

JAWABAN TUGAS PERCOBAAN

1. Jalankan code program diatas dan benahi jika menemukan kesalahan!

Jawaban :

Tidak ditemukan kesalahan

2. Bagaimana cara konstruktor pada subclass memanggil konstruktor di superclass nya? Apakah hal itu perlu dilakukan? Sertakan alasan anda !

Jawaban :

Untuk memanggil konstruktor dari superclass dalam subclass, dapat menggunakan kata kunci `super()` di dalam konstruktor subclass. Ini harus dilakukan jika superclass memiliki konstruktor yang mengambil argumen, karena konstruktor superclass perlu diinisialisasi dengan nilai-nilai tersebut.

3. Tambahkan constructor pada class Employee dengan parameter *String name*! amati perubahan apa yang terjadi, jelaskan jawaban anda!

Jawaban :

Kode tersebut hanya menambahkan konstruktor baru pada kelas Employee yang mengambil satu parameter String name. Ini tidak akan mempengaruhi konstruktor yang telah ada sebelumnya, yang masih tetap tersedia untuk digunakan.

4. Pada Class Manager baris ke 5, setelah variable day tambahkan variable bonus! Amati apa yang terjadi dan mengapa demikian?

Jawaban :

Menambahkan variabel bonus setelah variabel day pada kelas Manager menyebabkan kesalahan kompilasi karena konstruktor Manager membutuhkan argumen tambahan yang tidak disediakan saat membuat objek Manager di MainEmployee

5. Untuk apa digunakan keyword this pada class manager dan employee? Hapus keyword this dan amati apa yang terjadi?

Jawaban :

- Pada class Manajer Jika kita menghapus keyword this pada baris `this.bonus = bonus;` dalam method `setBonus`, maka kode akan tetap berfungsi dengan baik karena tidak ada variabel lokal atau parameter yang bernama bonus. Namun, menggunakan this di sini membantu menjelaskan bahwa kita merujuk pada variabel instance bonus.
- Pada class Employee, penggunaan this pada baris `this.name = name;`, `this.salary = salary;`, dan `this.hireday = calendar.getTime();` sangat penting karena variabel lokal name, salary, dan hireday memiliki nama yang sama dengan variabel instance di class Employee. Tanpa menggunakan this, Java akan bingung dan

memperlakukan name, salary, dan hireday sebagai variabel lokal, bukan variabel instance.

6. Tambahkan constructor pada class Employee dengan parameter Bertipe data string bernama name yang nantinya bila constructor ini akan dipanggil akan menginisialisasi variable name! Amati perubahannya pada class anak dan jelaskan! Benahi bila terjadi kesalahan!

Jawaban :

Penambahan konstruktor baru pada kelas Employee memungkinkan inisialisasi objek hanya dengan memberikan nilai untuk nama, tanpa mempengaruhi kelas anak Manager, karena kelas Manager akan secara otomatis mewarisi konstruktor dari kelas Employee.

7. Pada bab sebelumnya anda telah belajar mengenai konsep encapsulation, jelaskan mengapa pada super class menggunakan modifier protected? Apa yang terjadi jika modifier anda ubah menjadi private atau public? Jelaskan !

Jawaban :

Pada super class, menggunakan modifier protected memungkinkan akses ke atribut atau metode tersebut dari kelas anak (subclass), serta membatasi akses dari kelas di luar paket (package).

Jika modifier diubah menjadi private, atribut atau metode tersebut hanya dapat diakses di dalam kelas itu sendiri, dan tidak dapat diakses dari kelas anak (subclass). Ini akan melanggar prinsip warisan (inheritance), di mana kelas anak harus dapat mengakses atribut atau metode yang dideklarasikan di kelas induk (super class).

Jika modifier diubah menjadi public, atribut atau metode tersebut dapat diakses dari mana saja, baik dari dalam kelas itu sendiri, dari kelas anak (subclass), maupun dari kelas di luar paket. Ini mengurangi tingkat kontrol dan isolasi data, yang dapat menyebabkan masalah dalam memelihara dan mengelola kode dalam skala yang lebih besar. Ini melanggar prinsip encapsulation, di mana akses langsung ke atribut atau metode dari luar kelas sebaiknya dibatasi untuk mencegah modifikasi yang tidak diinginkan atau efek samping yang tidak terduga.

Jadi, menggunakan modifier protected pada super class mempertahankan tingkat akses yang sesuai dengan prinsip encapsulation, memungkinkan warisan yang benar, sambil memberikan kontrol akses yang cukup untuk subclass.

8. Ubahlah acces modifier method pada kelas employee menjadi :
- Private
 - Protected

Amati perubahan apa yang terjadi? Jelaskan jawaban anda dengan detail!

Jawaban :

- Jika akses modifier method pada kelas Employee diubah menjadi private, maka method tersebut hanya dapat diakses di dalam kelas Employee itu sendiri. Artinya, kelas lain atau kelas anak tidak akan dapat mengakses method tersebut. Ini berarti bahwa method tersebut tidak akan dapat diwarisi oleh kelas anak, dan hanya dapat digunakan secara internal di dalam kelas Employee itu sendiri.
- Jika akses modifier method pada kelas Employee diubah menjadi protected, maka method tersebut dapat diakses dari kelas anak (subclass), serta di dalam paket yang sama. Artinya, kelas anak dapat mewarisi method tersebut dan menggunakannya. Namun, kelas di luar paket tidak akan dapat mengakses method tersebut, kecuali jika mereka adalah kelas anak dari Employee atau berada di paket yang sama.