

Nama & NPM	Topik:	Tanggal:
Rengga Anggarah G1A022069	Pemrograman berorientasi objek	23/04/2023

1. buatlah perbedaan functional programming dan object oriented programming dalam bahasa pemrograman pyhon, serta berikan contoh masing-masing code nya?’

Jawab:

Functional Programming (FP) dan Object-Oriented Programming (OOP) adalah dua paradigma pemrograman yang berbeda dalam pendekatan untuk mengembangkan solusi perangkat lunak. Di bawah ini adalah perbedaan antara FP dan OOP dalam bahasa pemrograman Python secara rinci:

1. Pendekatan Utama

FP adalah pendekatan yang berfokus pada pemrosesan fungsi atau pemanggilan fungsi untuk menghasilkan output berdasarkan input. Sementara itu, OOP adalah pendekatan yang berfokus pada pemodelan dunia nyata dalam bentuk objek.

2. Fungsi vs. Objek

FP memperlakukan fungsi sebagai objek utama yang dapat digunakan sebagai argumen fungsi lainnya, dikembalikan sebagai hasil fungsi, dan disimpan dalam variabel. Sementara itu, OOP memperlakukan objek sebagai entitas utama yang dapat memiliki properti (variabel) dan metode (fungsi).

3. Mutabilitas

FP mendukung gaya pemrograman yang tidak berubah (immutable programming) yang berarti variabel tidak dapat diubah setelah dideklarasikan. Dalam OOP, objek biasanya dapat dimodifikasi dan dapat memiliki beberapa metode yang memodifikasinya secara internal.

4. Inheritance vs. Higher Order Functions

OOP menggunakan konsep pewarisan atau inheritance, di mana objek dapat mewarisi sifat dari objek lain. Sementara itu, FP menggunakan konsep fungsi higher order, di mana fungsi dapat menerima fungsi lain sebagai argumen dan mengembalikan fungsi.

5. Abstraksi

OOP menggunakan konsep abstraksi yang lebih tinggi, di mana kelas dapat mengelompokkan objek yang memiliki properti dan metode yang sama. FP menggunakan fungsi sebagai pengganti kelas dan menyediakan cara yang lebih sederhana untuk mengelompokkan logika program.

6. Exception Handling

Dalam OOP, exception handling dilakukan dengan menggunakan pernyataan try-except. Sementara itu, dalam FP, penanganan kesalahan lebih umumnya dilakukan dengan menggunakan monad, yaitu struktur data yang dapat mengontrol alur program pada saat runtime.

7. Efek Samping

FP menghindari efek samping dan mempromosikan gaya pemrograman fungsional yang bersih dan terisolasi. Di sisi lain, OOP memungkinkan efek samping dalam bentuk perubahan status objek.

Kesimpulannya, FP dan OOP adalah dua paradigma pemrograman yang berbeda dengan pendekatan yang berbeda dalam mengembangkan solusi perangkat lunak. Python dapat digunakan dalam kedua paradigma ini, dan pemilihan yang tepat tergantung pada kebutuhan dan tujuan dari proyek yang sedang dikembangkan.

2. berikan apa saja contoh pengimplementasian dari oop

Jawab:

Beberapa contoh pengimplementasian dari OOP (Object-Oriented Programming) adalah sebagai berikut:

1. Pembuatan class dan objek: Konsep dasar dari OOP adalah membuat class yang berisi data dan fungsi yang berkaitan dengan data tersebut. Objek kemudian dibuat dari class tersebut. Contoh class yang umum adalah class Mahasiswa, class Mobil, class Hewan, dan sebagainya.
2. Encapsulation: Encapsulation adalah konsep OOP yang berarti menyembunyikan informasi detail dari sebuah objek dan hanya menampilkan informasi yang diperlukan saja. Contoh implementasi encapsulation adalah dengan membuat atribut atau method pada class dengan akses modifier private atau protected.
3. Inheritance: Inheritance adalah konsep OOP yang berarti mewarisi sifat dari class yang lebih umum ke class yang lebih khusus. Contoh implementasi inheritance adalah class Bapak sebagai class induk dan class Anak sebagai class turunan. Anak memiliki sifat yang sama dengan Bapak, namun juga dapat memiliki sifat yang unik.
4. Polymorphism: Polymorphism adalah konsep OOP yang memungkinkan objek dari class yang berbeda memiliki method dengan nama yang sama namun dengan perilaku yang berbeda-beda. Contoh implementasi polymorphism adalah menggunakan method override atau overloading.
5. Abstraksi: Abstraksi adalah konsep OOP yang berarti hanya menampilkan informasi yang diperlukan dari sebuah objek dan menyembunyikan informasi detailnya. Contoh implementasi

abstraksi adalah dengan membuat interface atau abstract class yang hanya menampilkan method atau fungsi yang harus diimplementasikan oleh class turunannya.

6. Event-driven programming: Event-driven programming adalah konsep OOP yang memungkinkan program merespon suatu event atau kejadian yang terjadi pada objek. Contoh implementasi event-driven programming adalah dengan menggunakan listener atau handler untuk merespon suatu event pada GUI (Graphical User Interface) seperti tombol yang ditekan