

## **TALABA VA SHAKL SINFGA OID MASALALARNING ABSTRAKT SINFLAR VA ABC MODULI YORDAMIDA YECHISH.**

***Yusupov Mirsaid Abdulaziz o'g'li***

*Farg'ona davlat universiteti amaliy matematika va  
informatika kafedrası o'qituvchisi  
e-mail: mirsaidbeky@gmail.com*

***Raxmatjonova Gulshoda Raximjon qizi***

*Farg'ona davlat universiteti Fizika-matematika fakulteti  
Amaliy matematika yo'nalishi 3-kurs talabasi  
e-mail: mamasharipovelmurodjon@gmail.com*

**Annotatsiya.** Ushbu maqolada obyektga yo'naltirilgan dasturlash darsining muhim tushunchalaridan biri bo'lgan abstrakt sinflar va ularni Python dasturlash tilida amalga oshirish uchun mo'ljallangan abc (Abstract Base Classes) moduli batafsil tahlil qilinadi. Abstrakt sinflarning mohiyati, ularning dastur arxitekturasidagi o'rni hamda murakkab tizimlarni loyihalashdagi ahamiyati keng yoritib beriladi. Maqolada abc modulining asosiy komponentlari — ABC bazaviy sinfi va @abstractmethod dekoratori orqali abstrakt metodlarni yaratish mexanizmi izohlanadi. Abstrakt sinflardan foydalanish orqali kodning mantiqiy tuzilishini mustahkamlash, merosxo'rlik jarayonida majburiy metodlarni belgilash hamda dasturda xatoliklarning oldini olish imkoniyatlari ko'rsatib beriladi. Shuningdek, abstrakt sinflar va oddiy sinflar o'rtasidagi farqlar, interfeys tushunchasi bilan o'xshash va farqli jihatlar tahlil qilinadi. Nazariy tushunchalar amaliy misollar orqali mustahkamlanib, real dasturiy loyihalarda abstrakt sinflardan samarali foydalanish bo'yicha tavsiyalar beriladi.

**Kalit so'zlar.** Abstrakt sinf, abc moduli, Abstract Base Class, obyektga yo'naltirilgan dasturlash, Python dasturlash tili, abstrakt metod, @abstractmethod, merosxo'rlik (inheritance), interfeys, dastur arxitekturası, sinflar va obyektlar, kodning qayta ishlatilishi, dasturiy dizayn.

**Annotation.** This article provides a detailed analysis of abstract classes, one of the important concepts of the object-oriented programming paradigm, and the abc (Abstract Base Classes) module designed to implement them in the Python programming language. The essence of abstract classes, their role in program architecture, and their importance in designing complex systems are widely covered. The article explains the main components of the abc module — the ABC base class and the mechanism for creating abstract methods using the @abstractmethod decorator. The possibilities of strengthening the logical structure of the code, defining mandatory methods in the inheritance process, and preventing errors in the program are shown. The differences between abstract classes and simple (concrete) classes,

similarities and differences with the concept of an interface are also analyzed. Theoretical concepts are reinforced through practical examples, and recommendations are given for the effective use of abstract classes in real software projects.

**Keywords.** Abstract class, abc module, Abstract Base Class, object-oriented programming, Python programming language, abstract method, @abstractmethod, inheritance, interface, program architecture, classes and objects, code reuse, software design.

**Аннотация.** В данной статье представлен подробный анализ абстрактных классов, одной из важных концепций объектно-ориентированной парадигмы программирования, и модуля abc (Abstract Base Classes), предназначенного для их реализации в языке программирования Python. Широко освещены сущность абстрактных классов, их роль в архитектуре программ и важность при проектировании сложных систем. В статье объясняются основные компоненты модуля abc — базовый класс ABC и механизм создания абстрактных методов с использованием декоратора @abstractmethod. Показаны возможности усиления логической структуры кода, определения обязательных методов в процессе наследования и предотвращения ошибок в программе. Также анализируются различия между абстрактными классами и простыми (конкретными) классами, сходства и различия с концепцией интерфейса. Теоретические концепции подкрепляются практическими примерами, и даются рекомендации по эффективному использованию абстрактных классов в реальных программных проектах.

**Ключевые слова.** Абстрактный класс, модуль abc, абстрактный базовый класс, объектно-ориентированное программирование, язык программирования Python, абстрактный метод, @abstractmethod, наследование, интерфейс, архитектура программы, классы и объекты, повторное использование кода, проектирование программного обеспечения.

### **Kirish**

Zamonaviy dasturlashda murakkab va kengaytiriladigan tizimlarni yaratishda obyektga yo'naltirilgan dasturlash muhim rol o'ynaydi. Obyektga yo'naltirilgan dasturlash asosiy tushunchalari sinflar, obyektlar, merosxo'rlik, polimorfizm va inkapsulyatsiya-dasturiy ta'minotning modul va qayta ishlatiladigan tuzilmalarini yaratishga imkon beradi. Shu bilan birga, dastur arxitekturasini mustahkamlash va kod sifatini oshirish maqsadida, dasturchilar ko'pincha abstrakt sinflar tushunchasiga murojaat qilishadi.

Abstrakt sinflar — bu faqat merosxo'rlik orqali oddiy sinflarda amalga oshirilishi lozim bo'lgan metodlar va atributlarni belgilash imkonini beruvchi sinflardir. Ular yordamida dasturiy tizimning mantiqiy tuzilishini aniqroq

shakllantirish, kodning qayta ishlatilishini oshirish va dasturiy xatoliklarni kamaytirish mumkin. Python dasturlash tilida abstrakt sinflarni yaratish va boshqarish uchun maxsus mo'ljallangan abc (Abstract Base Classes) moduli mavjud. Bu modul dasturchilarga ABC bazaviy sinfi va @abstractmethod dekoratori orqali abstrakt metodlarni aniqlash imkonini beradi. Shu yo'l bilan merosxo'r sinflarda kerakli metodlarning majburiy amalga oshirilishi ta'minlanadi, bu esa kodning kengaytirilishi va saqlanishini osonlashtiradi. Abstrakt sinflar va oddiy sinflar o'rtasidagi farqlar, shuningdek, interfeys tushunchasi bilan bog'liqligi ham dasturchilar uchun muhimdir. Abstrakt sinflar konseptual jihatdan dasturiy interfeyslarni belgilashda ishlatiladi va bu orqali loyiha tarkibidagi sinflar o'rtasidagi kelishuvni mustahkamlaydi. Ushbu maqola abstrakt sinflar va abc modulini batafsil tahlil qiladi, ularning dasturiy loyihalardagi ahamiyatini ko'rsatadi hamda Python dasturlash tilida samarali foydalanish bo'yicha amaliy tavsiyalar beradi. Shu bilan birga, maqolada abstrakt va oddiy sinflar, interfeys tushunchasi, merosxo'rlik jarayoni hamda kodning qayta ishlatilishi kabi muhim jihatlar ham tahlil qilinadi.

### **Muhakama va natija.**

Ushbu maqolada ko'rib chiqilganidek, abstrakt sinflar va Python dasturlash tilidagi abc (Abstract Base Classes) moduli dasturiy tizimlarni loyihalashda muhim vosita hisoblanadi. Abstrakt sinflar, bir tomondan, dasturiy kodning modul va qayta ishlatiladigan tuzilishini yaratishga yordam beradi, boshqa tomondan esa merosxo'rlik jarayonida kerakli metodlarning majburiy bajarilishini ta'minlaydi. Bu esa dasturiy xatoliklarning oldini olish va kodning mantiqiy izchilligini saqlash imkonini beradi.

Pythonning abc moduli quyidagi asosiy komponentlarni taqdim etadi:

ABC bazaviy sinfi: barcha abstrakt sinflarning asosiy bazasi sifatida xizmat qiladi.

@abstractmethod dekoratori: metodlarni abstrakt qilib belgilash va merosxo'r sinflarda ularni amalga oshirishni majburiy qilish imkonini beradi.

### **Bu imkoniyatlar murakkab loyihalarda quyidagi afzalliklarni beradi:**

1. Umumiy interfeyslarni yaratish va standartlashtirish – abstrakt sinflar orqali turli oddiy sinflarda bajarilishi majburiy bo'lgan metodlar aniqlanadi, bu esa dasturiy tizimning barcha komponentlari o'rtasida kelishuvni ta'minlaydi.
2. Kodning kengaytirilishi va qayta ishlatilishi – yangi sinflar yaratishda mavjud abstrakt sinflardan meros olish orqali kodni takrorlashga hojat qolmaydi.
3. Dasturiy xatoliklarni kamaytirish – agar merosxo'r sinf abstrakt metodni amalga oshirmasa, Python kompilyatori darhol xatolik chiqaradi, bu esa dastur ishga tushishidan oldin muammolarni aniqlash imkonini beradi. Muhokama davomida shuni ham ta'kidlash lozimki, abstrakt sinflar va interfeys tushunchalari o'xshash bo'lsa-da, ular bir xil emas. Interfeys faqat metodlar imzosini belgilasa, abstrakt sinflar metodlarni qisman amalga oshirish va atributlarni belgilash imkonini ham beradi. Shu

sababli, dasturiy dizayn jarayonida har bir holatda qaysi yondashuv samarali bo'lishini aniqlash muhimdir.

### **Abstrakt sinflarning amaliy qo'llanilishi.**

1. Umumiy interfeys yaratish-abstrakt sinflar turli turdagi obyektlar uchun umumiy interfeys yaratishda ishlatiladi. Bu, ayniqsa, katta tizimlarda komponentlar o'rtasida standart kelishuvni ta'minlash uchun muhimdir.

2. Kodning qayta ishlatilishi va kengaytirilishi-abstrakt sinflar yordamida umumiy funktsionallik bir joyda yig'iladi, merosxo'r sinflarda esa faqat o'ziga xos xususiyatlar qo'shiladi.

3. Polimorfizmni qo'llash-abstrakt sinflar yordamida merosxo'r sinflar bilan polimorfizm ishlatiladi. Ya'ni, turli oddiy sinflar bitta abstrakt sinf interfeysi orqali ishlatilishi mumkin.

1-masala.Talaba nomli abstrakt sinf yarating.Talaba sinfi quyidagi xususiyatlarga ega bo'lsin.Talabaning ismi, familiyasi va ID raqami.Baho\_hisoblash() — talabaning yakuniy bahosini aniqlovchi abstrakt metod.Holatini\_an qilash() — talabaning o'qish holatini aniqlovchi abstrakt metod.Malumot\_olish() — talaba haqida umumiy ma'lumotni qaytaruvchi metod.Ushbu abstrakt sinfdan BakalavrTalaba,magistrTalaba sinflari meros olsin va metodlar o'ziga mos tarzda amalga oshiring.

### **Python kodi:**

```
from abc import ABC, abstractmethod
class Talaba(ABC):
    def __init__(self, ism, familiya, ID_raqam):
        self.ism = ism
        self.familiya = familiya
        self.ID_raqam = ID_raqam
    @abstractmethod
    def baho_hisoblash(self):
        pass
    @abstractmethod
    def holatini_an qilash(self):
        pass
    def malumot_olish(self):
        return f"\"Talaba ismi:{self.ism}\\nTalaba familiyasi:{self.familiya}\\nTalaba ID_raqami: {self.ID_raqam}\""
class BakalavrTalaba(Talaba):
    def __init__(self, ism, familiya, ID_raqam, ball):
        super().__init__(ism, familiya, ID_raqam)
        self.ball = ball
```

```
def baho_hisoblash(self):
    if self.ball >= 87:
        return "5"
    elif self.ball >= 71:
        return "4"
    elif self.ball >= 60:
        return "4"
    else:
        return "2"
def holatini_anqlash(self):
    return "O'qimoqda" if self.ball >= 60 else "Akademik qarzidor"
```

```
class MagistrTalaba(Talaba):
    def __init__(self, ism, familiya, ID_raqam, ilmiy_ish_balli):
        super().__init__(ism, familiya, ID_raqam)
        self.ilmiy_ish_balli = ilmiy_ish_balli
    def baho_hisoblash(self):
        return "O'tdi" if self.ilmiy_ish_balli >= 70 else "Yiqildi"
    def holatini_anqlash(self):
        return "Bitiruvchi" if self.ilmiy_ish_balli >= 70 else "Qayta topshiradi"
```

```
t1 = BakalavrTalaba("Ali", "Karimov", 4304565, 78)
t2 = MagistrTalaba("Dilshod", "Rahmonov", 2029862, 65)
```

```
print(t1.malumot_olish())
print("Baho:", t1.baho_hisoblash())
print("Holat:", t1.holatini_anqlash())
print()
print(t2.malumot_olish())
print("Natija:", t2.baho_hisoblash())
print("Holat:", t2.holatini_anqlash())
```

**Natija:**

Talaba ismi:Ali

Talaba familiyasi:Karimov

Talaba ID\_raqami: 4304565

Baho: 4

Holat: O'qimoqda

Talaba ismi:Dilshod

Talaba familiyasi:Rahmonov

Talaba ID\_raqami: 2029862

Natija: Yiqildi

Holat: Qayta topshiradi

2-masala.Shakl nomli abstrakt sinf yarating.Shakl sinfi quyidagi abstrakt metodlarga ega bo'lsin.Yuzani\_hisoblash() — shakl yuzasini hisoblaydi.

Perimetrni\_hisoblash() — shakl perimetrini hisoblaydi. Ushbu abstrakt sinfdan to'rtburchak, doira sinflari meros olsin va metodlarni o'ziga mos ravishda amalga oshiring. So'ngra har bir shakl uchun yuzasi va perimetrini ekranga chiqaring.

**Python kodi:**

```
from abc import ABC, abstractmethod
import math

class Shakl(ABC):
    @abstractmethod
    def yuzani_hisoblash(self):
        pass
    @abstractmethod
    def perimetrni_hisoblash(self):
        pass

class Tortburchak(Shakl):
    def __init__(self, eni, boyi):
        self.eni = eni
        self.boyi = boyi
    def yuzani_hisoblash(self):
        return self.eni * self.boyi
    def perimetrni_hisoblash(self):
        return 2 * (self.eni + self.boyi)

class Doira(Shakl):
    def __init__(self, radius):
        self.radius = radius
    def yuzani_hisoblash(self):
        return math.pi * self.radius ** 2
    def perimetrni_hisoblash(self):
        return 2 * math.pi * self.radius

tortburchak = Tortburchak(4, 5)
doira = Doira(3)
print("To'rtburchak yuzi:", tortburchak.yuzani_hisoblash())
print("To'rtburchak perimetri:", tortburchak.perimetrni_hisoblash())
print()
print("Doira yuzi:", doira.yuzani_hisoblash())
print("Doira perimetri:", doira.perimetrni_hisoblash())
```

**Natija:**

|                       |                                |
|-----------------------|--------------------------------|
| To'rtburchak yuzi: 20 | Doira yuzi: 28.274333882308138 |
|-----------------------|--------------------------------|



To'rtburchak perimetri: 18

Doira perimetri: 18.84955592153876

**Xulosa.**

Ushbu maqolada abstrakt sinflar va Python dasturlash tilidagi abc (Abstract Base Classes) modulining mohiyati, turlari va amaliy qo'llanilishi batafsil tahlil qilindi. Tadqiqot natijalari shuni ko'rsatdiki, abstrakt sinflar dasturiy tizimlarni loyihalashda quyidagi jihatlarda katta ahamiyatga ega:

1. Umumiy interfeys yaratish va standartlashtirish
2. Kodning qayta ishlatilishi va kengaytirilishini ta'minlash
3. Dasturiy xatoliklarni oldini olish
4. Polimorfizm va modul dizaynini soddalashtirish
5. Amaliy loyihalardagi ahamiyati

Shu bilan birga, abstrakt sinflar va interfeys tushunchasi o'rtasida o'xshashliklar mavjud, ammo ular bir xil emas. Interfeys faqat metodlar imzosini belgilasa, abstrakt sinflar metodlarni qisman amalga oshirish, atributlarni belgilash va umumiy funktsionallikni taqdim etish imkoniyatiga ham ega.

**Foydalanilgan adabiyotlar.**

1. Mark Lutz. *Learning Python*. Fifth Edition. Published by O'Reilly Media, Inc., USA. 2013. – 1648 p.
2. Luciano Ramalho. *Fluent Python*. Second Edition. Published by O'Reilly Media, Inc., USA. 2015. – 792 p.
3. Dusty Phillips. *Python 3 Object-Oriented Programming*. Second Edition. Published by Packt Publishing Ltd., UK. 2018. – 356 p.
4. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Published by Addison-Wesley Professional, USA. 1994. – 395 p.
5. Robert C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*. Published by Prentice Hall, USA. 2008. – 464 p.
6. Python Software Foundation. *Python Documentation: abc module*. Python 3.12. Published by Python Software Foundation, USA. 2025. – online resource.
7. Python Software Foundation. *\*Python Documentation: Object-Oriented Programming\**. Python 3.12. Published by Python Software Foundation, USA. 2025. – online resource.
8. Real Python. *Abstract Base Classes in Python*. Published by Real Python Pty Ltd. 2024. – online resource.
9. GeeksforGeeks. *Abstract Classes in Python*. Published by GeeksforGeeks Media LLP, India. 2024. – online resource.
10. TutorialsPoint. *Python Abstract Base Classes Tutorial*. Published by TutorialsPoint, India. 2024. – online resource.

- 11.Karimberdiyevich, O. M., & Abdulaziz o'g'li, Y. M. (2024). SUN'IY INTELLEKTNING AFZALLIKLARI VA KAMCHILIKLARI. *IZLANUVCHI*, 1(1), 75-85.
- 12.Abdulaziz o'g'li, Y. M. (2025). WPFDA ANIMATSIYA YARATISHNI QO'LLANISHI. *MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS*, 1(4), 172-175.
- 13.Karimberdiyevich, O. M., & Abdulaziz o'g'li, Y. M. (2024). NEYRO KOMPYUTERLAR. *YANGI O 'ZBEKISTON, YANGI TADQIQOTLAR JURNALI*, 1(5), 19-27.
- 14.Karimberdiyevich, O. M., & Abdulaziz o'g'li, Y. M. (2024). K-YAQIN QO'SHNI ALGORITMI. *IZLANUVCHI*, 1(1), 122-124.
- 15.Abdulaziz o'g'li, Y. M. (2025). MOLIYA VA HISOB-KITOB ILOVALARIDA WPF BILAN ISHLASH. *MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS*, 1(4), 189-193.
- 16.Karimberdiyevich, O. M. (2024). NEYROEMULYATORLAR VA ULARNING QO'LLANILISHI. *YANGI O 'ZBEKISTON, YANGI TADQIQOTLAR JURNALI*, 1(5), 82-89.
- 17.Abdulaziz ogli, Y. M. (2025). WPF DA IKKI O'LCHOVLI VA UCH O'LCHOVLI GRAFIKALAR BILAN ISHLASHNING HAYOTGA TATBIQLARI. *MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS*, 1(4), 176-179.
- 18.Karimberdiyevich, O. M., & Abdulaziz o'g'li, Y. M. (2024). MASHINANI 'RGANISHDA TASNIFLASH VA REGRESSIYA. *IZLANUVCHI*, 1(1), 114-
- 19.Karimberdiyevich, O. M., & Abdulaziz o'g'li, Y. M. (2024). PIVOT JADVALI YARATISH VA TAHRIRLASH. *YANGI O 'ZBEKISTON, YANGI TADQIQOTLAR JURNALI*, 1(5), 28-30.
- 20.Karimberdiyevich, O. M., & Abdulaziz o'g'li, Y. M. (2024). MASHINANI O 'RGANISHDA TASNIFLASH VA REGRESSIYA. *IZLANUVCHI*, 1(1), 114-121.
- 21.Karimberdiyevich, O. M. (2024). O'ZBEKISTON RESPUBLIKASIDA BIG DATA NI RIVOJLANISH TENDENSIYALARI. *TA'LIM, TARBIYA VA INNOVATSIYALAR JURNALI*, 1(1), 147-151.
- 22.Karimberdiyevich, O. M., & Abdulaziz o'g'li, Y. M. (2024). ICHKI MUAMMOLARNI TUSHUNISH. *TA'LIM, TARBIYA VA INNOVATSIYALAR JURNALI*, 1(1), 98-104.