

**KONSTRUKTUR. OBYEKT ATRIBUTINI HOSIL QILISH**

Tojimatov Israiljon Nurmamatovich

Farg'ona davlat universiteti

israiltojimatov@gmail.com

O'ktamjonova Nilufar Abdurahmon qizi

Farg'ona davlat universiteti

karimberdiyevanilufar625@gmail.com

Annotatsiya. Ushbu maqolada Python dasturlash tilida obyektga yo'naltirilgan dasturlashning muhim elementi bo'lgan konstruktor va obyekt atributlari tushunchasi keng yoritilgan. Konstruktor yordamida klassdan obyekt yaratilganda avtomatik tarzda atributlarga qiymat biriktirish, atributlar orqali obyektning individual holatini shakllantirish va ular ustida amallar bajarish jarayonlari nazariy va amaliy jihatdan tahlil qilinadi. Shuningdek, atributlarning turlari, ularning o'zgaruvchanligi, kod tuzilmasida tutgan o'rni hamda konstruktor orqali dastur modulliligini ta'minlashdagi roli izohlangan. Maqola o'quvchilarga obyektga yo'naltirilgan dasturlashning asosiy tamoyillarini tushunishga yordam beradi va Python tilida samarali kod yozish ko'nikmasini shakllantirishga xizmat qiladi.

Аннотация. В данной статье подробно рассматриваются понятия конструктора и атрибутов объекта как важнейшие элементы объектно-ориентированного программирования на языке Python. Раскрывается механизм автоматического присваивания значений атрибутам при создании объекта с помощью конструктора, а также их роль в формировании индивидуального состояния объекта и выполнении над ним операций. Освещаются типы атрибутов, их изменяемость, значение в структуре кода и роль конструктора в обеспечении модульности программы. Статья помогает читателям лучше понять основные принципы объектно-ориентированного



программирования и развить навыки эффективного написания кода на языке Python.

Abstract. This article provides a detailed overview of constructors and object attributes as fundamental components of object-oriented programming in the Python language. It explains how constructors automatically assign values to attributes during object creation, how these attributes define the state of an object, and how they are used in various operations. The paper also discusses types of attributes, their mutability, their role in code structure, and how constructors contribute to program modularity. This article helps readers gain a clearer understanding of core object-oriented programming principles and supports the development of effective Python coding skills.

Kalit so'zlar. Konstruktor, obyekt atributi, klass, metod, obyekt, dasturlash, python, atribut yaratish, obyekt yaratish, obyekt holati, konstruktor vazifasi, sinf tuzilmasi, funksional dasturlash, inkapsulyatsiya, obyektga yo'naltirilgan dasturlash, moslashtirish, kod strukturasi, avtomatlashtirish, atribut qiymati, klass ichidagi metod, obyekt ustida amallar, atributlarni o'zgartirish, xususiy atributlar, umumiy atributlar, dastur modulliligi, sinfga xos xususiyat, ma'lumotlar saqlash, obyekt farqlari, parametr uzatish.

Ключевые слова. Конструктор, атрибут объекта, класс, метод, объект, программирование, python, создание атрибутов, создание объекта, состояние объекта, роль конструктора, структура класса, функциональное программирование, инкапсуляция, объектно-ориентированное программирование, настройка, структура кода, автоматизация, значение атрибута, методы в классе, действия над объектом, изменение атрибутов, приватные атрибуты, общие атрибуты, модульность программы, особенности класса, хранение данных, различие объектов, передача параметров.

Keywords. Constructor, object attribute, class, method, object, programming, python, attribute creation, object creation, object state, constructor role, class structure, functional programming, encapsulation, object-oriented programming,



customization, code structure, automation, attribute value, methods in class, object manipulation, attribute modification, private attributes, shared attributes, program modularity, class-specific features, data storage, object differences, parameter passing.

Kirish. Hozirgi kunda axborot texnologiyalari hayotimizning deyarli barcha jabhalariga chuqur kirib bormoqda. Zamonaviy jamiyatda turli sohalar — ta’lim, tibbiyot, sanoat, moliya, biznes, aloqa, hatto san’at ham — raqamlashtirilmoqda va avtomatlashtirilmoqda. Bu esa, o‘z navbatida, zamonaviy va qulay dasturlash tillariga bo‘lgan talabni keskin oshiradi. Ana shunday tillardan biri bu — Python dasturlash tilidir.

Python — bu yuqori darajadagi, umumiy maqsadli, soddaligi bilan mashhur bo‘lgan dasturlash tili bo‘lib, u dastlab 1989-yilda Niderlandiyalik dasturchi Guido van Rossum tomonidan ishlab chiqilgan va 1991-yilda ommaga taqdim etilgan. Python tilining asosiy g‘oyasi — kodni inson tomonidan o‘qilishi va tushunilishi oson bo‘lishidir. Boshqacha qilib aytganda, Python dasturchiga murakkab algoritmlarni sodda va tushunarli sintaksis yordamida ifoda etish imkonini beradi.

Pythonning o‘ziga xos jihati shundaki, u interpreter til hisoblanadi, ya’ni dasturchi yozgan kod to‘g‘ridan-to‘g‘ri bajariladi, uni alohida kompilyatsiya qilishga hojat yo‘q. Bu esa dasturlash jarayonini tezlashtiradi va soddalashtiradi. Python tili turli operatsion tizimlarda (Windows, Linux, MacOS) ishlaydi va platformadan mustaqil bo‘lib, undan istalgan qurilmada foydalanish mumkin.

Bugungi kunda Python nafaqat tajribali dasturchilar, balki yangi o‘rganayotgan talabalar orasida ham keng tarqalgan. Bu til yordamida quyidagi yo‘nalishlarda samarali dasturlar tuzish mumkin:

Veb ilovalar (Django, Flask),

Ma’lumotlar tahlili (NumPy, Pandas),

Sun’iy intellekt va mashinali o‘rganish (TensorFlow, Scikit-learn),

Avtomatlashtirish va skript yozish (Selenium),

O‘yinlar yaratish (Pygame),

Grafik interfeyslar yaratish (Tkinter, PyQt).



Shuningdek, Python tili keng va faol dasturchilar hamjamiyatiga ega bo'lib, muntazam ravishda yangilanib, yangi kutubxonalar, vositalar va imkoniyatlar bilan boyitib borilmoqda. Bu esa uni boshqa dasturlash tillaridan ajratib turadi.

Konstruktor tushunchasi

Python dasturlash tilida konstruktor – bu sinfdan (klassdan) obyekt yaratilganda avtomatik ishga tushadigan maxsus metoddir. Konstruktor yordamida obyektga kerakli boshlang'ich atributlar beriladi va ularning qiymatlari aniqlanadi. Konstruktor init nomi bilan belgilanadi va klass ichida e'lon qilinadi.

```
class Talaba:  
    def init(self, ism, yosh):  
        self.ism = ism  
        self.yosh = yosh
```

Bu yerda init konstruktor metodi hisoblanadi. self parametri obyektning o'zini bildiradi, ism va yosh esa obyektga uzatiladigan tashqi qiymatlardir.

Obyekt atributlarini hosil qilish

Konstruktor orqali obyektning atributlari (xususiyatlari) yaratiladi. Bu atributlar obyektga tegishli bo'lgan ma'lumotlarni saqlaydi. Har bir obyekt uchun ushbu atributlar alohida qiymatga ega bo'ladi.

```
# Obyekt yaratish  
t1 = Talaba("Ali", 21)  
t2 = Talaba("Gulbahor", 19)  
print(t1.ism) # Natija: Ali  
print(t2.yosh) # Natija: 19
```

Bu yerda t1 va t2 nomli ikki obyekt yaratildi. Ular o'ziga xos ism va yosh atributlariga ega bo'ldi.

Konstruktorning afzalliklari

Kodning soddaligi va strukturasi: Konstruktor obyekt yaratilishi bilan birga atributlarni aniqlash imkonini beradi.

Avtomatlashtirish: Har safar obyekt yaratishda ma'lumotlar avtomatik tayinlanadi.



Modullik: Har bir klass o'z atributlari va konstruktoriga ega bo'ladi.

Moslashtirish: Turli parametrlar orqali har xil obyektlar yaratiladi.

Argumentlarsiz konstruktor (default)

Agar konstruktor hech qanday tashqi ma'lumot qabul qilmasa, unga default qiymatlar beriladi.

```
class Avtomobil:
```

```
    def init(self):
```

```
        self.rang = "Qora"
```

```
        self.model = "Malibu"
```

```
a1 = Avtomobil()
```

```
print(a1.rang) # Qora
```

```
print(a1.model) # Malibu
```

Bunday hollarda konstruktor har doim bir xil qiymatlar bilan obyektни yaratadi.

Konstruktor va metodlar bilan ishlash

Konstruktor yordamida yaratilgan atributlar klassdagi boshqa metodlarda ham ishlatiladi.

```
class Talaba:
```

```
    def init(self, ism, kurs):
```

```
        self.ism = ism
```

```
        self.kurs = kurs
```

```
    def info(self):
```

```
        print(f"Talaba: {self.ism}, {self.kurs}-kurs")
```

```
t1 = Talaba("Javohir", 3)
```

```
t1.info()
```

Natija:

Talaba: Javohir, 3-kurs

Bir nechta obyektlar bilan ishlash

Konstruktor yordamida bir klassdan istalgancha obyekt yaratish mumkin.



```
t1 = Talaba("Aziza", 2)
t2 = Talaba("Sardor", 4)
t3 = Talaba("Xurshid", 1)
t1.info()
t2.info()
t3.info()
```

Har bir obyekt o'z atributlariga ega bo'ladi va ular mustaqil ishlaydi.

Obyekt atributini hosil qilish

Python dasturlash tilida obyektga yo'naltirilgan dasturlash tamoyillariga ko'ra, har bir obyekt o'zining xususiy ma'lumotlariga ega bo'ladi. Bu ma'lumotlar obyekt atributlari deb ataladi. Obyekt atributlari — obyekt yaratilganida unga biriktiriladigan qiymatlar bo'lib, ular obyektning holatini belgilaydi. Boshqacha aytganda, atributlar – obyekt haqida ma'lumot saqlovchi belgilar hisoblanadi.

Obyekt atributlari odatda klass ichida aniqlanadi va ular obyekt yaratilganda unga tayinlanadi. Har bir obyekt o'z atributlariga ega bo'ladi va ular boshqa obyektlardan mustaqil holda ishlaydi. Masalan, "Talaba" nomli klass bo'lsa, uning atributlari "ism", "yosh", "kurs" kabi bo'lishi mumkin. Har bir talaba (obyekt) bu atributlarga individual qiymatlar ega qiladi.

Obyekt atributlarini yaratish uchun avvalo klass tuziladi, so'ngra klass ichida maxsus metod orqali atributlar aniqlanadi. Odatda bu metod konstruktor deb ataladi va u obyekt yaratilganda avtomatik ishlaydi. Konstruktor yordamida atributlarga boshlang'ich qiymatlar beriladi. Atributlarning nomlari odatda obyektga xos belgilarni bildiradi. Masalan, avtomobil obyektining atributlari "model", "rang", "tezlik" bo'lishi mumkin.

Python dasturlash tilida atributlar ikki xil bo'ladi:

1. Obyekt atributlari – faqat bitta obyektga tegishli bo'lgan xususiyatlar.
2. Klass atributlari – barcha obyektlar uchun umumiy bo'lgan xususiyatlar.

Obyekt atributlari o'zgaruvchan bo'ladi, ya'ni har bir obyekt uchun har xil qiymatlar saqlanishi mumkin. Bu esa dasturda moslashuvchanlik va aniqlikni ta'minlaydi. Obyektlar orasidagi farqlar atributlar orqali ajratiladi. Misol uchun, ikki



xil talabani tasvirlashda ular bir xil klassga tegishli bo'lsa ham, ism va yosh kabi atributlari orqali farqlanadi.

Obyekt atributlari dasturda ma'lumotlar tartibini saqlash, obyektlar holatini boshqarish, va metodlar orqali ular ustida amallar bajarishda muhim rol o'ynaydi. Dasturchilar atributlardan foydalangan holda har bir obyektga xos ma'lumotlarni aniqlaydi va ularga osongina murojaat qiladi.

Shuningdek, atributlar dasturdagi har bir obyektning to'liq ifodalashga yordam beradi. Ular orqali obyektning tanish, uni tahlil qilish, boshqa obyektlar bilan solishtirish mumkin. Bu esa dastur tuzilmasini aniq va tizimli qiladi.

Xulosa

Xulosa qilib aytganda, Python dasturlash tilida konstruktor va obyekt atributlari obyektga yo'naltirilgan dasturlashning asosiy poydevorini tashkil etadi. Konstruktor orqali obyektlar yaratish jarayoni soddalashadi va har bir obyektga o'ziga xos atributlar birlashtirish imkoniyati paydo bo'ladi. Bu atributlar obyektning holatini aniqlaydi, ularning ustida amallar bajarish esa dastur funktsionalligini ta'minlaydi.

Atributlarning to'g'ri tashkil etilishi kodni strukturaviy, tushunarli va qayta ishlashga yaroqli qiladi. Shuningdek, konstruktorlar yordamida dasturchi har bir obyekt uchun individual qiymatlar belgilash orqali ma'lumotlar yaxlitligini saqlaydi va dasturda moslashuvchanlikka erishadi.

Umuman olganda, konstruktor va obyekt atributlarini chuqur tushunish – Python tilida samarali va professional dasturlar yaratish uchun zarur bilimlardan biridir. Bu mavzuni o'zlashtirish orqali o'quvchilar va dasturchilar obyektga yo'naltirilgan yondashuv asosida mustahkam bilimga ega bo'ladilar.

FOYDALANILGAN ADABIYOTLAR RO'YXATI:

1. Абрамян М.Э. "Практикум по программированию на языке Python" — М.: Бином, 2021.
2. Шабалин Д. "Объектно-ориентированное программирование на Python", СПб: Питер, 2020.



3. Zelle, J. "Python Programming: An Introduction to Computer Science", Franklin, Beedle & Associates, 2016.
4. Lutz, M. "Learning Python", 5th Edition, O'Reilly Media, 2013.
5. Tadjibayeva D.S., Yusupova D.O. "Python dasturlash asoslari", Toshkent: Innovatsiya nashriyoti, 2022.
6. G'ulomov S., Qurbonov B. "Python dasturlash tili va amaliyot", Toshkent: Fan va texnologiya, 2021.
7. <https://docs.python.org/3/> — Python dasturlash tilining rasmiy hujjatlari va OOP bo'yicha ma'lumotlar (ingliz tilida).
8. <https://habr.com/ru/articles/443200/> — "Конструкторы и атрибуты объектов в Python" maqolasi (rus tilida, Habr platformasi).