ir psichologinių žinių kurso baigimo pažymėjimą.

Modulio pavadinimas – „Žaidimų programavimas naudojant Unity platformą“

Valstybinis kodas	406131350	
Modulio LTKS lygis	IV	
Apimtis mokymosi kreditais	20	
Asmens pasirengimo mokytis modulyje reikalavimai	<i>netaikoma</i>	
Kompetencijos	Mokymosi rezultatai	Rekomenduojamas turinys mokymosi rezultatams pasiekti
1. Naudoti Unity platformą nesudėtingiems žaidimams kurti.	1.1. Paaiškinti žaidimų kūrimo procesą, nagrinėjant žaidimo specifikaciją.	<p>Tema. Žaidimų kūrimo procesas</p> <ul style="list-style-type: none"> • Idėjos generavimas. Žaidimo tema • Žaidimų Tipai. Žaidimo dydis/ilgis • Žaidimo struktūra • Techninė dalis: Update loops, loading assets • Kas yra Game loop ir kaip jį sukurti • Pagrindiniai žaidimų mechanikos, monetizacijos principai ir modeliai • Dizaino pasirinkimas <p>Tema. Žaidimo dizainas ir architektūra</p> <ul style="list-style-type: none"> • Žaidimo variklių apžvalga • Supažindinti su galimais žaidimų varikliais (Unity3D ir kt.) • Varikliu pasirinkimo kriterijai • Žaidimo lygių kūrimas (angl. level design) nuo idėjos iki planavimo • Žaidimo testavimas: ką ir kur testuoti

	<ul style="list-style-type: none"> • Marketingas ir išleidimas • Leidybos būdai: Self-publishing, publisher ieškojimas • Marketingo priemonės. Traileriai. Alpha/Beta <p>Tema. Programuotojo profesinės veiklos aplinka, procesai ir tikslai</p> <ul style="list-style-type: none"> • Veiklos, uždaviniai, kuriuos programuotojas atlieka darbo vietoje • Gerosios praktikos taisyklės ir jų svarba darbo procesui bei rezultatų kokybei
1.2. Valdyti C# programavimo kalbos įrankius ir sintaksę.	<p>Tema. Visual Studio aplinka</p> <ul style="list-style-type: none"> • C# SDK ir Runtime diegimas ir konfigūravimas • C# projekto kūrimas ir konfigūravimas • C# SDK komandinės eilutės įrankių naudojimas • Global Assembly Cache paketų saugykla • NuGet paketų valdymas • Common Language Runtime vykdymo aplinka ir JIT compiler <p>Tema. C# programavimo pagrindai ir sintaksė</p> <ul style="list-style-type: none"> • C# kalbos elementai ir jų funkcijos • Sakiniai, išraiškos ir kintamieji • C# duomenų tipai • Metodai
1.3. Kurti nesudėtingą programinį kodą C# programavimo kalba.	<p>Tema. Darbas su duomenimis</p> <ul style="list-style-type: none"> • Duomenų savybės (angl. Properties) • Duomenų tipai, jų konvertavimas • Duomenų įvestis ir išvestis, naudojant specifines funkcijas (io streams) <p>Tema. LINQ, CSHARP Generics ir kolekcijos</p> <ul style="list-style-type: none"> • Generics bendriniai tipai • Net duomenų sąrašų (List, Array, Dictionary) naudojimas • Linq duomenų išrinkimas iš duomenų objektų <p>Tema. C# Base Class Library (Framework Class Library) klasių biblioteka</p> <ul style="list-style-type: none"> • Enum tipai • BCL (FCL) biblioteka <p>Tema. Kodavimo standartai</p> <ul style="list-style-type: none"> • C# programavimo standartai (angl. coding standard) - kintamųjų, klasių, interfeisų pavadinimų bendros taisyklės • Programinio kodo dokumentavimas, naudojant C# kalbos komentavimo priemones
1.4. Programuoti C#	<p>Tema. Objektinis programavimas</p>

<p>programavimo kalba, taikant objektinio programavimo principus.</p>	<ul style="list-style-type: none"> • Objekto sąvoka (klasė, objektas) • Enkapsuliacija • Paveldėjimas • Polimorfizmas • Abstrakcija ir abstrakčios klasės • Interfeisai • Objektinio programavimo gerųjų praktikų (SOLID principų) taikymas CSHARP: • Įvadas į SOLID projektavimo principus • SRP, OCP, LSP, ISP, DIP principai • Inversion of Control (IoC) konteineriai <p>Tema. Klasių kūrimas UML kalba</p> <ul style="list-style-type: none"> • Klasių ir sekų diagramos • Klasių kūrimas UML kalba <p>Tema. Daugiagijis (Multithreaded) programavimas</p> <ul style="list-style-type: none"> • Thread darbas su gijomis • Procesų sinchronizavimas – lock, monitor, semaphore, waithandle <p>Tema. Asinchroninis programavimas, naudojant C# aplinką</p> <ul style="list-style-type: none"> • Async raktažodis • Task klasė
<p>1.5. Kurti žaidimus Unity aplinkoje, taikant žaidimų kūrimo principus.</p>	<p>Tema. Darbo aplinkos paruošimas</p> <ul style="list-style-type: none"> • Scene navigavimas • Asset parsisiuntimas ir naudojimas • Asset manipuliavimas • Kameros paruošimas / Veikėjo judėjimo Asset parsisiuntimas ir naudojimas <p>Tema. Vartotojo sąsajos Unity aplinkoje kūrimas</p> <ul style="list-style-type: none"> • Kaip paruošti UI • Kas yra canvas ir kaip jį naudoti • Text box naudojimas ir kūrimas (Buttons ir kt.) • Unity fizikinių simuliacijų naudojimas: • Colliders, fiziks, gravity, ir kiti objektų nustatymai <p>Tema. Dalelių (particles) sistemos naudojimas</p> <ul style="list-style-type: none"> • Kaip sukurti naujus efektus • Kaip priskirti prie objektų efektus ir kaip naudoti efektus <p>Tema. Animacijos valdymo principai</p>

		<ul style="list-style-type: none"> • Kaip paruošti UNITY animatorių bei jų naudoti • Kaip naudoti UNITY 2D animatorių • Kaip naudoti UNITY 3D animatorių <p>Tema. UNITY garso kontrolė</p> <ul style="list-style-type: none"> • Kaip kurti naujus garsus • Kaip manipuluoti garsu <p>Tema. Specialiųjų programų naudojimas</p> <ul style="list-style-type: none"> • Speciali Unity programa shaders • High Render pipeline naudojimas <p>Tema. C# kodo pritaikymas Unity objektams</p> <p>Kaip naudoti kodą visiems objektams</p> <p>Kaip keisti bei kurti .png ir .jpeg tipo failus su kodu</p> <p>Tema. Matematikos, geometrijos ir fizikos naudojimas žaidimų programavime</p> <ul style="list-style-type: none"> • Vektoriai ir Matricos • Sinusai, Jėgos, Speed, Gravity ir Curves <p>Tema. Programų testavimas</p> <ul style="list-style-type: none"> • Programų testavimo principai • Programų testavimas, naudojant modulių testų praktiką Visual Studio aplinkoje • Testų tipai • Integracijos testai • Išimtys ir jų naudojimas
	1.6. Naudoti dirbtinio intelekto įrankius programuojant.	<p>Tema. Dirbtinio intelekto (DI) (angl. Artificial intelligence (AI)) pagrindai</p> <ul style="list-style-type: none"> • Mašininio mokymo apžvalga: algoritmai, taikymai ir tendencijos • Kalbos modeliai (angl. large language model (LLM)) ir jų alternatyvos (pvz.: GPT, Gemini, Claude, Grok, Phi4 ir kt.) • Modelių ypatumai (modelio dydis, haliucinacijos, konteksto lango dydis) • Užklausų dizainas (angl. prompt engineering): užklausos dalys, esamo kodo pateikimas, instrukcijų metodai <p>Tema. Dirbtinio intelekto (DI) (angl. Artificial intelligence (AI)) taikymas</p> <ul style="list-style-type: none"> • Dirbtinio intelekto taikymas programinio kodo rašymui • Dirbtinio intelekto taikymas testų generavimui • Dirbtinio intelekto taikymas dokumentuojant kodą
2. Programuoti sudėtingesnius žaidimus,	2.1. Apibūdinti grafinių elementų komponavimo principus.	<p>Tema. Kompozicijos ir grafinio dizaino samprata</p> <ul style="list-style-type: none"> • Grafinio dizaino samprata ir rūšys

<p>naudojant Unity platformą.</p>		<ul style="list-style-type: none"> • Kompozicija – sudedamoji dizaino dalis • Plastinės vizualinės kalbos elementai, jų svarba kūrinio sukūrimui <p>Tema. <i>Vizualieji dizaino elementai</i></p> <ul style="list-style-type: none"> • Plokštuma, taškas, linija, dėmė • Tekstūra, faktūra • Formos ir figūros • Šviesa, šešėlis, siluetas, tamsa <p>Tema. <i>Komponavimo principai</i></p> <ul style="list-style-type: none"> • Pusiausvyra • Simetrija, asimetrija, disimetrija • Statika, dinamika
	<p>2.2. Kurti dvimatės grafikos produktus, naudojant žaidimų kūrimo programinę įrangą.</p>	<p>Tema. <i>Dvimatės vektorinės grafikos kūrimas ir redagavimas</i></p> <ul style="list-style-type: none"> • Dvimatės vektorinės grafikos savybės • Programinė įranga skirta kurti ir redaguoti dvimatę vektorinę grafiką • Vektorinės grafikos programinės įrangos galimybės, programinės įrangos pasirinkimas • Dvimatės taškinės grafikos savybės • Programinė įranga skirta kurti ir redaguoti dvimatę taškinę grafiką • Taškinės grafikos programinės įrangos galimybės, programinės įrangos pasirinkimas <p>Tema. <i>Dvimatės taškinės grafikos kūrimas ir redagavimas</i></p> <ul style="list-style-type: none"> • Linijos, figūros ir figūrų kontūrai, jų savybės, formavimas ir redagavimas • Darbas su vektorinių vaizdų mazgais • Objektai, ruošiniai ir sluoksniai, jų savybės, formavimas ir redagavimas • Spalvinimas ir figūrų užpildai, spalvų sistemos, spalvų redagavimas, permatomumas • Spalvų derinimo principai • Tekstų įkėlimas ir formatavimas, šriftai, šriftų tvarkymas • Puslapiai ir grafikos išdėstymas <p>Tema. <i>Dvimačių taškinių vaizdų kūrimas</i></p> <ul style="list-style-type: none"> • Sluoksniai. Sluoksnių tipai, sluoksnių kūrimas, darbas su sluoksniais. Kaukės • Taškinės grafikos žymėjimas • Taškinės grafikos transformacijos • Taškinės grafikos spalvinis koregavimas, spalvų kanalai • Taškinės grafikos vaizdų retušavimas • Piešimo ir tapymo priemonės, jų panaudojimas kuriant vaizdus • Filtrai ir efektai

		<ul style="list-style-type: none"> • Darbas su tekstais <p>Tema. <i>Vaizdų eksportavimas</i></p> <ul style="list-style-type: none"> • Dvimačių vektorinių vaizdų eksportavimas įvairiais formatais • Dvimačių taškinių vaizdų eksportavimas įvairiais formatais • Taškinių vaizdų vertimas vektoriniais ir atvirkščiai <p>Tema. <i>Dvimačio personažo kūrimas</i></p> <ul style="list-style-type: none"> • Veikėjo kūrimo etapų išmanymas • Personažo eskizavo ir vizualizavimo kūrimas • Tinkamas veikėjo paruošimas programavimui • Scenos sluoksnių kūrimas ir veikimo supratimas • Fonų judėjimo rūšys ir kūrimo metodika <p>Tema. <i>2D Eskizas – pirminė idėja</i></p> <ul style="list-style-type: none"> • Erdvė, perspektyva • Dydis, proporcijos • Tinkamo eskizo parinkimas kuriamo projekto įgyvendinimui • Kūrinio elementų komponavimas: • Skirtingo intensyvumo kompozicijos plokštumoje • Abstrakčios geometrijos kompozicijos • Koliažas • Kaligrafiniai etiudai • Nuotaikų kūrimas spalvomis • Abstrakti harmoninga spalvinė kompozicija <p>Tema. <i>Kūrinio kompozicijos raiškos elementai</i></p> <ul style="list-style-type: none"> • Raiškos elementų konstravimas, harmonizavimas • Plastinės raiškos elementų akcentavimas • Grafinės kompozicijos kūrimas • Spalvinės kompozicijos kūrimas • Erdvinės kompozicijos kūrimas
	<p>2.3. Kurti trimačius grafikos produktus, naudojant žaidimų kūrimo programinę įrangą.</p>	<p>Tema. <i>Trimačių vaizdų kūrimas</i></p> <ul style="list-style-type: none"> • Trimatė aplinka ir orientavimasis joje, peržiūros • Scena, jos nuostatos • Trimačių objektų modeliavimas – tinkleliai, paviršiai, kreivės • Tapyba ir skulptūra trimačiuose objektuose • Medžiagos ir tekstūros

		<ul style="list-style-type: none"> • Trimačių objektų spalvinis koregavimas • Kameros ir apšvietimas • Įvairios realybės simuliacijos • Darbas su trimačiais tekstais • Trimatės grafikos vertimas dvimate ir atvirkščiai <p>Tema. Modelių atvaizdavimas</p> <ul style="list-style-type: none"> • Trimatės grafikos atvaizdavimas įvairiais formatais • Trimačių modelių paruošimas žaidimų kūrimui <p>Tema. Trimačio personažo kūrimas</p> <ul style="list-style-type: none"> • Veikėjo kūrimo etapų išmanymas • Personažo eskizo ir vizualizavimo kūrimas • Tinkamas veikėjo paruošimas programavimui • Trimatės scenos (aplinkos) kūrimas: • Fono kūrimo etapų išmanymas • Fono eskizavimo ir vizualizavimo kūrimas • Tinkamas aplinkos paruošimas programavimui • Scenos sluoksnių kūrimas ir veikimo supratimas <p>Tema. Kompozicijos ir grafinio dizaino samprata</p> <ul style="list-style-type: none"> • Grafinio dizaino samprata ir rūšys • Kompozicija – sudedamoji dizaino dalis • Plastinės vizualinės kalbos elementai, jų svarba kūrinio sukūrimui • Vizualieji dizaino elementai: • Plokštuma, taškas, linija, dėmė • Tekstūra, faktūra • Formos ir figūros • Šviesa, šešėlis, siluetas, tamsa <p>Tema. Komponavimas</p> <ul style="list-style-type: none"> • Pusiausvyra • Simetrija, asimetrija, disimetrija • Statika, dinamika <p>Tema. 3D eskizas – pirminė idėja</p> <ul style="list-style-type: none"> • Erdvė, perspektyva • Dydis, proporcijos • Tinkamo eskizo parinkimas kuriamo projekto įgyvendinimui
--	--	--

		<p>Tema. Kūrinio elementų komponavimas</p> <ul style="list-style-type: none"> • Skirtingo intensyvumo kompozicijos plokštumoje • Abstrakčios geometrijos kompozicijos • Kaligrafiniai etiudai • Nuotaikų kūrimas spalvomis • Abstrakti harmoninga spalvinė kompozicija • Kūrinio kompozicijos raiškos elementai, jų konstravimas, harmonizavimas, bei plastinės raiškos elementų akcentavimas: • Grafinės kompozicijos kūrimas • Spalvinės kompozicijos kūrimas • Erdvinės kompozicijos kūrimas
Mokymosi pasiekimų vertinimo kriterijai	<p>Išsamiai paaiškintas žaidimų kūrimo procesas, nagrinėjant žaidimo specifikaciją. Pademonstruotas C# programavimo kalbos sintaksės ir įrankių valdymas. Sukurtas nesudėtingas programinis kodas C# programavimo kalba. Pademonstruoti objektinio programavimo principai programuojant C# programavimo kalba. Sukurti žaidimai Unity aplinkoje, taikant žaidimų kūrimo principus. Pademonstruotas dirbtinio intelekto įrankių naudojimas programuojant. Apibūdinti grafinių elementų komponavimo principai. Sukurti dvimatės grafikos produktai, naudojant žaidimų kūrimo programinę įrangą. Sukurti trimačius grafikos produktai, naudojant žaidimų kūrimo programinę įrangą. Atliekant darbus laikomasi ergonomikos, saugaus darbo, priešgaisrinės saugos, darbo higienos ir aplinkosaugos reikalavimų, užtikrinančių IT specialistų sveikatą, produktyvumą ir saugią darbo aplinką.</p>	
Reikalavimai mokymui skirtiems metodiniams ir materialiesiems ištekliams	<p><i>Mokymo(si) medžiaga:</i></p> <ul style="list-style-type: none"> • Vadovėliai ir kita mokomoji medžiaga • Testas turimiems gebėjimams vertinti <p><i>Mokymo(si) priemonės:</i></p> <ul style="list-style-type: none"> • Techninės priemonės mokymo(si) medžiagai iliustruoti, vizualizuoti, pristatyti 	
Reikalavimai teorinio ir praktinio mokymo vietai	<p>Klasė ar kita mokymuisi pritaikyta patalpa su techninėmis priemonėmis mokymo(si) medžiagai pateikti (kompiuteriu ir prieiga prie interneto, vaizdo projektoriumi) ir kompiuteriais, skirtais mokinių darbui.</p> <p>Praktinio mokymo klasė (patalpa), aprūpinta kompiuteriais, prieiga prie interneto, programine įranga.</p>	
Reikalavimai mokytojų dalykiniam pasirengimui (dalykinei kvalifikacijai)	<p>Modulį gali vesti mokytojas, turintis:</p> <ol style="list-style-type: none"> 1) Lietuvos Respublikos švietimo įstatyme ir Reikalavimų mokytojų kvalifikacijai apraše, patvirtintame Lietuvos Respublikos švietimo, mokslo ir sporto ministro 2014 m. rugpjūčio 29 d. įsakymu Nr. V-774 „Dėl Reikalavimų mokytojų kvalifikacijai aprašo patvirtinimo“, nustatytą išsilavinimą ir kvalifikaciją; 2) Informatikos mokslų studijų krypties grupės ar lygiavertį išsilavinimą arba vidurinį išsilavinimą ir jaunesniojo programuotojo ar lygiavertę kvalifikaciją, ne mažesnę kaip 3 metų jaunesniojo programuotojo profesinės veiklos patirtį ir pedagoginių ir psichologinių žinių kurso baigimo pažymėjimą. 	

6.3. PASIRENKAMIEJI MODULIAI

Modulio pavadinimas – „Vartotojo sąsajos kūrimas naudojant TypeScript programavimo kalbą“

Valstybinis kodas	406131351	
Modulio LTKS lygis	IV	
Apimtis mokymosi kreditais	5	
Asmens pasirengimo mokytis modulyje reikalavimai	<i>Baigtas šis modulis:</i> Vartotojo sąsajos projektavimas ir programavimas	
Kompetencijos	Mokymosi rezultatai	Rekomenduojamas turinys mokymosi rezultatams pasiekti
1. Kurti tipinę programinę įrangą TypeScript programavimo kalba.	1.1. Apžvelgti TypeScript programavimo kalbos atsiradimo istoriją ir pagrindines taikymo sritis.	Tema. Įvadas į TypeScript programavimo kalbą <ul style="list-style-type: none"> • Programavimo kalbos istorija • TypeScript versijavimas ir licencija • TypeScript kalbos konceptas Tema: TypeScript taikymo sritys <ul style="list-style-type: none"> • Enterprise programos • Web programos • Asinchroninės aplikacijos
	1.2. Valdyti TypeScript programavimo kalbos įrankius ir sintaksę.	Tema: TypeScript aplinka <ul style="list-style-type: none"> • TypeScript programavimo aplinka ir įrankiai • Sąvokos ir jų paskirtis: Source code, Bytecode, TypeScript Compiler, TypeScript Node.js aplinka • Komandinės eilutės komandos (tsc, ts-node, npm run build) • Integruota kūrimo aplinka (IDE) Tema: TypeScript programavimo pagrindai ir sintaksė <ul style="list-style-type: none"> • Kintamieji (deklaravimas, inicializavimas, literalai) • Bendrosios kintamųjų vardų kūrimo taisyklės ir gerosios praktikos • Primityvūs duomenų tipai. Tipų konvertavimas • Primityvių tipų masyvai (Array) • Operatoriai su primityviais tipais: priskyrimo, palyginimo, aritmetiniai, loginiai • Sąlyginiai sakiniai: If, if-else, switch • Ciklai: while, do-while, for, break, continue
	1.3. Kurti nesudėtingą programinį kodą TypeScript programavimo	Tema. Funkcijos <ul style="list-style-type: none"> • Funkcijų apibrėžimas ir tipavimas

	kalba.	<ul style="list-style-type: none"> • Parametrų numatytosios reikšmės • Rest parametrai • Gražinimo tipai <p>Tema. Objektai ir klasės</p> <ul style="list-style-type: none"> • Objektų kūrimas ir manipuliavimas • Klasės kūrimas • get ir set metodai objekto savybėms valdyti • Statiniai nariai <p>Tema. Asinchroninis kodas</p> <ul style="list-style-type: none"> • Promise ir jų naudojimas • Funkcijos async/await • Klaidos tvarkymas asinchroninėse funkcijose (try-catch) <p>Tema. Modulių ir importavimo valdymas</p> <ul style="list-style-type: none"> • Moduliai TypeScript • Eksportai ir importai (pagal nutylėjimą) <p>Tema: Kodavimo standartai</p> <ul style="list-style-type: none"> • TypeScript kodavimo standartai • Dokumentacijos kūrimas naudojant TypeScript dokumentavimo įrankius (pvz., TSDoc)
2. Naudoti TypeScript programavimo kalbą vartotojo sąsajos kūrime.	2.1. Kurti žiniatinklio programas, naudojant TypeScript programavimo kalbą.	<p>Tema. TypeScript ir komponentų kūrimas karkasuose</p> <ul style="list-style-type: none"> • Komponentų tipavimas • Tipų saugumas ir klaidų aptikimas • Interfeisai ir sąsajos komponentams <p>Tema. TypeScript ir duomenų srautų valdymas karkasuose</p> <ul style="list-style-type: none"> • Tipų tikslumas su asinchroniniais duomenimis • Promise ir async/await tipavimas • Integracija su API
	2.2. Testuoti žiniatinklio programas, naudojant TypeScript programavimo kalbą.	<p>Tema: Testavimas</p> <ul style="list-style-type: none"> • Testų rašymas su tipų tikrinimu • Unit testai ir klaidų aptikimas • Testavimo bibliotekų integracija <p>Tema: Testų struktūra ir organizavimas TypeScript projektuose</p> <ul style="list-style-type: none"> • Testų organizavimas pagal komponentus ir funkcijas • Testų atskyrimas nuo verslo logikos • Testų duomenų generavimas su TypeScript

Mokymosi pasiekimų vertinimo kriterijai	Išsamiai papasakota apie TypeScript programavimo kalbos atsiradimo istoriją ir pagrindines taikymo sritis. Pademonstruotas TypeScript programavimo kalbos sintaksės ir įrankių valdymas. Sukurtas nesudėtingas programinis kodas TypeScript programavimo kalba. Sukurtos žiniatinklio programos, naudojant TypeScript programavimo kalbą. Ištestuotos žiniatinklio programos, naudojant TypeScript programavimo kalbą. Atliekant darbus laikomasi ergonomikos, saugaus darbo, priešgaisrinės saugos, darbo higienos ir aplinkosaugos reikalavimų, užtikrinančių IT specialistų sveikatą, produktyvumą ir saugią darbo aplinką.
Reikalavimai mokymui skirtiems metodiniams ir materialiesiems ištekliams	<i>Mokymo(si) medžiaga:</i> <ul style="list-style-type: none"> • Vadovėliai ir kita mokomoji medžiaga • Testas turimiems gebėjimams vertinti <i>Mokymo(si) priemonės:</i> <ul style="list-style-type: none"> • Techninės priemonės mokymo(si) medžiagai iliustruoti, vizualizuoti, pristatyti
Reikalavimai teorinio ir praktinio mokymo vietai	Klasė ar kita mokymuisi pritaikyta patalpa su techninėmis priemonėmis mokymo(si) medžiagai pateikti (kompiuteriu ir prieiga prie interneto, vaizdo projektoriumi) ir kompiuteriais, skirtais mokinių darbui. Praktinio mokymo klasė (patalpa), aprūpinta kompiuteriais, prieiga prie interneto, programine įranga.
Reikalavimai mokytojų dalykiniam pasirengimui (dalykinei kvalifikacijai)	Modulį gali vesti mokytojas, turintis: 1) Lietuvos Respublikos švietimo įstatyme ir Reikalavimų mokytojų kvalifikacijai apraše, patvirtintame Lietuvos Respublikos švietimo, mokslo ir sporto ministro 2014 m. rugpjūčio 29 d. įsakymu Nr. V-774 „Dėl Reikalavimų mokytojų kvalifikacijai aprašo patvirtinimo“, nustatytą išsilavinimą ir kvalifikaciją; 2) Informatikos mokslų studijų krypties ar lygiavertę išsilavinimą arba vidurinį išsilavinimą ir jaunesniojo programuotojo ar lygiavertę kvalifikaciją, ne mažesnę kaip 3 metų jaunesniojo programuotojo profesinės veiklos patirtį ir pedagoginių ir psichologinių žinių kurso baigimo pažymėjimą.

Modulio pavadinimas – „Dirbtinio intelekto taikymas programinės įrangos kūrimo ir testavimo procesuose“

Valstybinis kodas	406131352	
Modulio LTKS lygis	IV	
Apimtis mokymosi kreditais	5	
Asmens pasirengimo mokytis modulyje reikalavimai	<i>Baigtas šis modulis:</i> Vartotojo sąsajos projektavimas ir programavimas	
Kompetencijos	Mokymosi rezultatai	Rekomenduojamas turinys mokymosi rezultatams pasiekti
1. Taikyti bendruosius dirbtinio intelekto veikimo principus informacinių technologijų sektoriaus	1.1. Apibūdinti dirbtinio intelekto sampratą ir veikimo principus.	Tema. <i>Dirbtinio intelekto samprata</i> <ul style="list-style-type: none"> • Dirbtinio intelekto sąvoka ir istorinis vystymasis • Dirbtinio intelekto taikymo sritys Tema. <i>Dirbtinio intelekto veikimo principai</i>

profesinėje veikloje.		<ul style="list-style-type: none"> • Mokymosi algoritmai ir duomenų svarba • Sprendimų priėmimo ir prognozavimo mechanizmai
	1.2. Parinkti dirbtinio intelekto modulius pagal konkrečią situaciją ir sprendžiamą problemą.	<p>Tema. Modulių veikimo principai</p> <ul style="list-style-type: none"> • Modulių klasifikavimas pagal funkcijas: mokymasis, analizė, atpažinimas, prognozavimas, komunikacija • Duomenų įvedimas, apdorojimas, išvesties generavimas • Pagrindinių algoritmų apžvalga (neuroniniai tinklai ir pan.) • Mokymosi tipai: prižiūrimas, neprižiūrimas, sustiprinamasis mokymas <p>Tema. Dirbtinio intelekto modulių rūšys ir klasifikacija</p> <ul style="list-style-type: none"> • Dirbtinio intelekto modulio samprata • Mašininio mokymosi (Machine Learning) moduliai • Giluminio mokymosi (Deep Learning) moduliai • Natūralios kalbos apdorojimo (NLP) moduliai • Kompiuterinio regėjimo (Computer Vision) moduliai
	1.3. Taikyti dirbtinio intelekto modulius praktinėms problemoms spręsti.	<p>Tema. Dirbtinio intelekto modulių taikymas duomenų analizės problemoms spręsti</p> <ul style="list-style-type: none"> • Mašininio mokymosi modulių (sklearn, XGBoost) taikymas klasifikacijai ir regresijai • Klasifikavimo užduotys (pvz.: el. laiškų rūšiavimas) • Regresijos užduotys (pvz.: kainų prognozė) <p>Tema. Duomenų paruošimas ir dirbtinio intelekto mokymas praktikoje</p> <ul style="list-style-type: none"> • Duomenų apdorojimas, normalizavimas, traukimas į treniravimo/validacijos aibes • Modelio kūrimas, testavimas ir rezultatų vertinimas (tikslumas, F1, RMSE)
2. Kurti programinės įrangos sprendimus naudojant dirbtinį intelektą.	2.1. Paaiškinti dirbtinio intelekto taikymo sritis programinės įrangos kūrimo procese.	<p>Tema. Dirbtinio intelekto vaidmuo programinės įrangos kūrime</p> <ul style="list-style-type: none"> • Dirbtinio intelekto įrankių integracija programinės įrangos kūrimo etapuose: projektavimo, testavimo ir priežiūros • Dirbtinio intelekto taikymas analizuojant projektų duomenis, identifikuojant rizikas, planuojat užduotis <p>Tema. Dirbtinio intelekto integravimo iššūkiai programinės įrangos projektuose</p> <ul style="list-style-type: none"> • Technologiniai iššūkiai • Organizaciniai iššūkiai • Etiniai iššūkiai
	2.2. Taikyti dirbtinio intelekto įrankius programinės įrangos reikalavimų analizės ir projektavimo etapuose.	<p>Tema. Reikalavimų analizės etapas</p> <ul style="list-style-type: none"> • Dirbtinio intelekto naudojimas vartotojų poreikių analizei • Dirbtinio intelekto įrankių taikymas reikalavimų tikslinimui ir klasifikavimui <p>Tema. Projektavimo etapas</p>

		<ul style="list-style-type: none"> • Vartotojo sąsajos (UI/UX) dizaino generavimas dirbtinio intelekto įrankiais • Architektūrinių sprendimų parinkimas pasitelkiant dirbtinio intelekto rekomendacijas • Projektavimo klaidų ir rizikų prognozavimas naudojant DI analizę
	2.3. Taikyti dirbtinio intelekto įrankius programavimo ir testavimo etapuose programinės įrangos kūrimo procese.	<p>Tema. Programavimo etapas</p> <ul style="list-style-type: none"> • Dirbtinio intelekto įrankių naudojimas kodo generavimui ir optimizavimui • Automatinis kodo dokumentavimas ir stiliaus analizė • Klaidų aptikimas ir siūlomų pataisymų generavimas • Dokumentacijos generavimas ir atnaujinimas <p>Tema. Testavimo etapas</p> <ul style="list-style-type: none"> • Testų generavimas pagal programos logiką pasitelkiant dirbtinį intelektą • Klaidingo elgesio prognozavimas ir testavimo prioretizavimas naudojant istorinius duomenis ir dirbtinį intelektą
Mokymosi pasiekimų vertinimo kriterijai	Paašškinta dirbtinio intelekto samprata ir veikimo principai. Parinkti dirbtinio intelekto moduliai konkrečiai situacijai ir problemai išspręsti. Pritaikyti dirbtinio intelekto moduliai praktinėms problemoms išspręsti. Paašškintos dirbtinio intelekto taikymo sritys programinės įrangos kūrimo procese. Pritaikyti dirbtinio intelekto įrankiai programinės įrangos reikalavimų analizės ir projektavimo etapuose. Pritaikyti dirbtinio intelekto įrankiai programavimo ir testavimo etapuose programinės įrangos kūrimo procese. Atliekant darbus laikomasi ergonomikos, saugaus darbo, priešgaisrinės saugos, darbo higienos ir aplinkosaugos reikalavimų, užtikrinančių IT specialistų sveikatą, produktyvumą ir saugią darbo aplinką.	
Reikalavimai mokymui skirtiems metodiniams ir materialiesiems ištekliams	<p><i>Mokymo(si) medžiaga:</i></p> <ul style="list-style-type: none"> • Vadovėliai ir kita mokomoji medžiaga • Testas turimiems gebėjimams vertinti <p><i>Mokymo(si) priemonės:</i></p> <ul style="list-style-type: none"> • Techninės priemonės mokymo(si) medžiagai iliustruoti, vizualizuoti, pristatyti 	
Reikalavimai teorinio ir praktinio mokymo vietai	Klasė ar kita mokymuisi pritaikyta patalpa su techninėmis priemonėmis mokymo(si) medžiagai pateikti (kompiuteriu ir prieiga prie interneto, vaizdo projektoriumi) ir kompiuteriais, skirtais mokinių darbui. Praktinio mokymo klasė (patalpa), aprūpinta kompiuteriais, prieiga prie interneto, programine įranga.	
Reikalavimai mokytojų dalykiniam pasirengimui (dalykinei kvalifikacijai)	Modulį gali vesti mokytojas, turintis: 1) Lietuvos Respublikos švietimo įstatyme ir Reikalavimų mokytojų kvalifikacijai apraše, patvirtintame Lietuvos Respublikos švietimo, mokslo ir sporto ministro 2014 m. rugpjūčio 29 d. įsakymu Nr. V-774 „Dėl Reikalavimų mokytojų kvalifikacijai aprašo patvirtinimo“, nustatytą išsilavinimą ir kvalifikaciją; 2) Informatikos mokslų studijų krypties grupės ar lygiavertį išsilavinimą arba vidurinį išsilavinimą ir jaunesniojo programuotojo ar lygiavertę kvalifikaciją, ne mažesnę kaip 3 metų jaunesniojo programuotojo profesinės veiklos patirtį ir pedagoginių ir psichologinių žinių kurso baigimo pažymėjimą.	

6.4. BAIGIAMASIS MODULIS

Modulio pavadinimas – „Įvadas į darbo rinką“

Valstybinis kodas	4000004
Modulio LTKS lygis	IV
Apimtis mokymosi kreditais	5
Kompetencijos	Mokymosi rezultatai
1. Formuoti darbinius įgūdžius realioje darbo vietoje.	1.1. Įsivertinti ir realioje darbo vietoje demonstruoti įgytas kompetencijas. 1.2. Susipažinti su būsimo darbo specifika ir adaptuotis realioje darbo vietoje. 1.3. Įsivertinti asmenines integracijos į darbo rinką galimybes.
Mokymosi pasiekimų vertinimo kriterijai	Siūlomas baigiamojo modulio vertinimas – <i>atlikta (neatlikta)</i> .
Reikalavimai mokymui skirtiems metodiniams ir materialiesiems ištekliams	<i>Nėra.</i>
Reikalavimai teorinio ir praktinio mokymo vietai	Darbo vieta, leidžianti įtvirtinti įgytas jaunesniojo programuotojo kvalifikaciją sudarančias kompetencijas.
Reikalavimai mokytojų dalykiniam pasirengimui (dalykinei kvalifikacijai)	Mokinio mokymuisi modulio metu vadovauja mokytojas, turintis: 1) Lietuvos Respublikos švietimo įstatyme ir Reikalavimų mokytojų kvalifikacijai apraše, patvirtintame Lietuvos Respublikos švietimo, mokslo ir sporto ministro 2014 m. rugpjūčio 29 d. įsakymu Nr. V-774 „Dėl Reikalavimų mokytojų kvalifikacijai aprašo patvirtinimo“, nustatytą išsilavinimą ir kvalifikaciją; 2) Informatikos mokslų studijų krypties grupės ar lygiavertį išsilavinimą arba vidurinį išsilavinimą ir jaunesniojo programuotojo ar lygiavertę kvalifikaciją, ne mažesnę kaip 3 metų jaunesniojo programuotojo profesinės veiklos patirtį ir pedagoginių ir psichologinių žinių kurso baigimo pažymėjimą; Mokinio mokymuisi realioje darbo vietoje vadovaujantis praktikos vadovas turi turėti ne mažesnę kaip 3 metų programuotojo profesinės veiklos patirtį.