LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS OBJEK (PBO)
TUGAS 6

Nama : Rio Aditya

NIM : 121140140

Kelas : RB

Program Studi Teknik Informatika

Institut Teknologi Sumatera

2023

BAB 1

1. KELAS ABSTRAK

Kelas abstrak ditandai dengan mewarisi kelas ABC (Abstract Base Class) Fungsi abstrak pada kelas abstrak ditandai dengan memberikan decorator @abstractmethod pada fungsi Meskipun kelas abstrak tidak dapat dibuat objeknya, dalam kelas abstrak kita dapat mendefinisikan konstruktor. Nanti akan dipanggil dari sub-kelas yang mewarisi kelas abstrak

Dalam pemrograman berorientasi objek, kelas abstrak merupakan kelas yang tidak dapat dibuat instance-nya. Namun, bisa membuat kelas yang mewarisi dari kelas abstrak. Biasanya, menggunakan kelas abstrak untuk membuat cetak biru untuk kelas lain. Demikian pula, metode abstrak adalah metode tanpa implementasi. Kelas abstrak mungkin atau mungkin tidak termasuk metode abstrak. Python tidak secara langsung mendukung kelas abstrak. Tapi itu memang menawarkan modul yang memungkinkan Anda mendefinisikan kelas abstrak. Untuk mendefinisikan kelas abstrak, Anda menggunakan abcmodul (kelas dasar abstrak).

```python
from abc import ABC, abstractmethod

# kelas Bentuk2D merupakan kelas abstrak dan tidak bisa diinstansiasi
class Bentuk2D(ABC):
  @abstractmethod
  def get_luas(self):
    # fungsi ini wajib diimplementasikan pada child-class
    pass

  @abstractmethod
  def get_keliling(self):
    # fungsi ini wajib diimplementasikan pada child-class
    pass
```

2. INTERFACE

Interface merupakan koleksi dari method atau fungsi-fungsi yang perlu disediakan oleh implementing class (child class). Interface mengandung metode yang bersifat abstract. Metode abstract akan memiliki satu-satunya deklarasi karena tidak adanya implementasi. Pengimplementasian sebuah interface merupakan sebuah cara untuk menulis kode yang elegan dan terorganisir. Dalam bahasa pemrograman Java, dikenal konsep Interface. Java tidak mengenal konsep Multiple Inheritance. Konsep Multiple Inheritance dimungkinkan dengan Interface. Dalam Python, konsep interface tidak menjadi komponen tersendiri. Python mendukung konsep Multiple Inheritance tanpa memerlukan sintaks khusus. Tapi kita perlu mengetahui konsep interface untuk memudahkan desain program skala besar. Interface biasanya digunakan untuk mendefinisikan kontrak. Mendefinisikan apa saja yang bisa dilakukan sebuah kelas, tanpa memperdulikan bagaimana implementasinya. Secara umum mirip dengan konsep kelas dan fungsi abstrak. Namun, seluruh fungsi-nya didefinisikan sebagai abstrak. Interface memiliki 2 jenis yaitu :

1. Informal Interface
   Informal interface merupakan kelas yang mendefinisikan metode yang dapat diganti tetapi tanpa force enforcement. Interface tersebut juga sering disebut Protocol atau duck typing. The duck typing sebenarnya mengeksekusi sebuah metode eksekusi pada sebuah object yang diharapkan. Jika yang diharapkan tidak memiliki masalah maka akan berjalan baik adanya. Interface dalam python disebut sebagai informal karena tidak dapat diperkuat secara formal.

```python
class Binatang:
    def __init__(self, binatang):
        self.list_binatang = binatang

    def __len__(self):
        return len(self.list_binatang)

    def __constraint__(self, binatang):
        return binatang in self.list_binatang

class zoo(Binatang):
    def __iter__(self):
        return iter(self.list_binatang)

way_kambas = zoo(["Gajah", "Burung", "Monyet"])

print(len(way_kambas))

print("Gajah" in way_kambas)
print("Badak" in way_kambas)
print("Monyet" not in way_kambas)

for binatang in way_kambas:
    print(binatang)
```

```
3
True
False
False
Gajah
Burung
Monyet
```

2. Formal Interface

   Seperti namanya jenis ini melakukan perintahnya secara formal. Dalam beberapa kondisi, protocols atau duck typing sering menciptakan kebingungan. Solusinya adalah dengan adanya formal interface. Dalam membuat formal, setidaknya dibutuhkan Abstract Base Classes. ABC sifatnya sederhana seperti base classes didefinisikan sebagai sebuah "abstrak" dan class abstract berisi beberapa metode. Lalu jika ada suatu kelas yang mengimplentasikan atau didorong dari kelas dasar ini maka semua kelas dasar harus mengimplentasikan semua metode tersebut. Interface dalam python disebut sebagai informal karena tidak dapat diperkuat secara formal.

```python
from abc import ABC
from abc import abstractmethod

class FiturGPS(ABC):
  @abstractmethod
  def get_lokasi(self):
    pass

class FiturBT(ABC):
  @abstractmethod
  def pair(self, other):
    pass

  @abstractmethod
  def unpair(self):
    pass
```

```python
class Smartphone(FiturGPS, FiturBT):
  def get_lokasi(self):
     print("Lokasi saat ini")

  def pair(self, other=None):
    self.other = other
    if other is not None:
      print("sukses!")
    else:
      print("gagal..")

  def unpair(self):
    if self.other is not None:
      print("terputus")
    else:
      print("tidak ada sambungan")
```

3. METACLASS

Metaclass di Python merupakan kelas dari kelas yang mendefinisikan bagaimana kelas berperilaku. Kelas itu sendiri merupakan turunan dari metaclass. Kelas dalam Python menentukan bagaimana instance kelas akan berperilaku. Untuk memahami metaclass dengan baik, seseorang harus memiliki pengalaman sebelumnya bekerja dengan kelas Python. Sebelum kita menyelam lebih dalam ke metaclasses, mari kita keluarkan beberapa konsep.

```python
ini_integer = 1234
ini_float = 1.234
ini_string = "Rio Aditya"

print("ini adalah ", type(ini_integer))
print("ini adalah ", type(ini_float))
print("ini adalah ", type(ini_string))
```

```
PS D:\Tugas Kuliah\Semester 4\PBO> python -u
ini adalah  <class 'int'>
ini adalah  <class 'float'>
ini adalah  <class 'str'>
PS D:\Tugas Kuliah\Semester 4\PBO>
```

BAB 2

KESIMPULAN

1. Interface merupakan koleksi dari method atau fungsi-fungsi yang perlu disediakan oleh implementing class (child class). Interface mengandung metode yang bersifat abstract. Metode abstract akan memiliki satu-satunya deklarasi karena tidak adanya implementasi. Interface dapat digunakan ketika ingin menerapkan polimorfisme, di mana objek dari kelas-kelas yang berbeda tetapi mengimplementasikan interface dapat digunakan secara serupa.
2. Kelas abstrak merupakan kelas yang tidak dapat dibuat instance-nya. Namun, bisa membuat kelas yang mewarisi dari kelas abstrak. Biasanya, menggunakan kelas abstrak untuk membuat cetak biru untuk kelas lain. Demikian pula, metode abstrak adalah metode tanpa implementasi. Kelas abstrak mungkin atau mungkin tidak termasuk metode abstrak. Kelas abstrak dapat digunakan ketika menghindari implementasi default jika ingin mencegah kelas-kelas turunan untuk menggunakan implementasi default yang telah diberikan, dapat menggunakan metode-metode abstrak dalam kelas abstrak. Dengan cara ini, kelas turunan harus memberikan implementasi khusus untuk metode-metode tersebut. Perbedaanya yaitu kelas abstrakterdapat fitur yang secara umum dimiliki oleh semua objek, sedangkan interface digunakan jika ada fitur yang ingin di implementasikan.
3. Kelas konkret merupakan kelas yang memiliki implementasi lengkap untuk semua metode yang didefinisikan dalam kelas tersebut. Kelas konkret dapat diinstansiasi secara langsung dan digunakan untuk membuat objek. Ini merupakan jenis kelas yang paling umum dan sering digunakan dalam pemrograman berorientasi objek. Kelas konkret dapat digunakan ketika memberikan implementasi konkret jika memiliki logika atau perilaku konkret yang ingin diimplementasikan, maka kelas konkret akan digunakan. Kelas konkret dapat menyediakan implementasi lengkap untuk metode-metode yang didefinisikan dalam kelas tersebut.
4. Metaclass di Python merupakan kelas dari kelas yang mendefinisikan bagaimana kelas berperilaku. Kelas itu sendiri merupakan turunan dari metaclass. Kelas dalam Python menentukan bagaimana instance kelas akan berperilaku. Untuk memahami metaclass dengan baik, seseorang harus memiliki pengalaman sebelumnya bekerja dengan kelas Python. Sebelum kita menyelam lebih dalam ke metaclasses, mari kita keluarkan beberapa konsep.

DAFTAR PUSTAKA

Modul 09. Kelas Abstrak dan Interface

https://www.pythontutorial.net/python-oop/python-abstract-class/

https://www.genius.education/blog/cara-membuat-interface-python-pysimplegui-dan-beberapa-fitur-lain

https://www.datacamp.com/tutorial/python-metaclasses