**Modul 3**

**Konstruktor dan Overloading**

* 1. **Konstruktor**

Construktor/Konstruktor adalah fungsi khusus yang sama dengan nama class dan tidak

mempunyai tipe kembalian, tidak juga viod. Kegunaan utama adalah untuk mendefinisikan suatu objek dari class. Secara default, construktor tidak ber-parameter.

Beberapa karakteristik yang dimiliki oleh constructor:

a. Method constructor harus memiliki nama yang sama dengan nama class.

b. Tidak mengembalikan suatu nilai (tidak ada keyword return).

c. Satu class memiliki lebih dari satu constructor (overloading constructor).

d. Dapat ditambah access modifier public, private, protected maupun default.

e. Suatu constructor bisa dipanggil oleh constructor lain dalam satu class.

Struktur dari konstruktor :

Class nama\_kelas

{

Nama\_kelas() // konstruktor

{

//isi dari konstruktor

}

//isi dari kelas

}

Contoh 1

public class Mahasiswa

{

public int nilai; public String nama, nim;

public Mahasiswa(String a, String b, int c) // konstruktor

{

nim = a; nama = b; nilai = c;

}

}

Contoh 2

public class Mahasiswa

{

public int nilai; public String nama, nim;

public Mahasiswa(String a, String b, int c)

{

nim = a;

nama = b;

nilai = c;

}

public void cetak()

{

System.out.println("nim : " +nim);

System.out.println("nama : " +nama);

System.out.println("nilai : " +nilai);

}

}

class tesMahasiswa

{

public static void main(String args[])

{

System.out.println("program mulai");

Mahasiswa m = new Mahasiswa("10218001", "rara", 89);

m.cetak();

}

}

* 1. **Overload**

Overload dapat terjadi pada metode atau konstruktor. Overloada pada metode yaitu pendefinisian dua atau lebih metode dengan nama yang sma tapi parameternya berbeda. Perbedaan parameter dapat ditentukan dari tiga hal, yaitu :

1. Berbeda tipe datanya, meskipun jumla parameternya sama
2. Berbeda jumlahnya, meskipun tipe data parameternya sama
3. Kombinasi dari opsi pertama dan kedua
   1. **This**

Kata kunci ini digunakan dalam sebuah kelas untuk menyatakan object yang sedang berlaku. Kata kunci *this* sangat berguna untuk menunjukkan suatu member dalam class-nya sendiri. *this* dapat digunakan baik untuk data member maupun untuk function member, serta dapat juga digunakan untuk konstruktor

Contoh 3

import java.util.Scanner;

//contoh overload konstruktor

public class segiempat

{

public double panjang = 4.5;

public double lebar = 5;

public segiempat()

{

}

public segiempat(double panjang)

{

this.panjang = panjang;

}

public segiempat(double panjang, double lebar)

{

this.panjang = panjang;

this.lebar = lebar;

}

public void hit\_luas()

{

double luas;

luas = panjang \*lebar;

System.out.print("panjang : " +panjang);

System.out.println(" lebar: " +lebar);

System.out.println("luas segiempat : " +luas);

}

}

class demokelas

{

public static void main(String [] args)

{

double x, y;

segiempat se1 = new segiempat();

se1.hit\_luas();

segiempat se2 = new segiempat(3);

se2.hit\_luas();

segiempat se3 = new segiempat(3, 7);

se3.hit\_luas();

Scanner input = new Scanner (System.in);

System.out.print("input panjang : ");

x = input.nextDouble();

System.out.print("input lebar: ");

y = input.nextDouble();

segiempat se4 = new segiempat(x, y);

se4.hit\_luas();

}

}

Contoh 4

public class Matematika

{

public double kuadrat (double nilai)

{

return nilai \* nilai;

}

public int kuadrat (int nilai)

{

return nilai \* nilai;

}

public double kuadrat (String nilai)

{

double bilangan;

bilangan = Double.valueOf(nilai).doubleValue();

return bilangan \* bilangan;

}

}

class demometode

{

public static void main(String [] args)

{

Matematika matematika1= new Matematika ();

//matematika1.nilai = 25;

System.out.println(matematika1.kuadrat(25.0));

System.out.println(matematika1.kuadrat(25));

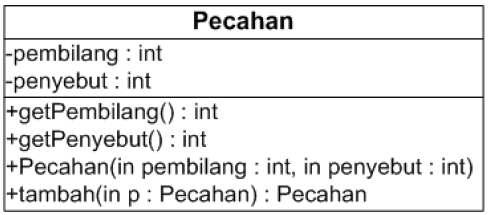
System.out.println(matematika1.kuadrat("25"));

}

}

Contoh 5

Diketahui sebuah class seperti di samping. Konstruktor memiliki 2 paramter untuk menge-set kedua atribut dari class Pecahan.

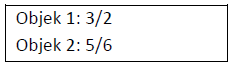


Method tambah merupakan method untuk menambahkan sebuah objek Pecahan dengan objek Pecahan lainnya. Method ini merupakan function dengan nilai yang dikembalikan merupakan hasil penambahan kedua objek Pecahan dengan tipe data class Pecahan tersebut.

Berdasarkan informasi tersebut,

1. Kodekanlah class di samping beserta logika dari setiap method

2. Bentuklah sebuah Main class yang membentuk 2 objek sebagai berikut:



Setelah membentuk kedua objek, tambahkan kedua objek tersebut, dan tampilkan hasil penambahan kedua objek tersebut

public class Pecahan {

public int pembilang, penyebut;

public int getpembilang() {

return pembilang; }

public int getpenyebut() {

return penyebut; }

Pecahan(int pembilang, int penyebut) {

this.penyebut = penyebut; this.pembilang = pembilang; }

public Pecahan tambah(Pecahan p)

{

int data1, data2; int data3; Pecahan d1;

d1 = p; data1 = p.penyebut; data2 = p.pembilang;

Pecahan d2 = new Pecahan(5,6);

System.out.print("objek2 : "); System.out.print(+d2.getpembilang());

System.out.print("/"); System.out.print(+d2.getpenyebut());

System.out.println();

p.penyebut = data1 \* d2.penyebut;

p.pembilang = d2.penyebut\*data2 + data1\*d2.pembilang;

return p;

}

class demopecahan

{

public static void main(String [] args)

{

Pecahan p;

int x, y;

Pecahan d1 = new Pecahan(3,2);

System.out.print("objek1 : ");

System.out.print(+d1.getpembilang());

System.out.print("/");

System.out.print(+d1.getpenyebut());

System.out.println();

//pecahan hasil = new pecahan(x, y);

p = d1.tambah(d1);

x = p.pembilang;

y = p.penyebut;

System.out.print( "hasil : " +x);

System.out.print("/");

System.out.print(+y);

System.out.println();

}

}

* 1. **Tugas Praktikum**

Diketahui sebuah class seperti di samping. Terdapat dua konstruktor untuk menge-set kedua atribut dari class Tabungan

Method simpanuang digunakan utuk menyimpan uang yang akan disimpan dan akan menambahkan dengan saldo sebelumnya.

Method tarikuang digunakan untuk menarik uang yang ada ditabungan, jika uang yang ditarik lebih kecil dari uang yang ada ditabungan maka tabungan akan berkurang sesuai dengan jumlah yang akan ditarik, jika uang yang ditarik lebih besar dari uang yang ada ditabungan maka penarikan gagal dan saldo ditabungan jumlahnya tidak berkurang.

Berdasarkan informasi tersebut,

Kodekanlah class di samping beserta logika dari setiap method

Bentuklah sebuah Main class yang membentuk, dengan output sebagai berikut :

nomer pin anda : 1111

simpan uang : 20000

saldo anda : 20000

simpan uang : 40000

saldo anda : 40000

simpan uang : 10000

OK

saldo sekarang = 30000

simpan uang : 50000

GAGAL

saldo sekarang = 30000

nomer pin anda telah berganti menjadi : 2222

|  |  |
| --- | --- |
| **Diperiksa Tanggal :** | **Paraf Dosen :** |
| **Catatan :** | |