LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK

Disusun Oleh :

Dimas Faturrohim (121140224)

PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI SUMATERA
2023

DAFTAR ISI

LAPORAN

Penjelasan singkat kelas abstrak, interface dan metakelas

Kelas abstrak, interface, dan metaclass adalah konsep penting dalam pemrograman berorientasi objek. Kelas abstrak adalah kelas yang tidak dapat diinstansiasi dan digunakan untuk mewakili konsep umum. Interface adalah kontrak yang menyatakan metode dan properti yang harus diimplementasikan oleh kelas lain. Metaclass adalah kelas yang digunakan untuk membuat kelas lainnya.

Kelas abstrak

Kelas Abstrak Kelas abstrak adalah kelas yang tidak dapat diinstansiasi. Kelas ini hanya digunakan untuk mewakili konsep umum dan menjadi tempat untuk mendefinisikan metode yang harus diimplementasikan oleh kelas anak. Untuk membuat kelas abstrak di Python, kita dapat menggunakan modul abc (Abstract Base Classes). Contoh implementasi kelas abstrak:

```
      from abc import ABC, abstractmethod

class Shape(ABC):
    @abstractmethod
    def area(self):
        pass

class Rectangle(Shape):
    def __init__(self, width, height):
        self.width = width
        self.height = height

    def area(self):
        return self.width * self.height

r = Rectangle(10, 20)
print(r.area()) # Output: 200
```

Interface

Interface adalah kontrak yang menyatakan metode dan properti yang harus diimplementasikan oleh kelas lain. Di Python, tidak ada sintaksis untuk mendefinisikan interface secara eksplisit. Namun, kita dapat menggunakan kelas abstrak untuk mensimulasikan konsep interface. Contoh implementasi interface.

```
from abc import ABC, abstractmethod

class Shape(ABC):
    @abstractmethod
    def area(self):
```

```
        pass

class Drawable(ABC):
    @abstractmethod
    def draw(self):
        pass

class Circle(Shape, Drawable):
    def __init__(self, radius):
        self.radius = radius

    def area(self):
        return 3.14 * (self.radius ** 2)

    def draw(self):
        print("Drawing circle...")

c = Circle(5)
print(c.area()) # Output: 78.5
c.draw() # Output: Drawing circle...
```

Metaclass

Metaclass adalah kelas yang digunakan untuk membuat kelas lainnya. Dalam Python, setiap kelas sebenarnya merupakan objek dari metaclass. Secara default, metaclass yang digunakan adalah type. Namun, kita dapat membuat metaclass sendiri dengan mendefinisikan kelas yang mewarisi dari type dan mengimplementasikan metode baru. Contoh implementasi metaclass:

```
class MyMetaClass(type):
    def __new__(cls, name, bases, dct):
        # Menambahkan atribut 'created_by' pada kelas yang dibuat
        dct['created_by'] = 'MyMetaClass'
        # Membuat kelas baru dengan metode new
        return super().__new__(cls, name, bases, dct)

class MyClass(metaclass=MyMetaClass):
    pass

# Mencetak atribut 'created_by' yang ditambahkan oleh metaclass
print(MyClass.created_by)
```

KESIMPULAN

Interface adalah sebuah kontrak yang menetapkan metode dan properti yang harus diimplementasikan oleh kelas lain. Dalam interface, tidak ada implementasi kode, hanya definisi metode dan properti yang harus diimplementasikan oleh kelas lain. Interface memungkinkan berbagai kelas yang berbeda untuk diatur secara homogen, meskipun memiliki implementasi yang berbeda. Interface digunakan ketika kita ingin membuat kelas-kelas yang berbeda, tetapi memiliki metode dan properti yang sama.

Kelas abstrak adalah kelas yang tidak dapat diinstansiasi dan memiliki setidaknya satu metode abstrak. Metode abstrak hanya berisi definisi metode yang harus diimplementasikan oleh kelas anak. Kelas abstrak memaksakan implementasi metode yang sama pada kelas-kelas anak, sehingga memudahkan kita dalam membuat kode yang konsisten dan mudah dipelihara. Perbedaan antara kelas abstrak dan interface adalah pada implementasinya. Kelas abstrak dapat memiliki implementasi kode, tetapi juga dapat memiliki metode abstrak yang harus diimplementasikan.

Kelas konkret adalah kelas yang dapat diinstansiasi dan memiliki implementasi lengkap dari setiap metode yang dimilikinya. Kelas konkret digunakan ketika kita ingin membuat sebuah objek yang memiliki implementasi lengkap dari setiap metodenya. Dalam bahasa pemrograman Python, hampir semua kelas yang kita definisikan adalah kelas konkret. Kita juga dapat menggunakan konsep pewarisan untuk membuat kelas konkret baru yang memiliki metode yang sama dengan kelas parentnya.

Metaclass adalah sebuah kelas yang digunakan untuk membuat kelas baru. Setiap kelas di Python memiliki sebuah metaclass yang menentukan bagaimana kelas tersebut akan dibuat dan beroperasi. Metaclass digunakan ketika kita ingin mengontrol cara sebuah kelas dibuat dan beroperasi. Dalam metaclass, kita dapat mengimplementasikan metode seperti new(), init(), call(), atau getitem() untuk mengatur perilaku kelas saat dibuat, diinstansiasi, atau diakses. Perbedaan utama antara metaclass dan inheritance biasa adalah pada saat pembuatan kelas. Dalam inheritance biasa, sebuah kelas dibuat dengan mewarisi metode dan properti dari kelas parent. Sedangkan dalam metaclass, sebuah kelas dibuat dengan menggunakan sebuah kelas khusus yang bertindak sebagai metaclass.

DAFTAR PUSTAKA

Beazley, D. M. (2019). *Python Essential Reference (5th Edition).* Addison-Wesley Professional.

Lutz, M. (2013). *Learning Python (5th Edition).* O'Reilly Media.

Pilgrim, M. (2009). *Dive Into Python 3.* Apress.

Rossum, G. V. (2009). *Python 3 Reference Manual.* CreateSpace.