Nama : Amelinda Renjani

NIM : 20220040163

Kelas : TI 22 A

TUGAS PEMROGRAMAN BERBASIS PLATFORM

SESI 3

1. Percobaan 1

Analisis:

- this.x dalam metode Info merujuk pada nilai x dari objek saat ini, yang merupakan objek Child yang baru dibuat.
- Saat memanggil super.x dalam metode Info, itu merujuk pada nilai x dari class induk, yaitu class Parent.
- Variabel x dalam metode Info, yang merupakan parameter metode, akan mengambil nilai yang diteruskan saat metode dipanggil, dalam hal ini, 20.

2. Percobaan 2

Alasan kode program error beserta solusinya:

Dalam kode yang diberikan, terdapat error karena class Manajer mencoba mengakses variabel nama yang dideklarasikan sebagai private di class Pegawai. Dalam Java, ketika suatu variabel dideklarasikan sebagai private, itu hanya dapat diakses langsung oleh class yang mendefinisikannya. Class lain tidak dapat mengaksesnya secara langsung.

Untuk mengatasi error tersebut, variabel nama di class Pegawai diakses menggunakan metode getNama() dan diatur nilainya menggunakan metode setNama(String nama). Dalam class Manajer, metode isiData(String n, String d) digunakan untuk mengatur nilai nama dengan menggunakan setter yang disediakan oleh class Pegawai.

Hasil Analisis:

- Kode tersebut menunjukkan penerapan konsep pewarisan (inheritance) di mana class Manajer mewarisi sifat-sifat (variabel dan metode) dari class Pegawai.
- Konsep enkapsulasi diterapkan dengan mendeklarasikan variabel nama sebagai private, sehingga hanya dapat diakses melalui metode getter dan setter yang telah disediakan.
- Class Manajer menambahkan fitur tambahan dalam bentuk variabel anggota departemen, yang tidak dimiliki oleh class Pegawai.
- Metode isiData(String n, String d) digunakan untuk mengisi data sekaligus mengatur nilai nama dan departemen dari objek class Manajer.

3. **Percobaan 3:**

   Alasan kode program error dan solusinya:

   Dalam percobaan ketiga ini, terjadi error karena ketika membuat sebuah class turunan (Child) dari class induk (Parent), Java akan mencoba untuk memanggil konstruktor dari class induk (Parent2). Namun, dalam class Parent2 tidak ada konstruktor yang didefinisikan, dan itu menyebabkan error. Solusi untuk mengatasi error ini adalah dengan menambahkan konstruktor public Parent2 di class Parent2.

   Analisis:

   - Class Parent2 merupakan class dasar atau induk. Meskipun dalam contoh ini class tersebut kosong, namun secara konseptual, ini bisa berisi atribut atau metode yang akan diwarisi oleh class anaknya.
   - Class Child2 adalah turunan dari class Parent2. Ini ditunjukkan dengan penggunaan kata kunci extends di deklarasi class. Dalam hal ini, class Child2 mewarisi semua sifat dari class Parent2.
   - Variabel x adalah variabel anggota (instance variable) class Child2, yang memiliki nilai awal 5. Variabel ini akan dimiliki oleh setiap objek class Child2 yang dibuat.
   - Konstruktor class Child2 adalah metode khusus yang memiliki nama yang sama dengan nama classnya. Dalam konstruktor ini, variabel x diinisialisasi dengan nilai 5.

4. **Percobaan 4**

   Analisis :

   - Kode tersebut menunjukkan penggunaan konsep pewarisan (inheritance) di mana class Manager mewarisi sifat dan perilaku dari class Employee.
   - Konstruktor dalam class Manager memanfaatkan konstruktor class Employee menggunakan kata kunci super().
   - Setiap konstruktor memiliki fungsi yang berbeda sesuai dengan jumlah parameter yang diberikan dan nilai-nilai yang ingin diinisialisasi.
   - Kode tersebut juga menunjukkan penggunaan overloading konstruktor untuk memberikan fleksibilitas dalam pembuatan objek.

5. **Percobaan 5**

   Analisis:

   - Kode tersebut mengimplementasikan konsep pewarisan (inheritance), di mana class turunan (SadObject dan HappyObject) mewarisi sifat dan perilaku dari class induk (MoodyObject).
   - Polimorfisme juga ditunjukkan di sini, di mana objek m dari tipe MoodyObject dapat merujuk ke objek dari class turunan yang berbeda (SadObject dan HappyObject),

dan perilaku metode yang dipanggil akan bergantung pada objek sebenarnya yang dipegang oleh variabel referensi m.

- Method overriding terjadi di class turunan (SadObject dan HappyObject) di mana metode getMood() dioverride untuk menghasilkan mood yang sesuai.
- Implementasi dari metode laugh() dan cry() berbeda di setiap class turunan, menunjukkan fleksibilitas dalam implementasi yang sesuai dengan kebutuhan class tersebut.

6. Percobaan 6

Analisis:

- Kode tersebut menggambarkan konsep pewarisan (inheritance) di mana class B mewarisi sifat dan perilaku dari class A.
- Konstruktor class A akan dieksekusi terlebih dahulu saat membuat objek class A maupun class B.
- Penggunaan super() dalam konstruktor class B memungkinkan untuk memanggil konstruktor class induk sebelum melakukan inisialisasi class turunan.
- Setelah konstruktor class B dieksekusi, nilai dari variabel anggota var_a dan var_b diubah, menunjukkan bahwa class turunan dapat mengakses dan memanipulasi variabel anggota dari class induk.
- Kode tersebut memberikan contoh sederhana tentang bagaimana pewarisan bekerja dalam Java dan bagaimana konstruktor class turunan dapat memanggil konstruktor class induk.

7. Percobaan 7

Analisis:

- Kode tersebut mengilustrasikan konsep pewarisan, di mana class Anak mewarisi sifat dan perilaku dari class Bapak.
- Metode show_variabel() di class Anak melakukan overriding terhadap metode yang sama di class Bapak. Ini artinya, metode yang digunakan adalah milik class Anak, bukan class Bapak.
- Override method memungkinkan class turunan untuk memberikan implementasi yang berbeda terhadap metode yang sudah didefinisikan di class induknya.
- Ketika metode show_variabel() dipanggil pada objek objectAnak, yang dieksekusi adalah metode yang ada di class Anak, sehingga mencetak nilai dari variabel a, b, dan c.

8. Percobaan 8

Analisis:

- Kode tersebut menunjukkan konsep pewarisan (inheritance) dalam pemrograman berorientasi objek (OOP) di Java, di mana class Baby mewarisi sifat dan perilaku dari class Parent3.
- Konstruktor Baby memanggil konstruktor superclass Parent3 menggunakan kata kunci super(), sehingga konstruktor dari class Parent3 akan dieksekusi terlebih dahulu sebelum konstruktor dari class Baby.
- Dengan demikian, penggunaan super() memungkinkan untuk memanggil konstruktor superclass dan melakukan inisialisasi dari class induk sebelum melakukan inisialisasi dari class turunannya.
- Override method tidak terjadi dalam contoh tersebut karena tidak ada method dalam class Parent3 yang di-overriding oleh class Baby.
- Namun, jika ingin melakukan overriding method dapat menambahkan method di class Parent3 yang kemudian di-override oleh class Baby.