Sourcecode:

Percobaan1:

public class Parenttt {

public int x = 5;

}

public class childd extends Parenttt{

public int x = 10;

public void info(int x){

System.out.println("Nilai x sebagai parameter = " + x);

System.out.println("Data member x di Class childd = "+this.x);

System.out.println("Data member x di Class Parenttt = "+super.x);

}

}

public class NilaiX{

public static void main(String args[]){

childd tes = new childd();

tes.info(20);

}

}

Penjelasan:

Superclass ada pada class parentt, subclass ada pada class childd dan NilaiX

Percobaan 2

Superclass ada di class pegawai:

public class Pegawai {

String nama;

public double gaji;

}

Subclass ada di class manajer dan menggunakan metode override

public class Manajer extends Pegawai {

public String department;

public void IsiData(String n, String d){

nama=n;

String departmen = d;

}

}

Percobaan 3

Pada percobaan 3 terdapat eror dan penyebabnya saya sudah memperbaikinya

public class Parentt {

//Kosong

}

public class child extends Parentt{

int x;

public child(){

x = 5;

}

}

Percobaan 4

Pada percobaan 4 ada kesalahan yang tidak bisa di run, kesalahan itu ada pada java.import

import java.util.Date;

public class Employee {

private static final double BASE\_SALARY = 15000.00;

private String Name ="";

private double Salary = 0.0;

private Date birthDate;

public Employee(){}

public Employee(String name, double salary, Date DoB){

this.Name=name;

this.Salary=salary;

this.birthDate=DoB;

}

public Employee(String name, double salary){

this(name,salary,null);

}

public Employee(String name, Date DoB){

this(name,BASE\_SALARY,DoB);

}

public Employee(String name){

this(name,BASE\_SALARY);

}

public String GetName(){return Name;}

public double GetSalary(){return Salary;}

}

Subclass

import java.util.Date;

public class Employee {

private static final double BASE\_SALARY = 15000.00;

private String Name ="";

private double Salary = 0.0;

private Date birthDate;

public Employee(){}

public Employee(String name, double salary, Date DoB){

this.Name=name;

this.Salary=salary;

this.birthDate=DoB;

}

public Employee(String name, double salary){

this(name,salary,null);

}

public Employee(String name, Date DoB){

this(name,BASE\_SALARY,DoB);

}

public Employee(String name){

this(name,BASE\_SALARY);

}

public String GetName(){return Name;}

public double GetSalary(){return Salary;}

}

Subclass

public class TestManager {

public static void main(String[] args){

Manager Utama = new Manager("Jhon",5000000,"Financial");

System.out.println("Name:"+ Utama.GetName());

System.out.println("Salary:"+ Utama.GetSalary());

System.out.println("Department:"+ Utama.GetDept());

Utama = new Manager("Michael", "Accounting");

System.out.println("Name:"+ Utama.GetName());

System.out.println("Salary:"+ Utama.GetSalary());

System.out.println(Utama.GetDept()+ "Department:");

}

}

Percobaan 5

Superclass

public class MoodyObject {

protected String getMood(){

return"moody";

}

public void speak(){

System.out.println("I am"+getMood());

}

void laugh(){}

void cry(){}

}

Subclass

Menggunkaan metode override

public class SadObject extends MoodyObject {

@Override

protected String getMood(){

return "sad";

}

@Override

public void cry(){

System.out.println("Hoo hoo");

}

}

Subclass

public class MoodyTest {

public static void main(String[]args){

MoodyObject m = new MoodyObject();

//test parent class

m.speak();

//test inheritance class

m = new HappyObject();

m.speak();

m.cry();

//test inheritance class

m = new SadObject();

m.speak();

m.cry();

}

}

Percobaan 6

Superclass

public class A {

String var\_a = "Variabel A";

String var\_b = "Variabel B";

String var\_c = "Variabel C";

String var\_d = "Variabel D";

A(){

System.out.println("Konstruktor A dijalankan");

}

}

Subclass

Menggunakan metode override

public class B extends A {

B(){

System.out.println("Konstruktor B dijalankan");

var\_a = "Var\_a dari class B";

var\_b = "Var\_b dari class B";

}

public static void main(String args[]){

System.out.println("Object A dibuat");

A aa= new A();

System.out.println("menampilkan nama variabel obyek aa");

System.out.println("aa.var\_a");

System.out.println("aa.var\_b");

System.out.println("aa.var\_c");

System.out.println("aa.var\_d");

System.out.println("");

System.out.println("Object B dibuat");

B bb= new B();

System.out.println("menampilkan nama variabel obyek bb");

System.out.println("bb.var\_a");

System.out.println("bb.var\_b");

System.out.println("bb.var\_c");

System.out.println("bb.var\_d");

}

}

Subclass

Percobaan7

Superclass

public class Bapak {

int a;

int b;

void show\_variable(){

System.out.println("Nilai a="+ a);

System.out.println("Nilai b="+ b);

}

}

Subclass

public class Anak extends Bapak {

int c;

void show\_variabel(){

System.out.println("Nilai a="+ a);

System.out.println("Nilai b="+ b);

System.out.println("Nilai c="+ c);

}

}

Subclass

public class InheritExample {

public static void main(String[] args){

Bapak objectBapak = new Bapak();

Anak objectAnak = new Anak();

objectBapak.a=1;

objectBapak.b=1;

System.out.println("Object Bapak (Superclass):");

objectBapak.show\_variable();

objectAnak.c=5;

System.out.println("Object Anak (Superclass dari Bapak ):");

objectAnak.show\_variabel();

}

}