NAMA : RAJA A.H. SIHOMBING
NIM : 4243550015
KELAS: PSIK 24A
UTS PBO

1. Prinsip encapsulation, inheritance, polymorphism, dan abstraction dan contoh analogi dalam kehidupan nyata:
- Encapsulation (Pengkapsulan):
Membungkus data dan metode yang bekerja pada data tersebut dalam satu unit dan membatasi akses langsung ke data.
Analogi dalam kehidupan nyata: kapsul obat
- Inheritance (Pewaris):
Memungkinkan class baru mewarisi sifat dan perilaku dari class yang sudah ada.
Analogi dalam kehidupan nyata: Seperti hubungan orang tua dan anak.
- Polymorphism (Polimorfisme):
Kemampuan suatu objek untuk mengambil banyak bentuk atau merespon berbeda terhadap metode yang sama.
Analogi dalam kehidupan nyata: Seorang guru yang bisa berperan sebagai pengajar di kelas, orang tua di rumah, atau pelanggan toko.
- Abstraction (Abstraksi):
Menyembunyikan detail implementasi kompleks dan hanya menampilkan fungsionalitas esensial.
Analogi dalam kehidupan sehari-hari: Mengemudi mobil - kita hanya perlu tahu cara mengoperasikan kemudi, gas, dan rem tanpa perlu memahami detail kerja mesin di bawah kap.

2. Kelebihan menggunakan java versi terbaru (Java 21) dibanding versi-versi sebelumnya dalam konteks pengembangan berbasis OOP dan 2 contoh fitur modern java 21 dan penjelasan:
a. Record Patterns (JEP 440)
Memungkinkan dekonstruksi record dengan cara yang elegan.
Manfaat: Menyederhanakan penanganan objek immutable (seperti record) dengan pola dekonstruksi yang lebih ekspresif.
b. Virtual threads (JEP 444)
Meskipun bukan fitur OOP langsung, virtual threads memungkinkan implementasi pola concurrent OOP yang lebih efisien.
Manfaat: Memudahkan pengembangan aplikasi concurrent berbasis OOP tanpa khawatir overhead thread tradisional.

KIKY

3. Perbedaan class dan object
class : class adalah blueprint / template yang mendefinisikan atribut dan perilaku
object : object adalah instance konkret dari class tersebut

4. Enkapsulation pada class Bank Account
Gunakan atribut balance sebagai private dan akses dengan set Balance (>< deposite >< withdrawl), ini mencegah perubahan langsung balance sembarangan. Enkapsulation penting untuk menjaga data sensitif dan mencegah kesalahan atau manipulasi.

5. Constructor chaining pada pewarisan di Java.
Constructor chaining adalah proses memanggil constructor lain (baik dalam class yang sama atau superclass) dari constructor. Jika constructor superclass tidak dipanggil secara eksplisit, Java akan secara otomatis memanggil constructor default (tanpa parameter) dari superclass. Jika superclass tidak memiliki constructor default, akan terjadi error kompil

Ilustrasi class karyawan dan manager

```
Class karyawan {
    String nama;
    String id;
    Karyawan (String nama, String id)
{
        this.nama = nama;
        this.id = id;
    }
}

Class Manager extends karyawan{
        String departemen;
        Manager (String nama, String id, String departemen){
        super (nama, id);
        this.departemen = departemen;
    }
}
```

KIKY

6. Jelaskan bagaimana penggunaan interface mendukung konsep polymorphism, dan berikan contoh penggunaan dalam sistem pemesanan makanan online.

: interface mendukung polymorphism karena memungkinkan berbagai class berbeda mengimplementasikan method yang sama dengan cara masing-masing.

Contoh interface:

```
interface payment {
    void processPayment(double amount);
}
class CreditCard implements Payment {
    public void processPayment(double
amount) { ... }
}
```

7. Perbandingan Abstract class, interface, dan Sealed class

- Abstract class: untuk hierarki dengan logika umum
- ~~untuk beberapa perilaku (Java 8+ bisa berisi default~~
- Interface: untuk kontrak fleksibel (Java 8+ bisa berisi default method).
- Sealed class: untuk membatasi pewarisan

Kapan digunakan

- Abstract class: Bagi logika umum
- Interface: Bagi perilaku berbeda
- Sealed class: Bagi hierarki terkontrol

KKY