

КЫРГЫЗ РЕСПУБЛИКАСЫНЫН ИЛИМ, ЖОГОРКУ БИЛИМ БЕРҮҮ ЖАНА  
ИННОВАЦИЯ МИНИСТРЛИГИ  
И.АРАБАЕВ АТЫНДАГЫ КЫРГЫЗ МАМЛЕКЕТТИК УНИВЕРСИТЕТИ  
ЖАҢЫ МААЛЫМАТ ТЕХНОЛОГИЯЛАР ИНСТИТУТУНУН ОРТО КЕСИПТИК  
БИЛИМ БЕРҮҮ БӨЛҮМҮ



### ЖУМУШЧУ ПРОГРАММАСЫ

Дисциплина: Python тилинде программалоо

Адистиги: 230111 - Компьютердик системаларда программалоо

Окутуунун формасы: күндүзгү

Курс: 1

Семестр: 1,2

Аудиториялык сааттар: 80

Практикалык сабактар: 40/40

Жыйынтыгы: экзамен

Окуу - методикалык комплекс Кыргыз Республикасынын орто билим берүүнүн мамлекеттик билим берүү стандартынын негизинде түзүлдү.

Окуу методикалык комплексин иштеп чыккан: Нурланбек кызы Гүлзат

И.Арабаев атындагы КМУнун ЖМТИнин  
Орто билим берүү бөлүмүнүн жыйынында  
Каралды жана бекитилди

Протокол № 1  
от « 02 » 09 2025г.

Бөлүм башчысы: Н.С. Сейткадиева

И.Арабаев атындагы КМУнун ЖМТИнин  
Окуу-методикалык комиссиясы тарабынан  
бекитилди

Протокол № 1  
от « 04 » 09 2025г.

ОМК председатели:

Бишкек 2025г.

## **1. Дисциплинанын максаттары**

«Python тилинде программалоо» дисциплинасын өздөштүрүүнүн максаттары болуп төмөнкүлөр саналат:

Студенттерге Python тилинин синтаксиси, маалыматтардын структурасы жана алгоритмдик ой жүгүртүүнүн негиздери боюнча базалык билим берүү;

Программалык кодду жазуу, каталарды табуу (debugging) жана заманбап иштеп чыгуу чөйрөлөрүндө практикалык иштерди өз алдынча аткаруу көндүмдөрүн калыптандыруу;

Python тилин системалык автоматташтырууда, маалыматтарды иштеп чыгууда жана инженердик маселелерди чечүүдө колдонуунун заманбап ыкмаларын жана мүмкүнчүлүктөрүн көрсөтүү.

### **1.2. Дисциплинаны окутуунун милдеттери**

Билим берүүчүлүк :

1. Информатика курсунан алынган билимдерди тереңдетүү жана аларды программалоо тармагында системалаштыруу;
2. Python тилинде таза жана стандартка ылайык код жазууну (PEP 8), программалык документтерди түзүүнү жана долбоорлорду туура жасалгалоону үйрөтүү;
3. Алгоритмдештирүүнүн негиздери, маалыматтардын структуралары жана программалык камсыздоонун архитектурасы боюнча базалык билимдерди берүү;
4. Заманбап программалоо технологияларына жана инженердик иштеп чыгууларга болгон кызыгууну арттыруу.

Өнүктүрүүчүлүк :

1. Студенттердин дүйнөнү системалуу-маалыматтык кабыл алуусун калыптандыруу, өз алдынча билим алууга умтулуусун арттыруу жана маалыматтык коомдо кесипкөй адис катары адаптациялануусун камсыз кылуу;
2. Көйгөйлөрдү чечүүдө креативдүүлүктү өнүктүрүү жана программалык маселелерге инженердик-техникалык мамиле кылууга даярдоо;
3. Логикалык жана алгоритмдик ой жүгүртүүнү, деталдарга болгон кунт коюуну жана программалык кодду анализдөө жөндөмүн өнүктүрүү;
4. Техникалык чыгармачылыкты жана программалык камсыздоо чөйрөсүндө инновациялык чечимдерди табуу фантазиясын өстүрүү;
5. Инженердик жоопкерчиликти, интеллектуалдык туруктуулукту жана татаал маселелердин өзөгүн көрө билүү жөндөмүн өнүктүрүү;
6. Программалык камсыздоо менен иштөөдө тактыкка, санариптик эс тутумду жана концентрацияны жогорулатууга үйрөтүү.

Тарбиялык:

1. Студенттердин маалыматтык жана программалоо маданиятын калыптандыруу;
2. Топто иштөөдө (командалык иштеп чыгууларда) толеранттуулукту жана кесиптик этиканы тарбиялоо;
3. Программалык маселелерди чечүүдө жана каталар менен иштөөдө (debugging) максималдуу өз алдынчалыкка жетишүү;

4. Сабакка даярданууда тыкандыкка, жоопкерчиликтүүлүккө жана код жазуудагы тактыкка тарбиялоо;
5. Өзүнүн жумуш убактысын жана иштеп чыгуу этаптарын туура пландоого үйрөтүү;
6. Акыл-эс жана эрк күчүн, кунт коюуну, логикалуулукту жана өнүккөн техникалык элестетүүнү тарбиялоо.

Билиши керек :

1. Кесиптик ишмердүүлүктө маалыматтык технологиялардын жана Python программалоо тилинин мүмкүнчүлүктөрүн;
2. Python тилинин негизги синтаксисин, маалымат типтерин жана конструкцияларын;
3. Негизги алгоритмдерди (и издөө, сорттоо, рекурсия) жана аларды ишке ашыруу жолдорун;
4. Программалык камсыздоону иштеп чыгуунун чөйрөлөрүн (IDE) жана куралдарын;
5. Объектиге багытталган программалоонун (ООП) негиздерин жана принциптерин;
6. Файлдык системалар менен иштөө ыкмаларын жана маалыматтарды сактоо форматтарын (TXT, JSON, CSV);
7. Python тилинин стандарттык жана тышкы китепканаларынын түзүлүшүн.

Кыла алышы керек:

1. Компьютердик жана программалоо каражаттарын практикалык маселелерди чечүүдө колдонуу;
2. Кесиптик ишмердүүлүктө базалык программалык камсыздоону жана Python скрипттерин колдонуу;
3. Ар кандай татаалдыктагы алгоритмдерди түзүү жана аларды кодго айландыруу;
4. Маалыматтардын структураларын (тизмелер, сөздүктөр, топтомдор) түзүү жана иштетүү;
5. Программалык кодогу каталарды табуу (debugging) жана аларды ондоо;
6. Файлдарды автоматтык түрдө иштетүүчү жана системалык процесстерди башкаруучу программаларды жазуу;
7. Ар кандай форматтагы маалыматтарды конвертациялоочу скрипттерди түзүү;
8. Даяр программалык китепканаларды долбоорлорго интеграциялоо.

Өздөштүрүшү керек / Ээ болушу керек (Владеть):

1. Техникалык маалыматты издөө, иштетүү жана практикада колдонуу көндүмдөрүнө;
2. Программалык коду жазуунун жана долбоорлорду туура түзүмдөштүрүүнүн маданиятына;
3. Программалык камсыздоону тестирилөө жана оптималдаштыруу ыкмаларына;
4. Программалоо китепканалары жана API интерфейстери менен иштөө көндүмдөрүнө;
5. Даяр программалык продуктуларды жана скрипттерди тармактык чөйрөгө жайгаштыруу жана колдоо тажрыйбасына.

**Негизги билим берүү программасынын предметтик компетенциясы**  
(НББППК)

Инсанды тарбиялоо чөйрөсүндө 230701 “Колдонмо информатика (тармактар боюнча)”

адистиги боюнча кесиптик орто билим берүүнүн негизги кесиптик билим берүү программасынын максаты студенттердин социалдык жана инсандык сапаттарын: чечкиндүүлүктү, уюшкандыкты, эмгекчилдикти, жоопкерчиликти калыптандыруу болуп саналат. , жарандык, баарлашуу, сабырдуулук, алардын жалпы маданиятын жогорулатуу. ОПОПтун максаттарына жана кесиптик ишмердиктин милдеттерине ылайык, төмөнкү компетенцияларга ээ болушу керек:

а) универсалдуу:

жалпы илимий (ОК):

өз ишин уюштура билүү, кесиптик милдеттерди аткаруунун ыкмаларын жана каражаттарын тандап алуу, алардын натыйжалуулугун жана сапатын баалоо (ОК-1);

көйгөйлөрд чечүү, стандарттуу жана стандарттуу эмес кырдаалдарда чечим кабыл алуу, демилгени жана жоопкерчиликти көрсөтүү

(ОК-2);

кесиптик милдеттерди натыйжалуу аткаруу, кесиптик жана инсандык өнүгүү үчүн зарыл болгон маалыматтарды издөө, чечмелөө жана колдонуу (ОК-3);

заманбап билим берүү жана маалыматтык технологияларды колдонуу менен, жогорку өз алдынчалык менен жаңы билимдерди алууга жөндөмдүү (ОК-9);

өз ишин илимий негиздөө баалай алат, өз ишмердүүлүгүнүн натыйжаларына жогорку өз алдынчалык менен баа бере алат (ОК-10);

кесиптик (РС):

процесстик маалыматтын мазмуну (РС-1);

процесстин динамикалык маалымат мазмуну (ПК-2);

жабдууларды ишке даярдоо (ПК-3);

компьютердин, перифериялык түзүлүштөрдүн жана телекоммуникациялык системалардын иштешин контролдоо, алардын нормалдуу иштешин камсыз кылуу (ПК-5);

Бул предметтик компетенциялар студенттерге программалоо жана алгоритмдөө жаатында ийгиликтүү иш алып барууга жардам берип, аларды башка технологиялык же IT чөйрөлөрүндө дагы кеңири колдоно билүүгө даярдайт. Ар бир компетенция окуучулардын жеке жана кесиптик өсүүсүнө багытталган.

## 11.1 Окуу предметтеринин өз ара байланыштары:

**Пререквизиттер:** «Системалык жана колдонмо программалоо» предметин ийгиликтүү өздөштүрүү үчүн студент математикалык логиканын негиздерин, алгоритмдештирүүнүн базалык принциптерин жана мектептик «Информатика» курсун терең билиши керек. Студент компьютердин түзүлүшү, файлдык системалар, иштөө системаларынын (Windows/Linux) негизги функциялары жана тексттик код редакторлору менен иштөө боюнча баштапкы түшүнүктөргө ээ болуусу зарыл.

**Постреквизиттер:** «Системалык жана колдонмо программалоо» предмети кесиптик дисциплиналардын бүтүндөй циклин өздөштүрүү үчүн базалык пайдубал болуп саналат. Бул курста алынган Python программалоо көндүмдөрүнүн негизинде студенттер төмөнкү сабактарды терең үйрөнө алышат:

- «Объектиге багытталган программалоо» (Python ООП);
- «Маалыматтар базасын башкаруу системалары» (SQL жана Python өз ара байланышы);

- «Веб-программалоо жана тармактык технологиялар»;
- «Маалыматтарды талдоо жана жасалма интеллект негиздери

## Окуу предметинин программалык мазмуну.

### I. Курстун сүрөттөлүшү

Python тилинде программалоо курсу бул студенттерге программалоонун негизги принциптерин жана алгоритмдик ой жүгүртүүнү үйрөтүүгө багытталган. Курс студенттерге ар кандай маселелерди чечүү үчүн алгоритмдерди түзүүгө, бул алгоритмдерди программалоо тилдери менен ишке ашырууга жана программанын натыйжалуулугун баалоого жардам берет.

Курс практикалык багытта түзүлүп, студенттерди программалоо жана алгоритмдер менен иштөөнүн теориялык жана практикалык жактарын бириктирип окутат. Окуучуларга ар кандай программалык маселелерди чечүү үчүн алгоритмдерди түзүү, талдоо жана коддоо процесстерин өздөштүрүүгө жардам берет.

Негизги темалар:

Алгоритмдин негизги түшүнүктөрү: Алгоритм деген эмне, алгоритмдин түрлөрү жана аны ишке ашыруу жолдору.

Программалоонун негизги принциптери: Негизги структуралар, шарттык операторлор (if-else), циклдер (for, while), функциялар.

Маалымат структуралары: Массивдер, тизмелер, стек, кезек, хэш таблицалар, жана графиктер.

Алгоритмдер: Сорттоо жана издөө алгоритмдери, рекурсия, динамикалык программалоо.

Программа түзүү процесси: План түзүү, код жазуу, тестирлөө жана оңдоо (отладка).

Программалык долбоорлор менен иштөө: Студенттер долбоорлор аркылуу программалоо көндүмдөрүн иш жүзүндө колдонуп үйрөнүшөт.

Курс информатиканы андан ары тереңдетип үйрөнүү жана программалык камсыздоо системаларын иштеп чыгуу максатында программалоону

Предметти окуу планы (практикалык иш):

№	Практикалык иштин темасы	саат
1-жарым жылдык 1-модуль		
1	Python чөйрөсүн тууралоо (VS Code). Киргизүү жана чыгаруу буйруктары (input/print).	2
2	Өзгөрмөлөр жана маалыматтардын типтери (int, float, str). Типтерди өзгөртүү.	2
3	Математикалык амалдар жана операторлор. Сызыктуу программаларды түзүү.	2
4	Математикалык китепкана (math) менен иштөө. Татаал формулаларды эсептөө.	2
5	Практикалык тапшырма: Геометриялык фигуралардын параметрлерин эсептөө.	2
6	Шарттуу операторлор (if, elif, else). Логикалык туюнтмалар.	2
7	Түзүмдүү тармактуу алгоритмдер. Кабатталган шарттар.	2
8	match-case оператору (Python 3.10+). Тандоо алгоритмдери.	2
9	while циклин колдонуу. Шарттуу кайталануулар.	2
10	for циклин колдонуу. range() функциясы менен иштөө.	
I Жарым жылдык 2-модуль		
1	Уюшкан (кабатталган) циклдер. Көпөлүү таблицаларын жана фигураларды түзүү.	2
2	break, continue жана pass операторлорун колдонуу.	2
3	Саптар (strings) менен иштөө. Индексөө жана кесүү (slicing).	2
4	Саптык методдор (find, replace, split, join). f-саптар менен форматтоо.	2
5	Тизмелер (lists) менен таанышуу. Тизме элементтерин кошуу жана өчүрүү.	2
6	Тизмелерди сорттоо жана издөө алгоритмдери.	2
7	Кортеждер (tuples) жана алардын өзгөчөлүктөрү.	2
8	Сөздүктөр (dictionaries) менен иштөө. Ачкыч жана маани түшүнүгү.	2
9	Топтомдор (sets) жана алар менен болгон логикалык операциялар.	2
10	Өз алдынча функцияларды түзүү (def). Аргументтер жана кайтарылган маани.	2

		ЖАЛПЫ СААТ:	40
		II жарым жылдык (1-модуль)	
1	Функциялардагы локалдык жана глобалдык өзгөрмөлөр.		2
2	Рекурсия түшүнүгү жана рекурсивдик функцияларды түзүү.		2
3	Лямбда-функциялар (анонимдүү функциялар) жана аларды колдонуу.		2
4	Модульдерди түзүү жана импорттоо (import, from...import).		2
5	Файлдарды ачуу жана окуу режимдери (open, read, readline).		2
6	Файлдарга маалымат жазуу жана кошуу (write, append).		2
7	менен иштөө. JSON файлдары Маалыматтарды структуралаштырылган сактоо.		2
8	Саптык методдор (find, replace, split, join). f-саптар менен форматтоо.		2
9	Топтомдор (sets) жана алар менен болгон логикалык операциялар.		2
10	CSV файлдары (таблицаалар) менен иштөө. Маалыматтарды талдоо.		2
11	ОС (os) модулу менен иштөө. Папкаларды түзүү, өчүрүү жана файлдарды тизмектөө.		2
12	Практикалык иш: "Автоматтык файл сорттоочу" скриптин жазуу.		2
		II жарым жылдык (2-модуль)	
			2
1	sys китепканасы: Терминалдан аргументтерди кабыл алуу.		2
2	Класстар жана объекттер. Конструктор init.		2
3	Класстардын атрибуттары жана методдору.		2
4	Мурастоо (Inheritance) түшүнүгү. Ата-эне жана бала класстар.		2
5	Полиморфизм жана инкапсуляция негиздери.		2
6	ООП негизинде кичи долбоор: "Банктык эсеп системасы".		2
7	Каталарды кармоо (try, except, finally). Программанын туруктуулугун камсыздоо.		2
8	datetime модулу: Убакыт жана дата менен иштөө.		2
		ЖАЛПЫ СААТ:	40

### Баалоо критерийлери:

№	Текшерүү этаптары	Текшерүү куралынын түрү	Балл
1	1-модуль	Практикалык тапшырмаларды текшерүү. Оозеки, тестирлөө. Сабакка катышуусу.	100
2	2-модуль	Практикалык тапшырмаларды текшерүү. Тестирлөө. Сабакка катышуусу.	100
3	Акыркы көзөмөл: • Практикалык сабак;	Тесттер жана графикалык иштер, рефераттар, презентациялар, өз алдынча иш,	100
	СӨИ.	практикалык тапшырмалар. Тестирлөө. Сабакка катышуусу.	
	Жалпы орточо бал		<b>100</b>

		Канааттандырарлык	Жакшы	Эң жакшы
1-модуль – 100 б.		60-79	80-89	90-100
2-модуль – 100 б.		60-79	80-89	90-100
Практикалык сабак – 50 б.	Акыркы көзөмөл	60-79	80-89	90-100
СӨИ – 50 б.				

## Практикалык сааттардын темасы

№	Темаларды аталышы	Мазмуну
1	Ал алгоритм. Ал алгоритм касиеттери. Алгоритмдерди жазуу үчүй формалар. Функционалдык блоктордун максат	Бул тема алгоритмдердин негиздерин жана алардын касиеттерин тушундурот. Ал алгоритмдерди жазуунун ар кандай формаларын (сөз менен жазуу, блок-схемалар, псевдокод) жана блок-схемадагы функционалдык блоктордун ролун чагылдырат. Мында алгоритмдердин аныктуулук, чечимдүүлүк, жалпылык сыяктуу негизги касиеттери каралат. Тема алгоритмдерди логикалык жана визуалдык формада көрсөтүүгө багытталган.
2	Маселелерди чечүүнүн негизги этаптары. Алгоритмдердин негизги структуралары. Маалыматтар жана алардын түрлөрү.	Бул тема маселелерди чечүүнүн этаптарын (маселени түшүнүү, чечүү планы, алгоритм түзүү, текшерүү) жана алгоритмдердин негизги структураларын (сызыктуу, тармакталган, циклдык) талкуулайт. Ошондой эле, маалыматтардын түрлөрү (сандык, символдук, логикалык) жана аларды иштеп чыгуу жолдору каралат. Тема программалоодогу негизги логикалык структуралар жана маалыматтар менен иштөө принциптерин түшүндүрөт.
3	Программалоо тилдери. Программалоо тилдеринин классификациясы. Программалоонун ыкмалары жана принциптери. Программалык камсыздоонун түрлөрү (программалык камсыздоо)	Бул тема программалоо тилдеринин негизги түшүнүгүн жана аларды классификациялоону (төмөнкү жана жогорку деңгээлдеги тилдер) карайт. Программалоо ыкмалары (процедуралык, объектке багытталган) жана принциптери да талкууланат. Ошондой эле, программалык камсыздоонун түрлөрү (системалык, колдонмо жана инструменталдык программалар) баяндалат. Лема программалоонун негизги концепцияларын жана программалык камсыздоонун түрлөрүн түшүндүрөт.
4	Python программалоо тилине киришүү. Python чөйрөсү. Жумушчу чөйрөнүн биринчи ачылышы. Python тилинин негизги элементтери.	Бул тема Python программалоо тилине киришүүнү, анын өзгөчөлүктөрүн жана иштөө чөйрөсүн тааныштырат. Python чөйрөсүнүн биринчи ачылышы, иштөө процесси жана негизги элементтер (өзгөрмөлөр, маалымат түрлөрү, операторлор) каралат. Тема студенттерге Python негиздерин жана алгачкы иш баштоо жолдорун көрсөтүүгө багытталган.
5	Python тилиндеги	Бул тема Python программасында операциялар (арифметикалык, логикалык), өзгөрмөлөрдү түзүү,

	операциялар, өзгөрмөлөр, литералдар, маалымат түрлөрү. Python программаларында маалыматтарды	жана ар кандай литералдар менен маалымат түрлөрүн (сандар, саптар, логикалык маанилер) талкуулайт. Ошондой эле, Python маалыматтарды киргизүү жана чыгаруу ыкмалары (input(), print()) каралат. Тема операторлордун жана туюнтмалардын ролун жана алардын программалардагы иштөө
6	Сызыктуу алгоритмдер: бүтүн, реалдуу, логикалык маалыматтар боюнча операциялар.	Бул тема сызыктуу алгоритмдерди, алардын структурасын жана иштөө принциптерин карайт. Бүтүн, реалдуу жана логикалык маалыматтар боюнча операцияларды (жыйноо, бөлүү, логикалык салыштыруу) колдонуу жолу менен алгоритмдердин түзүлүшү жана аткарылышы чагылдырылат. Тема студенттерге сызыктуу алгоритмдердин негизги
7	Python тилиндеги даяр модулдар Math, Random.	Бул тема Python программалоо тилиндеги даяр модулдарды, атап айтканда, Math жана Random модулдарын карайт. Math модули математика менен байланышкан функцияларды жана константаларды (мисалы, квадрат тамыр, тригонометриялык функциялар) сунуштайт. Ал эми Random модули кездемелерди чыгаруу, сан же элемент тандоо сыяктуу функцияларды камсыз кылат. Тема студенттерге бул модулдардын мүмкүнчүлүктөрүн жана алардын
8	Тармактуу алгоритми: Жөнөкөй шарттуу оператор, Кыскартылган шарттуу оператор, Курама шарттуу оператор.	Бул тема тармактуу алгоритмдерди, анын ичинде жонокой, кыскартылган жана курама шарттуу операторлорду карайт. Жөнөкөй шарттуу оператор белгилүү бир шарттын негизинде бир аракетти же иш-аракеттер тобун аткарууга мүмкүндүк берет. Кыскартылган шарттуу оператор бир эле сызыктагы шарттуу операторду колдонуп, коду кыскартып жазууга жардам берет. Ал эми Курама шарттуу оператор бир нече шарттарды бириктирип, татаал логикалык чечимдерди кабыл алууга мүмкүндүк берет. Тема студенттерге шарттуу операторлорду колдонуу аркылуу алгоритмдерди логикалык түрдө түзүүгө жана

9	<p>Көп тармактуу бутактар.</p> <p>Максималдуужана минималдуу элементтерди табуу алгоритмдери.</p>	<p>Бул тема көп тармактуу бутактарды жана алардын алгоритмдердеги ролун, ошондой эле максималдуу жана минималдуу элементтерди табуу алгоритмдерин карайт. Көп тармактуу бутактар бир жана шарттарга негизделген иш-аракеттерди аткарууга мүмкүндүк берет, бул алгоритмдердин татаалдылыгын жогорулатат. Максималдуу жана минималдуу элементтерди табуу алгоритмдери массив же тизмедеги элементтерди салыштырып, эн жогорку же эн төмөнкү маанини аныктоого арналган. Тема</p>
10	<p>Цикл алгоритми: While циклинин билдирүүсү. Инструкцияларды үзүп, улантыңыз.</p>	<p>Бул тема цикл алгоритмдерин, конкреттүү алганда While циклин жана анын билдирүүсүн карайт. While цикли белгилүү бир шарт туура болсо, инструкцияларды кайра-кайра аткарууга мүмкүндүк берет. Цикл ичиндеги инструкцияларды үзүп, циклды улантуу же токтотуу механизмдери (мисалы, break жана continue операторлору) түшүндүрүлөт. Тема студенттерге цикл</p>
11	<p>Цикл оператору үчүй. range() функциясы.</p>	<p>Бул тема цикл операторлору учун range() функциясын жана анын колдонулушун карайт. range() функциясы белгилүү бир интервалда бүтүн сандарды генерирлөө үчүн колдонулат, ал көбүнчө for цикли менен биргеликте колдонулат. Функция үч параметрди кабыл алат: башталгыч мааниси, акыркы мааниси жана кадам, бул болсо цикл аркылуу кайталанууну контролдоо үчүн пайдалуу. Тема студенттерге цикл операторлорунун логикасын жана range() функциясынын</p>
12	<p>Уюшкан циклдер.</p>	<p>Бул тема уюшкан циклдерди, алардын структурасын жана колдонулушун карайт. Уюшкан циклдер программалоодо бир нече инструкцияларды бир нече жолу аткарууга мүмкүндүк берет, адатта, for же while циклдери аркылуу ишке ашырылат. Уюшкан циклдер белгилүү шарттарды канааттандырганда же массивдер же тизмелер сыяктуу маалымат структуралары аркылуу элементтерди кайталоо</p>

		логикалык уюштурулушун жана алгоритмдерде колдонуу ыкмаларын үйрөнүүгө жардам берет.
13	Саптар менен шитөө: негизги түшүнүктөр, алар менен шитөө. Стринг ыкмалары. Сапты иштетүүнүн негизги алгоритмдери.	Бул тема саптар менен иштөөнүн негизги түшүнүктөрүн, алардын мүнөздөмөлөрүн жана Python саптар менен иштөө ыкмаларын карайт. Саптар - бул тексттик маалыматтарды сактоонун жана манипуляциялоонун негизги формасы, жана алардын узундугу, символдору сыяктуу атрибуттары болот. Саптар менен иштөөдө колдонулган методдор (мисалы, upper(), lower(), split(), join()) жана алгоритмдер (сапты бөлүү, түзүү, же өзгөртүү) талкууланат. Тема студенттерге саптар менен иштөө ыкмаларын жана
14	Python топтомдор менен иштөө.	Бул тема Python топтомдор менен иштөөнү жана алардын негизги концепцияларын карайт. Топтомдор - бул маалыматтарды сактоо үчүн колдонулган курама структуралар, ал ар кандай элементтерди, анын ичинде башка топтомдорду да камтый алат. Топтомдорду түзүү, узартуу, элементтерди кошуу, жок кылуу жана издөө ыкмалары (мисалы, add(), remove(), pop(), union(), intersection()) талкууланат. Тема студенттерге топтомдорду колдонуу аркылуу маалыматтарды эффективдүү башкарууну жана манипуляциялоону үйрөнүүгө жардам берет.

15	Python тилинде тизмелер менен иштөө.	<p>Бул тема Python тилинде тизмелер менен иштөөнү жана алардын негизги функцияларын карайт.</p> <p>Тизме — бул маалыматгарды сактоо үчүн колдонулган динамикалык структура, ал ар кандай элементтерди (сан, сап, объект) камтый алат.</p> <p>Тизменин негизги операциялары, мисалы, элементтерди кошуу, өчүрүү, өзгөртүү жана издөө (мисалы, <code>append()</code>, <code>remove()</code>, <code>sort()</code>, <code>index()</code>) талкууланат. Тема студенттерге тизмелерди колдонуу аркылуу маалыматгарды уюштуруу жана манипуляциялоону үйрөнүүгө жардам берет.</p>
16	Python тилинде кортеждер менен иштөө.	<p>Бул тема Python тилинде кортеждер менен иштөөнү жана алардын негизги өзгөчөлүктөрүн карайт. Кортеждер - бул <code>immutable</code> (өзгөртпөс) маалымат структуралары, алар элементтердин тизмесин сактоо үчүн колдонулат. Кортеждердин элементтери</p>

		ар кандай типте болушу мүмкүн жана аларды тузу, индексациялоо жана бөлүштүрү (мисалы, count(), index()) ыкмалары талкууланат. Тема студенттерге кортеждерди колдонуу аркылуу маалыматтарды сактоонун эффективдүү ыкмаларын үйрөнүүгө жардам берет.
17	Python тилиндеги сөздүктөр менен иштөө.	Бул тема Python тилиндеги сөздүктөрдү (dictionary) жана алардын негизги функцияларын карайт. Сөздүктөр - бул элементтерди жуптар түрүндө сактоочу курама структура, анда ар бир элементтин ачкычы (key) жана мааниси (value) болот. Сөздүктөрдү түзүү, элементтерди кошуу, өчүрүү жана өзгөртүү ыкмалары (мисалы, get (), keys ( ), values(), items()) талкууланат. Тема студенттерге сөздүктөрдү колдонуу аркылуу маалыматтарды эффективдүү уюштуруу жана жеткиликтүү кылууну үйрөтүүгө жардам берет.
18	Уюшкан тизмектерди (УТ) иштетүү. УТ тип негизги алгоритмдери.	Бул тема уюшкан тизмектерди (УТ) иштетүүнү жана алардын негизги алгоритмдерин карайт. УТ - бул маалыматтарды сактоонун структурасы, ал элементтерди топтоштуруп, бир нече функцияларды аткарууга мүмкүндүк берет, мисалы, кошуу, өчүрүү, издөө жана сорттоо. Уюшкан тизмектерде иштөө үчүн негизги алгоритмдер (мисалы, элементтерди алуу, массивди сорттоо, издеелер) талкууланат. Тема студенттерге уюшкан тизмектерди колдонуу аркылуу маалыматтарды эффективдүү башкаруу жана манипуляциялоо ыкмаларын үйрөнүүгө жардам берет.
19	Функциялар менен иштөө: Ыңгайлаштырылган функцияларды түзүү.	Бул тема функциялар менен иштөөнү жана ыңгайлаштырылган функцияларды түзүүнү карайт. Функциялар - бул кодду модулдаштыруу жана кайталоолордон качуу үчүн колдонулган структуралар, алар белгилүү бир операцияларды аткарат. Ыңгайлаштырылган функциялар студенттердин муктаждыктарына жараша түзүлүп, параметрлерди кабыл алып, натыйжаларды кайтарат. Тема функцияларды туура колдонуу, алардын маанисин, жана аргументтер менен кайтаруучу манилерди башкарууну үйрөнүүгө жардам берет.
20	Модульдарды түзүү.	Бул тема модульдарды түзүүнү жана Python

	Стандарттык библиотекадагы модулдар менен ишөө.	стандарттык библиотекасындагы модулдар менен иштөөнү карайт. Модулдар - бул кодду уюштуруу жана кайра колдонуу үчүн пайдалуу структуралар, алар функциялар, класстар жана өзгөрмөлөрдү камтыйт. Стандарттык библиотекадагы модулдар (мисалы, math, random, datetime) ар кандай функционалдар менен камсыз кылат жана программаларды жазууда убакытты жана күчтү унөмдөөгө жардам берет. Тема студенттерге модулдарды түзүү жана колдонуу аркылуу программалоонун натыйжалуулугун жогорулатууга жардам берет.
21	Ташбака модулу менен сүрөт тартуу.	Бул тема Python Ташбака модулун (Turtle module) колдонуп сүрөт тартууна карайт. Ташбака модулу графикалык сүрөттөрдү жаратуу үчүн колдонулган жөнөкөй инструмент, ал программистке экранда сүрөттөрдү жана формаларды тартууга мүмкүндүк берет. Ташбака модулундагы функциялар (мисалы, forward(), right(), circle()) сүрөт тартуу процессин жеңилдетет. Тема студенттерге Ташбака модулунун мүмкүнчүлүктөрүн жана графикалык программалоо ыкмаларын үйрөнүүгө жардам берет.
22	Turtle модулундагы функцияларды колдонуу.	Бул тема Turtle модулундагы функцияларды колдонууну жана алардын графикалык сүрөт тартууда кандайча пайдалануусун карайт. Turtle модулундагы негизги функциялар, мисалы, forward(), backward(), left(), right(), жана penup() сыяктуу командалар, сүрөт тартуу процессин жөнөкөй жана интуитивдуу кылат. Студенттер Turtle модулунун функциялары аркылуу суреттерди тартууда жана графикалык элементтерди түзүүдө өздөрүнүн креативдүүлүгүн колдонуп, программалоо боюнча түшүнүктөрүн тереңдетүүгө мүмкүнчүлүк алышат. Тема графикалык программалоону үйрөнүүнү жеңилдетет.

Адабияттар:

**Электрондук ресурстар:**

1. <http://www.twirpx.com/file/761154/>,

2. <http://dfiles.ru/files/7osfbmniij>

А) Негизги адабияттар:

1. В. В. Трафимова. Информатика. Москва 2015г.

2. Н. В. Храмова, К. М. Мырзакулова, Н. Ж. Мамбеталиева, Н. А.

Жамангулова. Учебно – методическое пособие по информатике. Бишкек –

2016г.

3. Н. В. Макарова, В. Б. Волков. Информатика. Москва 2011г.

4. А. В. Кузин, В. М. Демин. Разработка базы данных в системе MS Access 3-е издание. Москва 2009г.

5. Б. Ж. Баячорова. Основа информатики. Бишкек 2004г.

Б) Кошумча адабияттар:

6. Ю. Д. Романовой. Информатика и информационные технологий. Москва 2011г.

7. В. Долженков, А. Стученков. Самоучитель Excel 2010. Санкт – Петербург 2011г.

8. У. А. Дюшенбиев, У. Э. Мамбетакунов, Н. В. Храмова. Windows в вопросах и ответах. Бишкек 2003г.

### 13. Глоссарий (Сөздүк)

#### **Python (Пайтон)**

Жогорку деңгээлдеги, интерпретациялануучу программалоо тили. Ал жөнөкөй синтаксиси, окумдуулугу жана универсалдуулугу (веб-иштеп чыгуу, маалымат илими, жасалма интеллект ж.б.) менен айырмаланат.

#### **Интерпретатор (Interpreter)**

Программалык кодду сап-сап кылып окуп, аны дароо аткаруучу атайын программа. Пайтон тили интерпретациялануучу тилдерге кирет.

#### **IDE (Integrated Development Environment)**

Программалоо үчүн интеграцияланган иштеп чыгуу чөйрөсү. Код жазуу, каталарды табуу жана программаны ишке киргизүү үчүн куралдардын жыйындысы (Мисалы: VS Code, PyCharm, IDLE).

#### **Өзгөрмө (Variable)**

Маалыматтарды сактоо үчүн колдонулган компьютердин эс тутумундагы аталган уяча. Пайтондо өзгөрмөнүн тиби маани ыйгарылганда автоматтык түрдө аныкталат (динамикалык типтештирүү).

#### **Маалыматтардын типтери (Data Types)**

Программада иштетилүүчү маалыматтардын классификациясы. Негизги типтери: int (бүтүн сан), float (ондук сан), str (сап/текст), bool (логикалык маани).

#### **Тизме (List)**

Пайтондогу маалыматтардын иреттелген жыйындысы. Тизмедеги элементтерди өзгөртүүгө, кошууга жана өчүрүүгө болот. Квадраттык кашаалар [] менен белгиленет.

#### **Сөздүк (Dictionary)**

Маалыматтарды "ачкыч-маани" (key-value) жуптары түрүндө сактоочу структура. Тартипсиз жыйынды болуп саналат жана тармал кашаалар {} менен белгиленет.

#### **Кортеж (Tuple)**

Өзгөртүүгө мүмкүн болбогон (immutable) маалыматтардын иреттелген жыйындысы. Тегерек кашаалар () менен белгиленет.

#### **Функция (Function)**

Программанын каалаган жеринде кайталап колдонууга мүмкүн болгон коддун логикалык блогу. Пайтондо функциялар def ачкыч сөзү менен аныкталат.

#### **Цикл (Loop)**

Белгилүү бир шарт аткарылганга чейин коддун бир бөлүгүн кайталап аткаруу механизми. Пайтондо негизинен for жана while циклдери колдонулат.

#### **Шарттуу оператор (Conditional Operator)**

Программада белгилүү бир шартка жараша чечим кабыл алуу үчүн колдонулуучу конструкция (if, elif, else).

#### **Объектке багытталган программалоо (ОБП / ООР)**

Программаны "объекттер" жана "класстар" түрүндө моделдөөгө негизделген ыкма. Пайтон толугу менен ОБП принциптерине (мурастоо, инкапсуляция, полиморфизм) негизделген.

#### **Класс (Class)**

Объект түзүү үчүн колдонулган "чийме" же шаблон. Ал объекттин касиеттерин (атрибуттарын) жана жүрүм-турумун (методдорун) аныктайт.

#### **Модуль (Module)**

Python коду жазылган өзүнчө файл (.py). Модулдар кодду логикалык бөлүктөргө бөлүүгө жана башка программаларда кайра колдонууга мүмкүндүк берет.

#### **Китепкана (Library)**

Белгилүү бир маселелерди чечүү үчүн даярдалган модулдардын жана функциялардын жыйындысы (Мисалы: math - математика үчүн, requests - интернет менен иштөө үчүн).

#### **Pip (Пакет менеджер)**

Пайтон үчүн сырткы китепканаларды жана пакеттерди орнотуу, жаңылоо жана башкаруу

үчүн колдонулуучу стандарттык система.

### **Каталарды иштетүү (Exceptions Handling)**

Программа аткарылып жатканда пайда болгон каталарды (өзгөчө кырдаалдарды) кармоо жана программаны токтотуп койбоо механизми (try, except блоктору).

### **Синтаксис (Syntax)**

Программалоо тилинде код жазуунун катуу белгиленген эрежелеринин жыйындысы. Пайтондо коддун блокторун белгилөө үчүн чегинүүлөр (indentation) маанилүү роль ойнойт.

### **Виртуалдык чөйрө (Virtual Environment)**

Ар бир долбоор үчүн өзүнчө, изоляцияланган чөйрө түзүү куралы. Ал китепканалардын версияларынын ортосундагы карама-каршылыктарды болтурбоо үчүн керек.

### **Лямбда-функция (Lambda)**

Бир сап менен жазылган анонимдүү (аты жок) кыска функция.

### **Дебаггинг (Debugging)**

Программадагы каталарды (багтарды) издөө, анализдөө жана аларды оңдоо процесси.