**TUGAS 8**

**MATA KULIAH ALGORITMA DAN DASAR PEMROGRAMAN SEMESTER GENAP TAHUN AKADEMIK 2023-2024**



**Oleh:**

**AHMAD DIAN KAMAL**

**4123002**

**PROGRAM STRATA-1 JURUSAN SISTEM INFORMASI**

**UNIVERSITAS PESANTREN TINGGI DARUL 'ULUM JOMBANG**

**2024**

Latihan 1. Overriding

Apa yang terjadi bila program berikut ini dikompile dan dijalankan? Jelaskan !

package pkg8;

/\*\*

\*

\* @author KAMAL

\*/

public class Base{

private void amethod(int iBase){

System.out.println("Base.amethod");

}

}

class Over extends Base{

public static void main(String argv[]){

Over o = new Over();

int iBase=0;

o.amethod(iBase);

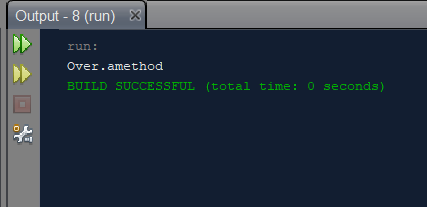
}

public void amethod(int iOver){

System.out.println("Over.amethod");

}

}



Berhasil tanpa kendala

Latihan 2. Overloading

Apa yang terjadi bila program berikut ini dikompile dan dijalankan? Jelaskan !

package pkg8;

/\*\*

\*

\* @author KAMAL

\*/

class MyParent {

int x, y;

MyParent(int x, int y){

this.x = x;

this.y = y;

}

public int addMe(int x, int y){

return this.x + x + y + this.y;

}

public int addMe(MyParent myPar){

return addMe(myPar.x, myPar.y);

}

}

class MyChild extends MyParent{

int z;

MyChild (int x, int y, int z) {

super(x,y);

this.z = z;

}

public int addMe(int x, int y, int z){

return this.x + x + this.y + y + this.z + z;

}

public int addMe(MyChild myChi){

return addMe(myChi.x, myChi.y, myChi.z);

}

public int addMe(int x, int y){

return this.x + x + this.y + y;

}

}

public class MySomeOne{

public static void main(String args[]){

MyChild myChi = new MyChild(10, 20, 30);

MyParent myPar = new MyParent(10, 20);

int x = myChi.addMe(10, 20, 30);

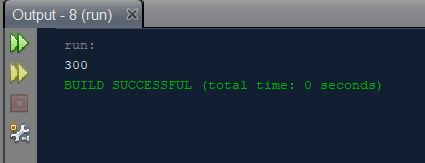
int y = myChi.addMe(myChi);

int z = myPar.addMe(myPar);

System.out.println(x + y + z);

}

}



Latihan 3. Overloading

Apa yang terjadi bila program berikut ini dikompile dan dijalankan? Jelaskan !

package pkg8;

/\*\*

\*

\* @author KAMAL

\*/

public class MyClass{

void myMethod(int i) {

System.out.println("int version");

}

void myMethod(String s) {

System.out.println("String version");

}

public static void main(String args[]){

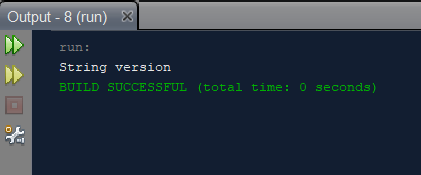
MyClass obj = new MyClass();

char ch = 'c';

obj.myMethod("hello");

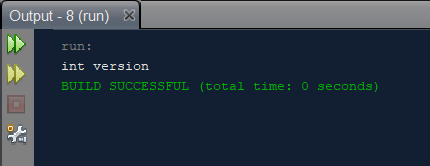
}

}



Untuk output pertama dapat menghasilkan output seperti diatas karena kita mengisi objek pada method obj.method(“hello”);

Maka akan terbaca string version



Untuk output pertama dapat menghasilkan output seperti diatas karena kita mengisi objek pada method obj.method(ch);

Maka akan terbaca int version

Latihan 4. Mengimplementasikan UML class diagram dalam program

package pkg8;

/\*\*

\*

\* @author KAMAL

\*/

class Orang {

private String nama;

private int umur;

public Orang(String nama) {

this.nama = nama;

}

public Orang(String nama, int umur) {

this.nama = nama;

this.umur = umur;

}

public void Info() {

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

if (umur != 0) {

System.out.println("Umur: " + umur);

}

}

}

class Dosen extends Orang {

private int nip;

public Dosen(String nama) {

super(nama);

}

public Dosen(String nama, int nip) {

super(nama);

this.nip = nip;

}

public Dosen(String nama, int nip, int umur) {

super(nama, umur);

this.nip = nip;

}

@Override

public void Info() {

super.Info();

System.out.println("NIP: " + nip);

}

}

public class TesLatihan {

public static void main(String args[]) {

System.out.println("Memasukkan identitas dosen 1 : Agus");

Dosen dosen1 = new Dosen("Agus");

System.out.println("Memasukkan identitas dosen 2 : Budi, NIP. 1458");

Dosen dosen2 = new Dosen("Budi", 1458);

System.out.println("Memasukkan identitas dosen 3 : Iwan, NIP. 1215, umur 47");

Dosen dosen3 = new Dosen("Iwan", 1215, 47);

System.out.println();

dosen1.Info();

System.out.println();

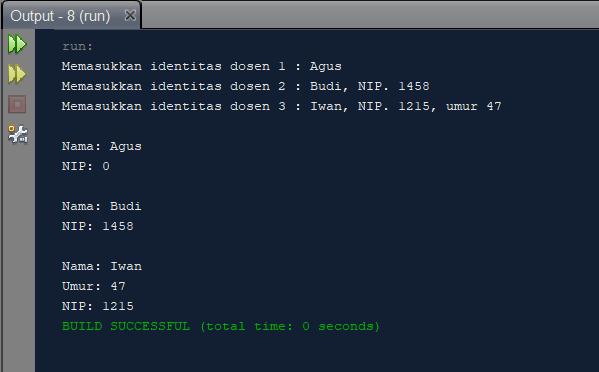
dosen2.Info();

System.out.println();

dosen3.Info();

}

}



* Inheritance memungkinkan kelas Dosen untuk mewarisi atribut dan metode dari kelas Orang.
* Overriding digunakan untuk memberikan implementasi spesifik dari metode Info() dalam kelas Dosen.
* Constructors di kelas turunan memanggil konstruktor kelas induk menggunakan super(), tetapi tidak diwariskan secara langsung.

Tugas **1:**

package pkg8;

/\*\*

\*

\* @author KAMAL

\*/

class RerataNilai {

public double average(double... angka) {

double hitung = 0;

for (double Angka : angka) {

hitung += Angka;

}

return hitung / angka.length;

}

}

public class TesTugas1{

public static void main(String args[]){

RerataNilai rn = new RerataNilai();

System.out.println("Rerata nilai 21 dan 13 adalah : " + rn.average(21, 13));

System.out.println("Rerata nilai 19.3 dan 39.5 adalah : " + rn.average(19.3, 39.5));

System.out.println("Rerata nilai 123, 567 dan 744 adalah : " + rn.average(123, 567, 744));

}

}

* Kelas RerataNilai:

Kelas ini memiliki satu metode average yang menerima sejumlah argumen variabel (double... angka).

Metode average menghitung rerata dari nilai-nilai yang diberikan.

* Metode average:

public double average(double... angka): Metode ini menggunakan fitur varargs di Java, yang memungkinkan metode untuk menerima sejumlah argumen yang tidak ditentukan sebelumnya.

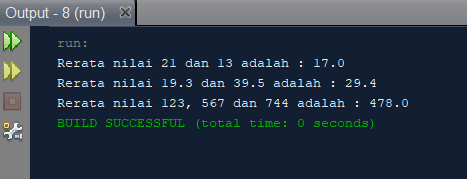
double hitung = 0;: Inisialisasi variabel hitung untuk menyimpan jumlah total nilai.

for (double Angka : angka) { hitung += Angka; }: Loop untuk menjumlahkan semua nilai yang diberikan sebagai argumen.

return hitung / angka.length;: Mengembalikan hasil pembagian total jumlah nilai dengan jumlah elemen yang diberikan (rata-rata).

* Kelas TesTugas1:

Kelas ini berisi metode main yang digunakan untuk menguji metode average dari kelas RerataNilai.



**Tugas 2:**

package pkg8;

/\*\*

\*

\* @author KAMAL

\*/

class Katak {

int umur;

String nama;

public Katak(int umur, String nama) {

this.umur = umur;

this.nama = nama;

}

public String caraBergerak() {

// Implementasi method ini dapat disesuaikan dengan cara bergerak kura-kura

return " Melompat ";

}

}

class Kecebong extends Katak {

double panjangEkor;

public Kecebong(int umur, String nama, double panjangEkor) {

super(umur, nama);

this.panjangEkor = panjangEkor;

}

// Override method caraBergerak dari class Katak

@Override

public String caraBergerak() {

// Implementasi method ini dapat disesuaikan dengan cara bergerak kecebong

return " Berenang ";

}

}

public class Main {

public static void main(String[] args) {

Katak katak = new Katak(5, "Froggy");

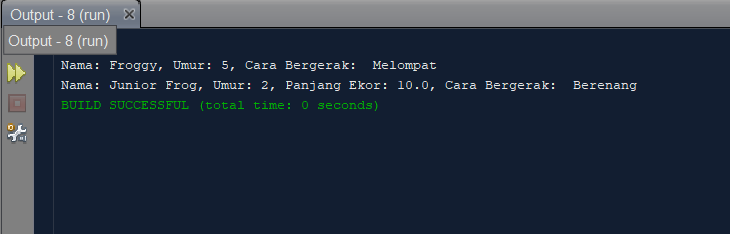
System.out.println("Nama: " + katak.nama + ", Umur: " + katak.umur + ", Cara Bergerak: " + katak.caraBergerak());

Kecebong kecebong = new Kecebong(2, "Junior Frog", 10);

System.out.println("Nama: " + kecebong.nama + ", Umur: " + kecebong.umur + ", Panjang Ekor: " + kecebong.panjangEkor + ", Cara Bergerak: " + kecebong.caraBergerak());

}

}



* Inheritance memungkinkan Kecebong untuk mewarisi atribut dan metode dari Katak.
* Overriding digunakan di Kecebong untuk memberikan implementasi spesifik dari metode caraBergerak().
* Program mencetak informasi yang tepat tentang objek katak dan kecebong, menunjukkan bagaimana mereka menggunakan metode yang di-overridden dan warisan dari superclass.

BUAT 3 CONTOH OVERLOADING DAN OVRIDDING :

OVERLOADING:

1. Source code

class bengkel {

void antrian(String nama){

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

}

void antrian(String nama, int nomor){

System.out.println("nama palanggan "+nama+" nomor antrian "+nomor);

}

void antrian(String nama, int nomor, String tipe){

System.out.println("nama palanggan "+nama+" sejumlah "+nomor+" tipe motor "+tipe);

}

void antrian(String nama, int nomor, String tipe, String kendala){

System.out.println("nama palanggan "+nama+" sejumlah "+nomor+" tipe motor "+tipe+" kerusakan "+kendala);

}

}

public class ovl{

public static void main(String[] args){

System.out.println("bengkel motor pak yon");

bengkel bs = new bengkel();

bs.antrian("anton");

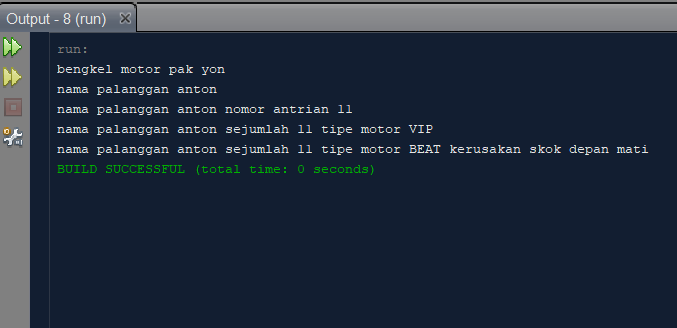
bs.antrian("anton", 11);

bs.antrian("anton", 11, "VIP");

bs.antrian("anton", 11, "BEAT", "skok depan mati");

}

}



2.

package pkg8;

/\*\*

\*

\* @author KAMAL

\*/

public class RekeningBank {

private double saldo;

public RekeningBank(double saldo) {

this.saldo = saldo;

}

public double hitungBunga(int tahun) {

return saldo \* 0.05 \* tahun;

}

public double hitungBunga(int tahun, double biayaTahunan) {

return saldo \* 0.03 \* tahun - biayaTahunan;

}

public static void main(String[] args) {

RekeningBank rekeningTabungan = new RekeningBank(10000);

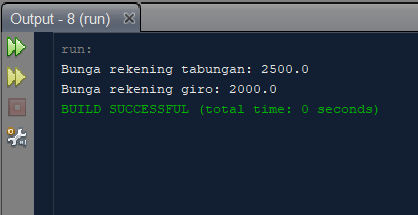
RekeningBank rekeningGiro = new RekeningBank(20000);

System.out.println("Bunga rekening tabungan: " + rekeningTabungan.hitungBunga(5));

System.out.println("Bunga rekening giro: " + rekeningGiro.hitungBunga(5, 1000));

}

}



3.

package pkg8;

/\*\*

\*

\* @author KAMAL

\*/

public class Kalkulator {

public int tambah(int a, int b) {

return a + b;

}

public double tambah(double a, double b) {

return a + b;

}

public int tambah(int a, int b, int c) {

return a + b + c;

}

public static void main(String[] args) {

Kalkulator kalkulator = new Kalkulator();

int jumlah1 = kalkulator.tambah(1, 2);

System.out.println("Jumlah dari 1 dan 2: " + jumlah1);

double jumlah2 = kalkulator.tambah(1.5, 2.3);

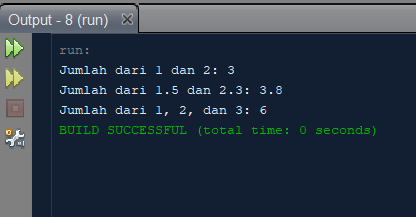
System.out.println("Jumlah dari 1.5 dan 2.3: " + jumlah2);

int jumlah3 = kalkulator.tambah(1, 2, 3);

System.out.println("Jumlah dari 1, 2, dan 3: " + jumlah3);

}

}



OVERIDING

1.

package pkg8;

/\*\*

\*

\* @author KAMAL

\*/

class KartuUno {

private String warna;

private String tipe;

private int nilai;

public KartuUno(String warna, String tipe, int nilai) {

this.warna = warna;

this.tipe = tipe;

this.nilai = nilai;

}

public KartuUno(String tipe, int nilai) {

this.warna = "tidak ada";

this.tipe = tipe;

this.nilai = nilai;

}

public KartuUno tambah(KartuUno kartu) {

if (this.tipe.equals(kartu.tipe) && this.warna.equals(kartu.warna)) {

return new KartuUno(this.warna, this.tipe, this.nilai + kartu.nilai);

} else {

throw new IllegalArgumentException("Tidak bisa menambahkan KartuUno dengan warna atau tipe yang berbeda.");

}

}

public int tambah(KartuUno kartu, int bonus) {

if (this.tipe.equals(kartu.tipe) && this.warna.equals(kartu.warna)) {

return this.nilai + kartu.nilai + bonus;

} else {

throw new IllegalArgumentException("Tidak bisa menambahkan KartuUno dengan warna atau tipe yang berbeda.");

}

}

@Override

public String toString() {

return this.warna + " " + this.tipe + " " + this.nilai;

}

}

public class Utama1 {

public static void main(String[] args) {

KartuUno k1 = new KartuUno("merah", "angka", 3);

KartuUno k2 = new KartuUno("merah", "angka", 4);

KartuUno k3 = new KartuUno("merah", "angka", 9);

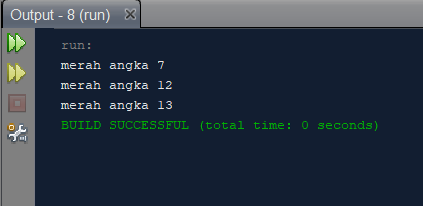
System.out.println(k1.tambah(k2));

System.out.println(k1.tambah(k3));

System.out.println(k2.tambah(k3));

}

}



2.

class Domino {

private int sisi1;

private int sisi2;

public Domino(int sisi1, int sisi2) {

this.sisi1 = sisi1;

this.sisi2 = sisi2;

}

public Domino(int sisi) {

this.sisi1 = sisi;

this.sisi2 = sisi;

}

public Domino add(Domino domino) {

return new Domino(this.sisi1 + domino.sisi1, this.sisi2 + domino.sisi2);

}

public Domino add(int side) {

return new Domino(this.sisi1 + side, this.sisi2 + side);

}

public static Domino add(Domino domino1, Domino domino2) {

return new Domino(domino1.sisi1 + domino2.sisi1, domino1.sisi2 + domino2.sisi2);

}

@Override

public String toString() {

return "[" + sisi1 + ", " + sisi2 + "]";

}

}

public class Main3 {

public static void main(String[] args) {

Domino d1 = new Domino(2, 4);

Domino d2 = new Domino(3, 5);

Domino d3 = new Domino(3);

Domino d4 = new Domino(2);

System.out.println(d1.add(d2)); // [5, 9]

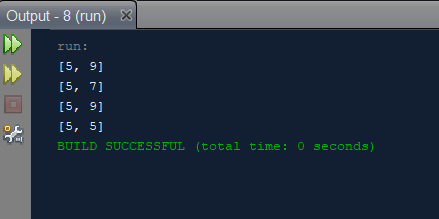
System.out.println(d1.add(3)); // [5, 7]

System.out.println(Domino.add(d1, d2)); // [5, 9]

System.out.println(d3.add(d4)); // [6, 6]

}

}



3.

package pkg8;

/\*\*

\*

\* @author KAMAL

\*/

class PemutarAudio {

public void putar(String lagu) {

System.out.println("Memutar lagu: " + lagu);

}

public void putar(String artis, String album) {

System.out.println("Memutar album: " + album + " oleh " + artis);

}

public void putar(String artis, String album, int acak) {

System.out.println("Memutar album: " + album + " oleh " + artis + " dengan acak: " + (acak > 0));

}

}

class PemutarAudioEkstra extends PemutarAudio {

@Override

public void putar(String lagu) {

System.out.println("Memutar lagu: " + lagu + " (Ekstra)");

}

@Override

public void putar(String artis, String album) {

System.out.println("Memutar album: " + album + " oleh " + artis + " (Ekstra)");

}

@Override

public void putar(String artis, String album, int acak) {

System.out.println("Memutar album: " + album + " oleh " + artis + " dengan acak: " + (acak > 0) + " (Ekstra)");

}

}

public class ovr {

public static void main(String[] args) {

PemutarAudio pemutar = new PemutarAudioEkstra();

pemutar.putar("Bohemian Rhapsody");

pemutar.putar("Queen", "A Night at the Opera");

pemutar.putar("Pink Floyd", "The Dark Side of the Moon", 1);

}

}

