

Дәріс №10. Объектілік бағдарламалау

Жоспары:

1. Python тілінде класс
2. Жасау көшірмелер нысандар

Python тілінде класс - объектіні сипаттайтын қасиеттер жиынтығын сипаттайтын тип. Әрбір сыныпта объектіні сипаттайтын функциялар мен айнымалы мәндерді қамтитын деректер құрылымы бар. Класс мүшелері әдістер деп аталатын функциялар, сондай-ақ атрибуттар деп аталатын айнымалылар (сынып құрылымында жарияланған) болуы мүмкін. Сыныптың мүшелеріне нүктелік белгілерді қолданатын бағдарламада сынып атауынан кейін сәйкес жұрнақ қою арқылы келесі синтаксисті пайдалана отырып, сілтеме жасауға болады: `class-name.method-name()` немесе `class-name.attribute-name`. Класс туралы мәлімдеме сыныптың кілт сөзінен басталады, одан кейін бағдарламашы көрсеткен сынып атауы (атын таңдаған кезде әдеттегі Python атау конвенциясын орындаңыз, бірақ бас әріптерден бастаңыз), содан кейін қос нүкте таңбасы. Әрі қарай шегіністерге класс құжат тізбегін, айнымалылар декларациясын — сынып атрибуттарын және класс әдістерін анықтауды анықтайтын нұсқаулар беріледі. Осылайша, класс блогының синтаксисі келесідей: Класс СыныпАты: `""" сынып-docstring """` класс-айнымалы-жариялау класс-әдіс-анықтама Оның атрибуттары мен әдістерін анықтайтын сынып декларациясы үлгі болып табылады. сыныптың жұмыс көшірмелері (даналары) шығарылуы мүмкін. Әдіс анықтамасында жарияланған барлық айнымалылар даналық айнымалылар ретінде белгілі Және тек жергілікті жерде қол жетімді әдіс, в олар қандай болды жариялады - дейін ол тыйым салынған байланыс сырттан сыныптық құрылымдар. Әдетте, дананың айнымалы мәндері сынып данасы жасалған кезде қоңырау шалушы жіберген деректерді қамтиды. Бұл деректер тек жергілікті түрде қолжетімді болғандықтан («ішкі пайдалану үшін» дейді) ол бағдарламаның қалған бөлігінен тиімді түрде жасырылады. Деректерді инкапсуляциялау деп аталатын бұл әдіс деректердің сынып құрылымында қауіпсіз сақталуын қамтамасыз етеді және объектіге бағытталған бағдарламалаудың (ООР) бірінші принципі болып табылады. Сыныптың барлық қасиеттеріне `self` префиксі бар нүкте белгісін қолдану арқылы жергілікті түрде қол жеткізуге болады, мысалы, `sound` деп аталатын атрибутты былай деп атауға болады: `self.sound`. Оған қоса, барлық әдіс анықтамаларында бірінші аргумент ретінде `self`, яғни аталған әдіс болуы керек сөйлесу әңгімеге ұқсайды (өзіндік). Дана жасау кезінде сынып автоматты түрде шақырды арнайы әдіс `__ішінде__` (өзіндік). Оның атрибуттарын инициализациялау үшін көбірек мәндерді беру қажет болса, жақшаға келесі аргументтер қосылуы мүмкін. Python тіліндегі толық сынып декларациясы келесідей болуы мүмкін: `class Critter :` `""" Негізгі класс барлығына Тірі тіршілік иелері. """ санау = 0` `Def __ішінде__ ( өзін , чат ) :` `self.sound = chat Critter.count += 1` `def talk( self ) :` `return self.sound` Осы мысалда берілген класс компоненттерін қарастырайық.

- Санау айнымалысы бүтін мәні сол сыныптың барлық даналары үшін қолжетімді класс айнымалысы болып табылады. Оған сынып ішінде және одан тыс `Critter.count` жазбасы арқылы қол жеткізуге болады.

- Әдістердің біріншісі `__init__()` класс данасы жасалған кезде автоматты түрде шақырылатын инициализация әдісі.

- Әдіс `__init__()` бұл жағдайда дыбыс данасы айнымалы мәнін сөйлесу аргументінен берілген мәнмен инициализациялайды және сол сыныптың данасы жасалған сайын санау класының айнымалы мәнін арттырады.

- Екінші `talk()` әдісі қалыпты функция ретінде жарияланады, тек ол автоматты түрде белгіленеді аргумент өзін - басқалар құндылықтар Үшін шақырылған оператордан аудару қажет емес.

- Бұл жағдайда `talk()` әдісі дыбыс данасы айнымалысында инкапсуляцияланған мәнді қайтарады

Жасау көшірмелер нысандар

сынып данасы, объект болу болып табылады Жай көшіру прототип, класс конструкторын шақыру арқылы жасалған, шақыру кезінде жақшаның ішіндегі дәлелдердің қажетті санын көрсете отырып. Бұл шақырудың аргументтері әдіс анықтамасында көрсетілгендерге сәйкес болуы керек `__init__()`, ішкі аргументтің өзін қоспағанда.

Сынып данасы конструктор қайтарған нысан және синтаксисті пайдаланып айнымалыға тағайындалуы мүмкін

дананың аты = Сынып атауы(аргументтер).

Құрылғанның әдістері мен атрибуттарына қол жеткізу үшін `мысалы`, дана атауы сияқты нүкте белгілерін пайдалануға болады .

әдіс атауы() немесе `instance-name.attribute-name` .

Әдетте, кейбір базалық класс Python модулінің бөлек файлында анықталады және сақталады. Сіз оны кейінірек басқа бағдарламаларға импорттай аласыз және осылайша сыныптың негізгі прототипі деп аталатын даналық нысандарды оңай жасай аласыз.

1. Жұмысты бастаңыз жаңа бағдарламасы қосулы Python бірге жарнамалар жаңа сипаттаушы құжат жолын қамтитын сынып.

сынып Құс :

"Негіз Сынып, анықтау қасиеттері құстар."

2. Содан кейін бүтін мәні тең класс төлсипатының айнымалы мәнін жариялайтын және инициализациялайтын мәлімдемені қосыңыз

нөл. санау

= 0

3. Қазір анықтау әдіс сынып, инициализациялау даналық айнымалы және сынып айнымалысының мәнін арттырады.

`def __init__( өзін , чат ) :`

`self.sound = чат`

`Bird.count += 1`

4. Соңында, шақырылғанда айнымалы дананың мәнін қайтаратын сынып әдісін қосыңыз. Содан кейін сыныпты бөлек файлға сақтаңыз.

```
Def әңгіме( өзін ):
```

```
    қайтару өздік.дыбыс
```

5. Импорттау арқылы сыныптың сипаттарын қолжетімді ету арқылы басқа Python бағдарламасын бастаңыз, содан кейін сол сынып үшін құжат жолын шығарыңыз.

```
    бастап Құс импорты *
```

```
    басып шығару( '\nСынып Даналар Оның:\n' , Құс.__док__ )
```

6. Енді сынып данасын жасау үшін мәлімдемені қосыңыз және айнымалы данаға жол аргументінің мәнін беріңіз.

```
    полли = Құс( 'Сквок, шықыр!' » )
```

7. Содан кейін айнымалы дананың мәнін басып шығарыңыз және сынып айнымалысының мәнін көрсету үшін сынып әдісін шақырыңыз.

```
    басып шығару( '\nНөмір Оның Құстар:'
```

```
    , polly.count ) print( 'Полли
```

```
    былай дейді:' , polly.talk() )
```

8. Жасау екінші көшіру сынып, өту басқа мағынасы даналық айнымалыға жол аргументі.

```
    Гарри = Құс( 'Твит, твит!' » )
```

9. Ақырында көрсету мағынасы айнымалы көшіру Және қоңырау шалу класс айнымалысының мәнін көрсету үшін класс әдісі.

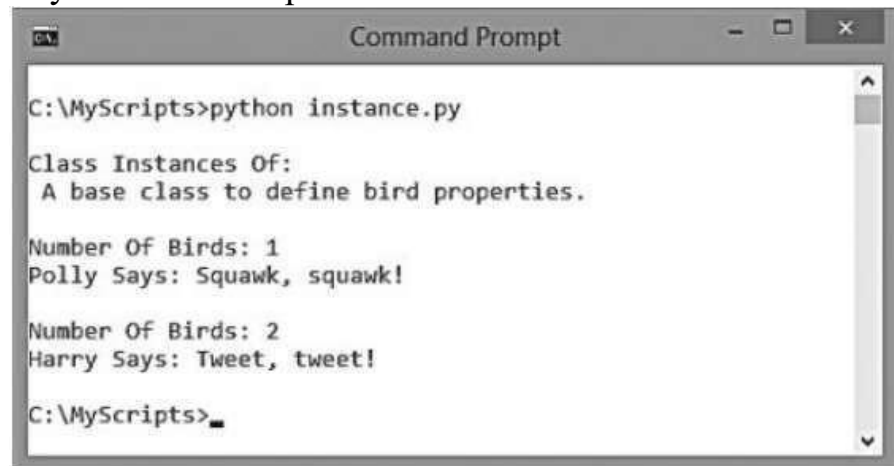
```
    басып шығару( '\nНөмір Оның
```

```
    Құстар:' , harry.count ) басып
```

```
    шығару( 'Гарри былай дейді:' ,
```

```
    harry.talk() )
```

10. Файлды жұмыс каталогында сақтаңыз, пәрмен жолын ашыңыз және бағдарламаны іске қосыңыз - сіз Bird класының екі данасын жасау нәтижесін көресіз.



```
Command Prompt
C:\MyScripts>python instance.py

Class Instances Of:
  A base class to define bird properties.

Number Of Birds: 1
Polly Says: Squawk, squawk!

Number Of Birds: 2
Harry Says: Tweet, tweet!

C:\MyScripts>
```