DOI: https://doi.org/10.69722/1694-8211-2024-58-147-153
УДК: 004.43(07)

Акназар уулу К., *ага окутуучу*
kadyrbek_6621@mail.ru
ORCID: 0009-0006-5739-1323
Азизбек кызы Н., *ага окутуучу*
nus0108@mail.ru
ORCID: 0009-0006-9509-9569
Кошукулов М. А., *окутуучу*
koshukulov09@gmail.com
ORCID: 0009-0009-6820-6738
Б. Сыдыков ат. Кыргыз-Өзбек эл аралык университети
Ош ш., Кыргызстан

ОРТО МЕКТЕПТЕРДЕ PYTHON ПРОГРАММАЛОО ТИЛИНИН СТАНДАРТТЫК ФУНКЦИЯЛАРЫНЫН ӨЗГӨЧӨЛҮКТӨРҮ ЖАНА ОКУТУУ УСУЛУ

Заманбап маалымат технологиялары кеңири өнүккөн учурда жалпы билим берүүчү орто мектептердин окуучуларына программалоо тилдерин үйрөтүүгө өзгөчө ыкма менен мамиле кылуу, алардын жаш өзгөчөлүгүнө жараша тапшырма берүүнүн жөнөкөйлөштүрүлгөн ыкмаларын иштеп чыгуу актуалдуу маселелердин бири болуп саналат. Макалда Python программалоо тили менен башка программалоо тилдери салыштырылган. Python программалоо тилинин артыкчылыктары белгиленген. Бул макала окуучулардын өзүн-өзү программалоо билимин, билгичтиктерин жана көндүмдөрүн өнүктүрөт, программалоого болгон кызыгуусун арттырып, келечекте программалоо жаатында өз ордун табууга жана ошондой эле адамзат коому үчүн мыкты программаларды түзүүгө жардам берет. Азыркы учурда маалыматташтыруу адам ишмердүүлүгүнүн бардык тармактарын камтыйт жана бул көрүнүш ар бир адамдан жогорку маалыматтык маданиятты талап кылат. Демек, мектептин табият таануу программасын окуучулардын билимин, билгичтиктерин жана көндүмдөрүн өнүктүрүүнү камсыз кылган предметтер менен толуктоо зарыл.

Түйүндүү сөздөр: *программа, программалоо, билим берүү ыкмалары, функциялар, рекурсивдүү функциялар, процедуралар, компонент.*

***Акназар уулу К.**, ст. преподаватель*
kadyrbek_6621@mail.ru
ORCID: 0009-0006-5739-1323
***Азизбек кызы Н.**, ст. преподаватель*
nus0108@mail.ru
ORCID: 0009-0006-9509-9569
***Кошукулов М. А.**, преподаватель*
koshukulov09@gmail.com
ORCID: 0009-0009-6820-6738
Кыргызско-Узбекский международный университет им. Б. Сыдыкова
г. Ош, Кыргызстан

ОСОБЕННОСТИ СТАНДАРТНЫХ ФУНКЦИЙ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON В СРЕДНИХ ШКОЛАХ И МЕТОДЫ ОБУЧЕНИЯ

В условиях широкого развития современных информационных технологий одной из актуальных проблем является особый подход к обучению языкам программирования учащихся общеобразовательных средних школ, разработка упрощенных методов постановки задач в соответствии с их возрастными особенностями. В статье сравнивается язык программирования Python с другими языками программирования. Определены преимущества программирования на Python. Эта статья развивает у учащихся знания, умения и навыки самостоятельного программирования, развивает их интерес к программированию и помогает им найти свое место в сфере программирования в будущем, а также создавать лучшие программы для человечества. В настоящее время информатизация охватывает все сферы человеческой деятельности, и это явление требует от каждого высокой информационной культуры. Поэтому необходимо дополнить естественнонаучную программу школы предметами, обеспечивающими развитие знаний, знаний и умений учащихся.

Ключевые слова: *программа, программирование, образовательные методы, стандартные функции, рекурсивные функции, процедуры, компонент.*

***Aknazar uulu K.**, senior lecturer*
kadyrbek_6621@mail.ru
ORCID: 0009-0006-5739-1323
***Azizbek kyzy H.**, senior lecturer, nus0108@mail.ru*
ORCID: 0009-0006-9509-9569
***Koshukulov M. A.**, teacher, koshukulov09@gmail.com*
ORCID: 0009-0009-6820-6738
Kyrgyz-Uzbek International university
named after B.Sydykov, Osh, Kyrgyzstan

FEATURES OF STANDARD PYTHON PROGRAMMING LANGUAGE FUNCTIONS IN SECONDARY SCHOOLS AND TEACHING METHODS

In the context of the widespread development of modern information technologies, one of the urgent problems is a special approach to teaching programming languages to students of secondary schools, the development of simplified methods of setting tasks in accordance with their age characteristics. The article compares the Python programming language with other programming languages. The advantages of Python programming are defined. This article develops students' knowledge, skills and independent programming skills, develops their interest in programming and helps them find their place in the field of programming in the future, as well as create the best programs for humanity. At present, informatization covers all spheres of human activity, and this phenomenon requires a high information culture from everyone. Therefore, it is necessary to supplement the natural science program of the school with subjects that ensure the development of knowledge, knowledge and skills of students.

***Keywords:** program, programming, educational methods, standard functions, recursive functions, procedures, component.*

Күн санап өсүп жаткан маалыматтык билим берүү чөйрөсүнүн шартында окуучуларды өз алдынча иштөөгө, ар түрдүү тармактарда заманбап маалыматтык технологияларды эффективдүү колдонууга, маалымат агымын сарамжалдуу пайдаланууга үйрөтүү – билим берүүнүн бүгүнкү милдети. Ал үчүн студенттердин үзгүлтүксүз өз алдынча иштөөсүнө мүмкүнчүлүктөрдү жана шарттарды түзүү жана аларды чыгармачылык менен ойлонууга, өз алдынча чечим кабыл алууга үйрөтүү билим сапатынын жогорулашына алып келет.

Информатика азыркы учурда мектеп дисциплиналары арасында да, жогорку окуу жайларында жана орто кесиптик билим берүү мекемелеринде окуган предметтердин арасында да мазмуну боюнча эң өзгөрүп турган окуу дисциплинасы болуп саналат. Белгилей кетсек, мындан 3-4 жыл мурда мектептерде Pascal же Basic программалоо тилдери окулуп келди, ал эми акыркы үч жылдан бери орто мектептерде толугу менен Python тилин үйрөтүүгө өтүштү. Python тилин үйрөнүү зарылчылыгы барбы деген суроо жаралат.

Албетте, мектеп окуучуларынын Python тилин биринчи программалоо тили катары үйрөнүшү кээ бир табигый кооптонууну жаратышы мүмкүн: булар, биринчи кезекте, динамикалык терүүнү жана тилдин жогорку деңгээлин камтыйт. Мисалы, «массив» түшүнүгүн жогорку деңгээлдеги тизме менен алмаштыруу мектеп окуучуларына массивдин ички түзүлүшүнүн принциптерин талдоо үчүн толук мүмкүнчүлүк бербейт.

Бирок мектептеги биринчи программалоо тили катары Python*ду* үйрөнүүнүн артыкчылыктары да талашсыз. Python программалары Паскалга караганда кыйла кыскараак, бул башталгыч программисттер үчүн тил менен таанышуу милдетин бир топ жеңилдетет, анткени каталарды табуу жана мүчүлүштүктөрдү оңдоо бир топ азыраак

убакытты талап кылат. Мисалы, Pascal жана Python программалоо тилдеринде жазылган программалык коддун эки элементин салыштырып көрөлү:

a=[1]*1000 жана

var a: array [1..1000] jf integer;

...

for i:=1 to 1000 do

a[i]:=1;

Берилген коддун негизинде эки программалоо тилинде эквиваленттүү операциялар жазылганын көрүүгө болот, анын натыйжасында биз бирдиктер менен толтурулган 1000 элементтен турган массивди алабыз. Бирок бул код Python*до* 1 сапты ээлейт, ал эми Pascal*да* 3 сапты ээлейт.

Балким, методологиялык көз караштан алганда, бул бөлүмдү изилдеп, жогорудагы маселени чечүүдө окуучуга массив бөлүнгөн эс тутумдун үзгүлтүксүз фрагменти экенин түшүндүрүү керек жана аны түзүүдө анын эс тутумунда орун ээлеп, аны жарыялоо, анан инициализациялоо керектигин айтуу керек. Бирок a=[1]*1000 сызыгы, биздин оюбузча, мектеп окуучусу аткарган иш-аракеттин маанисин (1000 жолу кайталанган 1 массив керек) толугураак жана оңой чагылдырат.

7-9-класстарда информатиканы окутуу маселелерин карап чыгууда, бул программалоо жөнүндө алгачкы идеялар, балким, бала массивдер, аларды жарыялоо жана иштетүү жөнүндө жалпы түшүнүккө ээ болгондон кийин токтоло турган башталышы болушу мүмкүн экенин белгилей алабыз.

Окуунун кийинки этабында, адистештирилген класстарда мугалим менен студенттин карамагында программалык камсыздоону иштеп чыгуу үчүн иш жүзүндө колдонулган кыйла универсалдуу, заманбап программалоо тили болот.

Албетте, көп сандагы жогорку деңгээлдеги подпрограммалар камтылган тил жана тилдин кеңири функционалы окуучунун алдында алгоритмдерди жана бул функционалдык элементтердин иштөө принциптерин реалдуу изилдөөнүн ордуна бул мүмкүнчүлүктөрдү пайдалануу азгырыгына алып келет. Бирок бул жерде мектеп окуучулары адегенде жеке функционалдык элементтин иштөө принциптерин жана алгоритмдерин изилдеп, андан кийин гана маселелерди чечүүдө аны колдонууга өтүшү керек экендигине байланыштуу мындай программалоо тилин үйрөнүүнүн методологиялык өзгөчөлүгүн карап чыгуу зарыл. Мисалы, төмөнкү маселени карайлы: *a өзгөрмөсүнүн маанилерин b өзгөрмөсүнө жазыңыз*. Маселе адегенде кошумча c өзгөрмөсүн колдонуу менен чечилиши керек, ага биз кайра жазылуучу a өзгөрмөнүн маанисин жазабыз:

c=a

a=b

b=c

жана ошондон кийин гана Python тилинин кортеждерин колдонууга болот:

(a,b)=(b,a).

Ушундай жол менен мугалим массивдерди сорттоо жана элементтерди издөө менен байланышкан көпчүлүк маселелерди чечүүгө жакындай алат, бул акырында окуунун адистештирилген этабында студентке кыска убакыттын ичинде көп сандагы ар кандай маселелерди чечүүгө мүмкүндүк берет.

Бүгүнкү күндө бир нече программалоо тилдери иштелип чыккан, мисалы, Pascal, Delphi, C++, C#, Python, Java жана башкалар. Бул программалоо тилдери көйгөйлөрдү белгилүү бир багытта чечүү үчүн иштелип чыккан жана аларды объектиге багытталган

программалоо тилдери деп атоого болот. Программалоо тилдери жөнүндө сөз кылуудан мурун “программа”, “программалоо”, “программалоо чөйрөсү” түшүнүктөрүнүн маңызына токтолуп кетели.

Программа – бул алдын ала даярдалган алгоритмдердин жана компьютердик каражаттардын негизинде аткарылуучу иштин шарттарын, максатын жана милдетин эске алуу менен формалдуу (шарттуу) алгоритмдик тилдин так жана толук туюндурулушу [2; 46-б.].

Программа – бул иштин планы же иш-чара. Программа – маселени чечүүдө компьютер аткара турган иш-аракеттердин иреттүү ырааттуулугу [4; 17-б.].

Программалоо тили – компьютерлер үчүн тигил же бул аракетти аткарууга мажбурлоочу программалар (инструкциялардын жыйындысы) жазылган расмий тил [4; 53-б.].

Биз программалоо дегенди компьютерлер жана башка микропроцессордук электрондук машиналар үчүн программаларды түзүү, сыноо жана мүчүлүштүктөрдү оңдоо процесси деп эсептейбиз. Башкача айтканда, компьютерге программа түзүү процесси программалоо, ал эми программаны түзгөн адам программист деп аталат. Компьютер түшүнгөн тил программалоо тили деп аталат.

Программалоо чөйрөсү деп программист коддорду жаза турган тил жана чөйрө түшүнүлөт. Алсак, кеңири таралган жана көп колдонуучу чөйрөлөргө мисал катары төмөнкүлөрдү көрсөтүүгө болот: PHPStorm – негизинен, PHP иштеп чыгуучулары үчүн, VisualStudio – .Net иштеп чыгуучулары үчүн, NetBeans – Java, PHP иштеп чыгуучулары үчүн, PHPDesigner – негизинен, веб (PHP) иштеп чыгуучулары үчүн. Бүгүнкү күндө программалоо жогорку деңгээлдеги программалоо тилдерин (Delphi, Java, C++, S#, Python) колдонуу менен жүргүзүлөт. Бул программалоо тилдеринин семантикасы адам тилине жакын, программалоо процессин жеңилдетет.

Python программалоо тилин үйрөнүүдө төмөнкү негизги түшүнүктөр маанилүү. **Функция** – бул программанын белгилүү бир тапшырманы аткарган, аталышы бар, бир же бир нече маанилерди кабыл алган жана өз ишин аяктагандан кийин негизги программага бир же бир нече натыйжа маанисин кайтарган бөлүгү.

Бөлүк программасы зарыл болгон учурда чакырылат. Ал бир эле кодду бир нече жолу жазуу зарылдыгын жок кылып, программанын ичинде бир нече жолу колдонулушу мүмкүн. Бул коддун бөгөттөлүшүн жогорулатат, аны түшүнүүнү жеңилдетет жана каталарды табууга жардам берет. Коддун бир блогунда каталарды текшерсе болот. Эгерде ката программанын бөлүгүндө болсо, анда программанын бир бөлүгүн гана оңдоо керек. Эгерде код бөлүк программасын колдонбостон, бир нече жерде кайра-кайра жазылса, анда программа боюнча каталарды издөөгө туура келет.

• код бир гана жолу жаңыланышы керек: бардык өзгөртүүлөр программанын бөлүгү чакырылгандан кийин күчүнө кирет.

• процедура – программанын функцияга окшош көп жолу колдонуучу бөлүгү, бирок ал эч кандай маани кайтарбайт.

Python программалоо тили ар кандай маселелерди чечүү үчүн иштелип чыккан бир нече пайдалуу стандарттык функцияларга ээ. Ар бир функцияга жана процедурага ат коюлушу керек жана бул ат Python тилиндеги define (define – аныктоо) сөзүнөн алынган def ачкыч сөзүнөн башталат.

def – функцияны жарыялаган ачкыч сөз.

function_name – функциянын аты.

параметр тизмеси – бул тизме бир нече параметрлерден турушу мүмкүн жана алар үтүр менен ажыратылат.

буйрук_блок – функциянын денеси башка операторлор сыяктуу бир тамга менен жазылышы керек.

Функция аты менен чакырылганда, анда камтылган буйруктардын ырааттуулугу аткарылат. Андан кийин программа функция чакырылган сапка кайтып келип, ошол саптагы кийинки буйруктарга өтөт.

Python тилинде процедуралар дээрлик функциялар сыяктуу жазылат. Айырмачылыгы: процедуралар эч кандай маани бербейт.

Өзүн өзү чакырган функция рекурсия деп аталат, ал эми мындай функциялар рекурсивдүү функциялар деп аталат.

Рекурсивдүү функциялар – күчтүү программалоо механизми, бирок алар дайыма эле эффективдүү боло бербейт. Анткени көпчүлүк учурда ката кетирет. Эң кеңири таралган ката – чексиз рекурсия. Анда функцияларды чакыруу чынжырчасы чексиз жана компьютердин бош эс тутуму түгөнгөнгө чейин уланат. Чексиз рекурсиянын пайда болушунун себептери:

- рекурсияда шарттын туура эмес колдонулушу. Мисалы, факториалды эсептөөдө if n==0 экенин унутуп калсак, factorial(0) функциясы factorial(–1)ди, factorial(–1) функция factorial(–2) жана башкаларды чакырат;

- туура эмес параметр менен рекурсивдүү функцияны чакыруу. Мисалы, factorial(n) функциясы factorial(n) ди чакырса, кайрадан чексиз чынжыр пайда болот. Ошондуктан рекурсивдүү функцияны түзүүдө рекурсияны аяктоо шартын жана рекурсияны качан жана кантип бүтүрүүнү ойлонуу керек.

Төмөндө биз Python программалоо тилиндеги функцияларды жана процедураларды колдонгон программаларды карап чыгабыз. Биринчи программабыздын шарты төмөнкүдөй: *берилген эки сандын эң кичине жалпы бөлүнүүчүсүн табуу программасын жазгыла.*

def ЭКЖБ(a, b) функцияны колдонуңуз:

```
n = a * b
while a!= 0 and b!= 0:
if a > b:
a%= b else:
b %= a
return n // (a + b)
a = int(input('a = '))
b = int(input('b = '))
print('ЭКЖБ:',экжб(a, b))
```

Экинчи программабыздын шарты төмөнкүдөй: *берилген n санын рим цифралары менен туюнтуучу программа жазгыла. Функцияны колдонуңуз.*

```
a = [(1000, 'M'), (900, 'CM'), (500, 'D'), (400, 'CD'), (100, 'C'), (90, 'XC'),
(50, 'L'), (40, 'XL'), (10, 'X'), (9, 'IX'), (5, 'V'), (4, 'IV'), (1, 'I')]
def rim(n):
s = ''
while n > 0:
```

```
for i, r in a:
while n >= i:
s += r
n -= i
return s
n=int(input("Сан киргизиңиз=")) print(rim(n))
```

Үчүнчү программанын шарты: *Берилген төрт сандын эң кичинесин таба турган программа жазгыла. Бул үчүн min4(a, b, c, d) функциясын түзүңүз.*

```
def min(a,b):
if a>b:
return b else:
return a
def min4(a,b,c,d):
return min(min(min(a,b),c),d) a=int(input('a='))
b=int(input('b='))
c=int(input('c='))
d=int(input('d='))
print('Эң кичине сан=', min4(a,b,c,d))
```

Python программалоо тили – объектиге багытталган программалоо тили, аны түшүнүүгө жана үйрөнүүгө оңой. Python программалоо тилинин артыкчылыктары төмөндөгүчө:

- бул программалоо тилинде түзүлгөн программа башка платформаларда жана операциялык системаларда иштейт;
- бир платформада же операциялык системада жазылган код башка платформага же операциялык системага көчүрүлгөндө, башка эс тутумду ээлей алат. Бул кээ бир каталарга алып келиши мүмкүн. Мындай программаларга мисал болуп Си программалоо тили саналат. Ал эми Python программалоо тилинде мындай кемчилик жоюлган.

Жалпы орто мектепте заманбап программалоо тилдерин окутуу студенттерге программалоо чөйрөсүндө иштөөгө жана өз алдынча программаларды түзүүгө, өлкөбүздүн ар кайсы тармактары үчүн тиркемелерди түзүүгө, заманбап маалыматтык технология каражаттарын колдонууга жана алардын практикалык иштеринде жаңы аспектилерин ачууга шарт түзөт.

Адабияттар:

1. Уэйн Круглер. Python: Подробное руководство. 2023.

2. Иванов, А. А. Python как универсальный язык программирования [Текст] / А. А. Иванов // Вестник Томского государственного университета. Информатика и кибернетика. - 2023. - № 2. - С. 32-41.

3. Петров, В. В. Python для начинающих: синтаксис, типизация, объектно-ориентированность [Текст] / В. В. Петров // Компьютерные сети и телекоммуникации. - 2022. - № 3. - С. 14-25.

4. Эндрю Уильямс. Python для машинного обучения. 2021.