PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK

Nama	Ruben Kristanto	No. Modul	4
NPM	2306214624	Tipe	TP

1. Abstraksi adalah salah satu konsep utama dalam OOP yang berfungsi untuk menyederhanakan sistem kompleks dengan hanya menampilkan bagian penting dan menyembunyikan detail implementasi yang tidak relevan. Dalam Java, abstraksi diimplementasikan menggunakan kelas abstrak dan interface yang telah kita pelajari sebelumnya hanya berisi method2 sederhana atau bahkan nama methodnya saja serta inisialisasinya yang akan diterapkan kepada child class atau class2 lainnya sesuai keperluan melalui override.

   Dalam Java, abstraksi sering diterapkan melalui metode getter dan setter, yang memungkinkan akses dan modifikasi variabel private tanpa memperlihatkan detail implementasinya. Ini melindungi data internal dan menjaga kesederhanaan dalam pengelolaan kode.

   Pentingnya abstraksi terletak pada kemampuannya untuk memecah sistem kompleks menjadi bagian-bagian yang lebih kecil dan mudah dikelola. Hal ini tidak hanya meningkatkan keterbacaan kode, tetapi juga mempermudah pemeliharaan dan pengembangan lebih lanjut. Misalnya, dalam kelas pengelolaan data pengguna, detail penyimpanan data dapat disembunyikan sambil menyediakan metode untuk berinteraksi dengan data tersebut, menjaga keamanan dan mencegah modifikasi yang tidak diinginkan.

   Berikut adalah contoh penerapannya :

```
// Java program to illustrate the

// concept of Abstraction

abstract class Shape {

    String color;



    // these are abstract methods

    abstract double area();
```

```
    public abstract String toString();

    // abstract class can have the constructor
    public Shape(String color)
    {
        System.out.println("Shape constructor called");
        this.color = color;
    }

    // this is a concrete method
    public String getColor() { return color; }
}
class Circle extends Shape {
    double radius;

    public Circle(String color, double radius)
    {

        // calling Shape constructor
        super(color);
        System.out.println("Circle constructor called");
        this.radius = radius;
    }

    @Override double area()
```

```
    {
        return Math.PI * Math.pow(radius, 2);
    }

    @Override public String toString()
    {
        return "Circle color is " + super.getColor()
            + "and area is : " + area();
    }
}
class Rectangle extends Shape {

    double length;
    double width;

    public Rectangle(String color, double length,
                     double width)
    {
        // calling Shape constructor
        super(color);
        System.out.println("Rectangle constructor called");
        this.length = length;
        this.width = width;
    }
```

```
    @Override double area() { return length * width; }


    @Override public String toString()
    {
        return "Rectangle color is " + super.getColor()
            + "and area is : " + area();
    }
}
public class Test {
    public static void main(String[] args)
    {
        Shape s1 = new Circle("Red", 2.2);
        Shape s2 = new Rectangle("Yellow", 2, 4);

        System.out.println(s1.toString());
        System.out.println(s2.toString());
    }
}
```

Referensi :

[1] tim netlab UI, "Praktikum Pemrograman Berorientasi Objek: Modul 5 - Abstraction," [Online]. Available: C:\Users\Kent Maynard\Documents\PCU\SKRIPSI\Sample Data MITRAPAK. Accessed: September 2024.

[2] GeeksforGeeks, "Abstraction in Java," GeeksforGeeks, Nov. 14, 2017. [Online]. Available: https://www.geeksforgeeks.org/abstraction-in-java-2/. [Accessed: Sep. 29, 2024].

2. Table ini berisi perbedaan antara interface dengan abstract class

Poin	Kelas Abstrak	Antarmuka
Definisi	Tidak dapat diwujudkan; berisi metode abstrak (tanpa implementasi) dan konkret (dengan implementasi)	Menentukan serangkaian metode yang harus diimplementasikan oleh suatu kelas; metode bersifat abstrak secara default.
Metode Implementasi	Dapat memiliki metode terimplementasikan dan abstrak.	Metode bersifat abstrak secara default; Java 8, dapat memiliki metode default dan statis.
Warisan	kelas hanya dapat mewarisi dari satu kelas abstrak.	Suatu kelas dapat mengimplementasikan beberapa antarmuka.
Pengubah Akses	Metode dan properti dapat memiliki pengubah akses apa pun (publik, dilindungi, privat).	Metode dan properti secara implisit bersifat publik.
Variabel	Dapat memiliki variabel anggota (final, non-final, statis, non-statis).	Variabel secara implisit bersifat publik, statis, dan final (konstanta).

Abstract class dapat berisi method yang sudah diisi maupun tidak tetapi method ini dapat di override perbedaannya dengan interface, interface hanya berisi nama method tersebut yang nanti akan dioverride juga oleh class yang meng extends atau implements.

Sebuah class dapat mewarisi abstract class hanya sekali sedangkan banyak interface dapat diterapkan dalam satu waktu.

Interface access modifiernya hanya bisa bersifat publik sedangkan abstract dapat menggunakan semua modifier seperti publik, protected, dan private.

Interface sejak java 8 dan versi-versi berikutnya dapat memiliki method yang ada isinya(metode terimplementasikan) melalui default method dan static methods.

Referensi :

[1] GeeksforGeeks, "Difference Between Abstract Class and Interface in Java," GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/difference-between-abstract-class-and-interface-in-java/. [Accessed: Sep. 29, 2024].

3. SQL (Structured Query Language) adalah bahasa standar yang digunakan untuk mengelola dan memanipulasi basis data relasional. SQL memungkinkan pengguna untuk melakukan operasi seperti mengambil data, menambahkan data, memperbarui data, dan menghapus data dari tabel dalam database. Selain itu, SQL digunakan untuk mendefinisikan struktur data, mengelola hak akses, serta mengontrol transaksi dalam basis data untuk menjaga konsistensi dan integritas data.

   PostgreSQL adalah sistem manajemen basis data relasional open-source yang kuat dan mendukung standar SQL. Dikenal karena keandalan, skalabilitas, dan kinerjanya, PostgreSQL memungkinkan penyimpanan dan pengelolaan data terstruktur dalam tabel relasional, dengan dukungan untuk fitur-fitur canggih seperti ACID (Atomicity, Consistency, Isolation, Durability), foreign keys, trigger, dan stored procedures. Selain itu, PostgreSQL mendukung berbagai tipe data, termasuk JSON dan array, serta dapat diperluas dengan ekstensi untuk menambahkan fungsionalitas tambahan. Sebagai database yang lintas platform, PostgreSQL banyak digunakan dalam aplikasi web, sistem enterprise, dan analisis data skala besar.

   Query digunakan untuk mencari atau mengubah sesuatu dalam database tau lebih tepatnya bagian dari table database.

   Table adalah bentuk dari database yang berisi kolom dan baris

   Baris, yang juga dikenal sebagai catatan, terdiri dari sel-sel yang menyimpan nilai-nilai untuk masing-masing kolom. Berbeda dengan kolom, baris dapat berisi berbagai tipe data. Keunikan dari baris adalah bahwa semua data di dalamnya mewakili satu objek atau entitas tertentu. Selain itu, tidak seperti kolom, jumlah baris dalam sebuah tabel tidak dibatasi dan tidak perlu diketahui sebelumnya.

   Kolom adalah bagian dari sebuah tabel. Dalam basis data relasional, kolom adalah kumpulan nilai data dari jenis tertentu; terdapat satu nilai kolom untuk setiap baris dalam basis data.

   Not NULL berarti bagian dari database tersebut harus ada isinya atau tidak bisa diisi dengan null

   Hubungan satu-ke-banyak (one-to-many) terjadi ketika satu catatan di satu tabel terkait dengan banyak catatan di tabel lain. Sebagai contoh, seorang Pelanggan dapat membuat banyak Pesanan. Misalnya, dalam tabel pelanggan, setiap pelanggan hanya memiliki satu baris, namun di tabel pesanan, pelanggan tersebut bisa memiliki beberapa pesanan yang berbeda. Dalam hal ini, satu pelanggan bisa memiliki beberapa pesanan, tetapi setiap pesanan hanya terkait dengan satu pelanggan. Di sisi lain, hubungan banyak-ke-satu (many-to-one) adalah kebalikan dari hubungan satu-ke-banyak, di mana banyak catatan dari satu tabel terkait dengan satu catatan di tabel lainnya.

Referensi :

[1] Moez Ali, "What is PostgreSQL? An Introduction," DataCamp. [Online]. Available: https://www.datacamp.com/blog/what-is-postgresql-introduction. [Accessed: Sep. 29, 2024].

[2]Varshachoudhary, “What is SQL?,” *GeeksforGeeks*, Oct. 04, 2021. https://www.geeksforgeeks.org/what-is-sql/. [Acessed: Sep. 29,2024].
[3]“SQL Terms Beginners Should Know,” *LearnSQL.com*, Nov. 10, 2020. https://learnsql.com/blog/sql-terms-for-beginners/. [Acessed: Sep. 29,2024].

4. Java Framework adalah sebuah platform atau kumpulan kode yang telah disediakan sebelumnya, yang digunakan oleh pengembang Java untuk membuat aplikasi Java maupun aplikasi web. Secara sederhana, Java Framework terdiri dari sejumlah kelas dan fungsi yang telah didefinisikan untuk menangani input, mengelola perangkat keras, dan berinteraksi dengan perangkat lunak sistem. Framework ini berfungsi sebagai fondasi yang memudahkan pengembang dalam membangun aplikasi dengan menambahkan kode mereka sendiri.

   Spring Boot adalah sebuah framework berbasis Java yang dirancang untuk menyederhanakan pengembangan aplikasi berbasis Spring, terutama dalam membuat aplikasi stand-alone dan siap produksi. Framework ini menawarkan konfigurasi otomatis (auto-configuration), yang meminimalkan kebutuhan konfigurasi manual, serta menyediakan embedded server seperti Tomcat, memungkinkan aplikasi dijalankan langsung tanpa server eksternal. Dengan menggunakan starter dependencies, Spring Boot memudahkan integrasi berbagai komponen, dan dilengkapi dengan fitur-fitur siap produksi seperti monitoring dan logging.

   Referensi :
[1]“What is Framework in Java - Javatpoint,” *www.javatpoint.com*, 2021. https://www.javatpoint.com/what-is-framework-in-java. [Accessed: 29 Sep. 2024]