Nama : Yulia Eka Ardhani
Kelas : TI 2C
Jobsheet 11 POLIMORFISME

4.2. Pertanyaan

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

Jawab : class InternshipEmployee dan PermanentEmployee

2. Class apa sajakah yang implements ke interface Payable?

Jawab : class PermanentEmployee dan class ElectricityBill

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 InternshipEmploye) ?

Jawab : karena class PermanentEmployee dan class InternshipEmploye merupakan class turunan dari class Employee

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

Jawab : karena class PermanentEmployee dan class ElectricityBill mengimplementasikan class interface Payable

5. Coba tambahkan sintaks:

p = iEmp;

e = eBill;

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

Jawab : karena class InternshipEmployee tidak mengimplementskan class interface Payable, dan class ElectricityBill tidak mengextends kan class Employee

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. Dari uraian tersebut bisa dilihat bahwa konsep polimorfisme bisa diterapkan pada class-class yang memiliki relasi inheritance (relasi generalisasi atau IS-A). Selain pada class-class yang memiliki relasi inheritance, polimorfisme juga bisa diterapkan pada interface. Ketika ada objek yang dideklarasikan dari suatu interface, maka ia bisa digunakan untuk mereferensi ke objek dari class-class yang implements ke interface tersebut.

5.2. Pertanyaan

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

Jawab : karena sama-sama memanggil info yang ada di class PermanentEmployee. Dan Untuk yang baris 8 menggunakan pemanggilan method virtual.

Nama : Yulia Eka Ardhani
Kelas : TI 2C
Jobsheet 11 POLIMORFISME

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

Jawab : Karena Employee diinisialisasikan dengan e, dan e = pEmp (pEmp extends e) yang dimana Employee memanggil instansiasi pEmp dari objek PermanentEmployee. Kalau pEmp.getEmployeeInfo(), inisialisasi objek PermanentEmploye langsung memanggil getEmployeeInfo()

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

Jawab : Virtual method invocation terjadi ketika ada pemanggilan overriding method dari suatu objek polimorfisme. Disebut virtual karena antara method yang dikenali oleh compiler dan method yang dijalankan oleh JVM berbeda.

6.2. Pertanyaan

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 e adalah Employee, dimana objek pEmp (objek dari PermanentEmployee) dan objek iEmp (objek dari InternshipEmployee) merupakan extends dari class Employee.

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

Jawab : karena p adalah interface Payable, dimana objek pEmp (objek dari PermanentEmployee) dan objek eBill (objek dari ElectricityBilling) telah mengimplementasikan class interface Payable.

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

Jawab : karena eBill (objek dari ElectricityBill) tidak mengextendskan class Employee, jadi saat Employee mau memanggil ebill terjadi error, dan class ElectricityBill harus extends Employee

7.2. Pertanyaan

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? Jika diperhatikan lebih detil eBill merupakan objek dari ElectricityBill dan pEmp merupakan objek dari PermanentEmployee?

Jawab : karena pada method pay() memang berparameter bertipe Payable p, yang dimana ada syntax jika p instanceof ElectricityBill maka akan menampilkan getBillInfo();, atau jika p instanceof PermanentEmployee maka akan menampilkan getEmployeeInfo();

Pernyataan instanceof sangat berguna untuk mengetahui tipe asal dari suatu polymorphic arguments.

Nama : Yulia Eka Ardhani
Kelas : TI 2C
Jobsheet 11 POLIMORFISME

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

Jawab : sebagai instansiasi class tertentu, seperti contonya dari kedua class tersebut.

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

```
3   public class Tester4 {
4       public static void main(String[] args) {
5           Owner ow = new Owner();
6           ElectricityBill eBill = new ElectricityBill(5, "R-1");
7           ow.pay(eBill);//pay for electricity bill
8           System.out.println("--------------------------------");
9
10          PermanentEmployee pEmp = new PermanentEmployee("Dedik", 500);
11          ow.pay(pEmp);//pay for permanent employee
12          System.out.println("--------------------------------");
13
14          InternshipEmployee iEmp = new InternshipEmployee("Sunarto", 5);
15          ow.showMyEmployee(pEmp);//show permanent employee info
16          System.out.println("--------------------------------");
17          ow.showMyEmployee(iEmp);//show internship employee info
18
            ow.pay(iEmp);
20      }
21  }
```

Mengapa terjadi error?

Jawab : karena class InternshipEmployee tidak didefinisikan di instanceof pada class Owner

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

Jawab : sebagai instansiasi class ElectricityBill, Pernyataan instanceof sangat berguna untuk mengetahui tipe asal dari suatu polymorphic arguments.

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 : karena penggunaan instanceof selalu diikuti dengan casting object dari parameter ke tipe asalnya.