



NAMA : OKTAVIAN VANDI TRI SHAKTI

NIM : 2041720208

KELAS : 2C

LAPORAN JOBSHEET 11

PERTANYAAN PERCOBAAN 1

1. Class apa sajakah yang merupakan turunan dari class Employee?

jawab :

terdapat 2 class yaitu `internshipemployee` dan `permanentemployee`.

2. Class apa sajakah yang implements ke interface Payable?

jawab :

terdapat 2 class yaitu `electricitybill` dan `permanentemployee`.

3. Perhatikan class `Tester1`, baris ke-10 dan 11. Mengapa `e`, bisa diisi dengan objek `pEmp` (merupakan objek dari class `PermanentEmployee`) dan objek `iEmp` (merupakan objek dari class `InternshipEmployee`) ?

jawab :

karena `employee` memiliki turunan class `internshipemployee` dan `permanentemployee` sehingga bisa di isi dengan objek tersebut.

4. Perhatikan class `Tester1`, baris ke-12 dan 13. Mengapa `p`, bisa diisi dengan objek `pEmp` (merupakan objek dari class `PermanentEmployee`) dan objek `eBill` (merupakan objek dari class `ElectricityBill`) ?

jawab :

karena interface `payable` digunakan di 2 class tersebut sehingga `p` bisa di isi object dari class `Permanentemployee` dan `electricitybill`.

5. Coba tambahkan sintaks:

`p = iEmp;`

`e = eBill;`

pada baris 14 dan 15 (baris terakhir dalam method `main`) ! Apa yang menyebabkan error?

jawab :

karena `p` harus memakai interface `payable` sedangkan `employee` tidak menggunakan interface tersebut dan untuk `e` error di sebabkan `employee` tidak mewarisi class `electricitybill`.

6. Ambil kesimpulan tentang konsep/bentuk dasar polimorfisme!

jawab :

ketika ada suatu objek yang dideklarasikan dari super class, maka objek tersebut bisa diinstansiasi sebagai objek dari sub class.

PERTANYAAN PERCOBAAN 2

1. Perhatikan class `Tester2` di atas, mengapa pemanggilan `e.getEmployeeInfo()` pada baris 8 dan `pEmp.getEmployeeInfo()` pada baris 10 menghasilkan hasil sama?



NAMA : OKTAVIAN VANDI TRI SHAKTI

NIM : 2041720208

KELAS : 2C

jawab :

karena employee 2 sudah di instansiasikan objek ke pEmp atau permanent employe sehingga pemanggilan sama dengan pEmp.

2. Mengapa pemanggilan method `e.getEmployeeInfo()` disebut sebagai pemanggilan method virtual (virtual method invocation), sedangkan `pEmp.getEmployeeInfo()` tidak?

jawab :

karena `e.getEmployeeinfo()` memanggil overriding method dari suatu objek polimorfisme dari class `permanentemploye`.

3. Jadi apakah yang dimaksud dari virtual method invocation? Mengapa disebut virtual?

jawab :

Disebut virtual karena antara method yang dikenali oleh compiler dan method yang dijalankan oleh JVM berbeda.

PERTANYAAN PERCOBAAN 3

1. Perhatikan array `e` pada baris ke-8, mengapa ia bisa diisi dengan objekobjek dengan tipe yang berbeda, yaitu objek `pEmp` (objek dari `PermanentEmployee`) dan objek `iEmp` (objek dari `InternshipEmployee`) ?

jawab :

karena `employee` mewarisi class `permanentemploye` dan `intershipemployee` sehingga bisa dimasukkan ke dalam array sesuai index yang di inginkan.

2. Perhatikan juga baris ke-9, mengapa array `p` juga diisi dengan objek-objek dengan tipe yang berbeda, yaitu objek `pEmp` (objek dari `PermanentEmployee`) dan objek `eBill` (objek dari `ElectricityBilling`) ?

jawab :

karena `payable` mewarisi class `electricitybill` dan `permanentemployee` sehingga bisa dimasukkan ke dalam array sesuai index yang di inginkan.

3. Perhatikan baris ke-10, mengapa terjadi error?

jawab :

karena `ebill` dari class `electricitybill` tidak mewarisi class dari `employe` sehingga tidak bisa di instansiasikan objek.

PERTANYAAN PERCOBAAN 4

1. Perhatikan class `Tester4` baris ke-7 dan baris ke-11, mengapa pemanggilan `ow.pay(eBill)` dan `ow.pay(pEmp)` bisa dilakukan, padahal jika diperhatikan method `pay()` yang ada di dalam class `Owner` memiliki argument/parameter bertipe `Payable`?



NAMA : OKTAVIAN VANDI TRI SHAKTI

NIM : 2041720208

KELAS : 2C

jawab :

Jika diperhatikan lebih detil eBill merupakan objek dari ElectricityBill dan pEmp merupakan objek dari PermanentEmployee? tetap bisa dilakukan karena interface payable digunakan diclass electricitybill dan permanentemployee.

2. Jadi apakah tujuan membuat argument bertipe Payable pada method pay() yang ada di dalam class Owner?

jawab :

agar bisa diberikan nilai class yang menggunakan interface dari payable.

3. Coba pada baris terakhir method main() yang ada di dalam class Tester4 ditambahkan perintah ow.pay(iEmp); Mengapa terjadi error?

jawab :

karena ow.pay memiliki parameter payable dan intershipemploye tidak menggunakan interface payable sehingga tidak bisa di instasiasikan objek.

4. Perhatikan class Owner, diperlukan untuk apakah sintaks p instanceof ElectricityBill pada baris ke-6 ?

jawab :

p digunakan untuk mengecek apakah suatu objek merupakan hasil instansiasi dari suatu class tertentu.

5. Perhatikan kembali class Owner baris ke-7, untuk apakah casting objek disana (ElectricityBill eb = (ElectricityBill) p) diperlukan ? Mengapa objek p yang bertipe Payable harus di-casting ke dalam objek eb yang bertipe ElectricityBill ?

jawab :

digunakan untuk mengubah tipe dari suatu objek eb class electricitybill ke p payable. Jika ada suatu objek dari subclass kemudian tipenya diubah ke superclass, maka hal ini termasuk ke upcasting.

TUGAS

```
Walking Zombie Data
Health : 100
Level : 1

Jumping Zombie Data
Health : 100
Level : 2

Barrier Strenght: 100

-----
Walking Zombie Data
Health : 42
Level : 1

Jumping Zombie Data
Health : 66
Level : 2

Barrier Strenght: 64
```