

Апстрактне основне класе

Апстрактна класа се сматра планом изградње других класа.

Креирањем апстрактне класе се омогућава креирање више метода које се морају креирати унутар сваке подкласе створене по плану апстрактне класе.

Апстрактна метода је метода која има декларацију али нема имплементацију.

Апстрактне класе нису способне да изврше инстанцијацију и зато им требају подкласе које омогућавају имплементацију тих апстрактних метода које су дефинисане у апстрактним класама.

Оне се најчешће користе када је потребно дизајнирати велике функционалне јединице.

Када је потребно имплементирати заједничку функционалност за све имплементације у компонентама, користе се апстрактне класе.

Када се потпуно имплементирају све апстрактне методе, тада се апстрактне класе називају интерфејси.

Разлози употребе апстрактних класа

Апстрактна класа омогућава дифолтну функционалност за подкласе.

Дефинисањем апстрактне основне класе, може се дефинисати заједнички интерфејс за апликативне програме (API – application program interface) за више подкласа.

Ове је корисно када трећа страна доприноси имплементацију, попут плагинса у апликацији, али помаже и при раду у великим тимовима или дугим кодовима, где је вођење рачуна о свим употребљеним класама отежано.

Начини употребе апстрактних класа

У Пајтону по дифолту, није могуће користити апстрактне класе, већ се користи модул abc (collections.abc подмодул) који даје основу за дефинисање апстрактних основних класа (abc – abstract base classes).

Модул користи помоћну класу ABC за обележавање методе основне класе као апстрактне а затим региструје специфичне класе као имплементације апстрактне класе.

Метод постаје апстрактан преко декоратора @abstractmethod.

ABC је дефинисан сет метода и особина које класа мора да имплементира да би се могла сматрати прилепљеном (duck-typed) инстанцом те класе.

Класа може проширити самој себи апстрактну основну класу да би се користила као инстанца те класе, али мора да има све тражене методе.

Употреба апстрактне основне класе

```
from abc import ABC, abstractmethod
```

```
class Zivotinja():
    def kretanje(self):
        pass
a = Zivotinja()          #nema izuzetaka

class Osoba(ABC):
    @abstractmethod
    def kretanje(self):
        pass
a = Osoba()              #izuzetak, jer se ne moze instancirati apstraktna klasa
```

Пример:

```
from abc import ABC, abstractmethod
class Zivotinja(ABC):
    @abstractmethod
    def kretanje(self):
        pass

class Covek(Zivotinja):
    def kretanje(self):
        print("covek koraca")
```

```
class Puz(Zivotinja):
    def kretanje(self):
        print("puz puzi")
```

```
R = Covek()
R.kretanje()           #covek koraca
R = Puz()
R.kretanje()          #puz puzi
```

Види се да пошто су све методе истог имена, приликом позивања методе подкласе долази до премошћавања методе надкласе чиме се метода подкласе успешно имплементира.

Израда лабораторијских вежби:

**Задатак 060:** Креирати апстрактну класу Полигон са апстрактном методом број\_страна. Подкласе Троугао и Хексагон наслеђују методу апстрактне класе али је и мењају. Приказати употребу апстрактне класе преко инстанцираних објеката подкласа.

```
from abc import ABC, abstractmethod
class Poligon(ABC):
    @abstractmethod
    def broj_strana(self):
        pass
```

```
class Trougao(Poligon):
    def broj_strana(self):
        print("Imam 3 strane.")
```

```
class Heksagon(Poligon):
    def broj_strana(self):
        print("Imam 6 strana.")
```

```
trougao = Trougao()
trougao.broj_strana()
heksagon = Heksagon()
heksagon.broj_strana()
```

**Задатак 061:** Апстрактне класе укључују атрибуте у методе па се могу тражити атрибути у одређеним класама дефинисањем преко @abstractproperty. Креирати апстрактну класу Родитељ са апстрактном методом опис. Подкласа Дете наслеђује методу апстрактне класе али је и мењају. Приказати употребу апстрактне методе преко дефинисања декоратора @abstractproperty.

```
import abc
from abc import ABC, abstractmethod
class Roditelj(ABC):
    @abc.abstractproperty
    def opis(self):
        return "Roditeljska apstraktna klasa."
```

```
class Dete(Roditelj):
    @property
    def opis(self):
        return "Decija klasa."
```

```
try:
    roditelj = Roditelj()
    print(roditelj.opis)
except Exception as err:
    print(err)
```

```
roditelj = Dete()
```

```
print(roditelj.opis)
```

Даје:

Can't instantiate abstract class Roditelj with abstract methods opis  
Decija klasa.

Види се да основна класа се не може инстанцирати пошто има само апстрактну верзију особности гетер методе.

**Задатак 062:** Креирати апстрактну класу са апстрактном методом. У две подкласе имплементирати апстрактну методу и извршити method override. Доказати да се објекти могу инстанцирати из подкласе али не и из надкласе.

```
from abc import ABC, abstractmethod
class ApstraktnaKlasa(ABC):
    def __init__(self, vrednost):
        self.vrednost = vrednost

    @abstractmethod
    def operacija(self):
        pass

class Sabiranje4(ApstraktnaKlasa):
    def operacija(self):
        return self.vrednost + 4

class Mnozenje4(ApstraktnaKlasa):
    def operacija(self):
        return self.vrednost * 4
```

```
x = Sabiranje4(10)
y = Mnozenje4(10)
print(x.operacija())
print(y.operacija())
z = ApstraktnaKlasa()
```

**Задатак 063:** Следећи код је креиран у намери да се имплементира апстрактна класа са апстрактним методама у две подкласе. Међутим добија се коментар о грешци приликом старовања кода.

```
from abc import ABC, abstractmethod
class ApstraktnaZivotinja:
    @abc.abstractmethod
    def kretanje(self):
        pass

    @abc.abstractmethod
    def govor():
        pass

class Patka(ApstraktnaZivotinja):
    def __init__(self):
        print('Apstraktna patka ' + ime + ' je kreirana.')
        self.ime = ime

    def kretanje(self):
        print('Gega se.')

    def govor():
        print('Kvak!')
```

Треба да даје:

Apstraktna patka Paja Patak je kreirana.

Kvak!

Gega se.

Задаци за самосталан рад:

60. Креирати апстрактну класу Животиња са апстрактном кретање. Подкласе Човек, Змија и Ајкула наслеђују методу апстрактне класе али је и мењају. Приказати употребу апстрактне класе преко инстанцираних објеката подкласа.
61. Коришћењем апстрактне класе са два аргумента (бројчане вредности), имплементирати апстрактну методу операција без тела методе. Креирати подкласе које имплементирају апстрактну методу тако што преко `method override` врше основне аритметичке операције.
62. Креирати апстрактну класу Тело са апстрактном методом површина без тела метода. У две подкласе Коцка и Квадар имплементирати апстрактну методу и извршити `method override`.
63. Поправити код из задатка 006 тако да се на екрану добије резултат позива метода подкласе.