Praktek Pertemuan 6

Disusun untuk memenuhi tugas mata kuliah Pemrograman Berorientasi Objek (Praktek)

Disusun Oleh:
Radja Restu Arsita (231511061) 2B – D3

Jurusan Teknik Komputer dan Informatika

Politeknik Negeri Bandung

2024

DAFTAR ISI

PEMBAHASAN

1.1 Modify class circle

a. Output

```
Circle[radius=1.0 color=red]
Circle[radius=20.0 color=Blue]
Circle[radius=20.0 color=Hijau telor asin]
```

b. Penjelasan

1) Output pertama merupakan output dari constructor default circle

2) Output keuda merupakan output dari constructor yang diminta untuk dibuat yaitu untuk menambahkan argumen radius dan color

3) Saya menggunakan setter color untuk mengubah warna nya sesuai dengan perintah yang diminta untuk membuat setter untuk color

1.2 Overriding the getArea() method

a. Output

```
Cylinder: radius=1.0 height=1.0 base area=12.566370614359172 volume=3.141592653589793
Cylinder: radius=1.0 height=10.0 base area=69.11503837897544 volume=31.41592653589793
Cylinder: radius=2.0 height=10.0 base area=150.79644737231007 volume=125.66370614359172
```

b. Penjelasan

Disini ada 3 contoh output yaitu

1) Menggunakan default constructor yang getArea nya sudah di override menggunakan rumus getArea untuk cylinder dengan memanggil super.getArea() sebagai variable untuk base-area

2) Menggunakan consturctor ke 2 yang mana ada argumen height yang harus di isi dan menggunakan rumus getArea untuk cylinder dengan memanggil super.getArea() sebagai variable untuk base-area

3) Menggunakan constructor yang ketiga yang mana ada argumen radius dan height untuk diisi dan menggunakan rumus getArea untuk cylinder dengan memanggil super.getArea() sebagai variable untuk base-area

Selain itu sesuai perintah saya mengubah getVolume() dengan metode override untuk memanggil super.getArea() dari circle sehingga bisa digunakan untuk mendapat kan volume dari silinder

1.3 Provide a toString() method

a. Output

```
Cylinder: subclass of Circle[radius=2.0 color=red] height=10.0
```

b. Penjelasan

Saya menggunakan override untuk method toString yang berada di cylinder sehingga bisa mengubah nya sesuai dengan perintah yang mana saya memasukan format yang diminta terlebih dahulu setelah itu saya memanggil super.toString() untuk mendapatkan format toString dari circle

2.1 Superclass Shape

a. Output

```
A Shape with color of red and filled
A Circle with radius = 1.0, whicch is a subclass of A Shape with color of red and filled
A Rectangle with width = 1.0 And length = 1.0 which is a subclass of A Shape with color of red and filled
A Square with side = 4.0, which is a subclass of A Rectangle with width = 4.0 And length = 4.0 which is a subclass of A Shape with color of red and filled
```

b. Penjelasan

Pada nomor 2 ini yang saya tangkap itu intinya adalah penerapan inheritance yang memiliki ketergantungan pada method toString() yang mana setiap class harus override toString dari super class nya masing-masing untuk mengubah output toString sesuai dengan class nya selain itu, disini juga memperjelas mengenai constructor itu bisa lebih dari satu untuk setiap class nya menyesuaikan dengan kebutuhan disini saya menampilkan 4 output :

1) Ouput toString dari Shape yang mana merupakan original dari toString itu sendiri

2) Output toString dari Circle yang mana merupakan override dari shape yang ditambah dengan format nilai radius di belakang nya

3) Output toString dari rectangle yang mana merupakan override dari shape yang ditambah

dengan format nilai width dan length nya

4) Output toString dari square yang mana merupakan override dari rectangle yang mana format toString dari rectangle sudah diubah yang sudah dijelaskan sebelum nya sehingga secara tidak langsung output dari toString square menampilkan informasi mengenai square ,rectangle dan shape

3. Multiple inheritance [Case 1]

a. Output

```
SEBELUM DI SORT :
Antonio Rossi 3150000.0 1989
Maria Bianchi 4200000.0 1991
Isabel Vidal 1050000.0 1993
SETELAH DI SORT :
Isabel Vidal 1050000.0 1993
Antonio Rossi 3150000.0 1989
Maria Bianchi 4200000.0 1991
```

b. Penjelasan

Disini saya mengubah input untuk constructor nya agar terlihat hasil sorting nya yang dimana employee mengimplementkan method dari sortable

4. Multiple inheritance [Case 2]

a. Output

```
SEBELUM DI SORT :
Antonio Rossi 2100000.0 1989
Maria Bianchi 4252500.0 1991
Isabel Vidal 3780000.0 1993
SETELAH DI SORT :
Antonio Rossi 2100000.0 1989
Isabel Vidal 3780000.0 1993
Maria Bianchi 4252500.0 1991
```

b. Penjelasan

Untunk membuat manager memiliki multiple inheritance kita bisa menggunakan implements

LESSON LEARNED

Dari tugas praktek PBO week 6 ini saya belajar mengimplementasikan penggunaan constructor yang mana sesuai dengan penjelasan yang sudah dijelaskan pada kelas teori bahwa constructor bisa dimiliki oleh setiap class nya masing-masing lebih dari satu disini saya mempraktek kan apa yang sudah di jelaskan di teori selain itu, saya mempelajari bagaimana cara untuk melakukan overriding dari subclass terhadap method yang ada di superclass nya lalu satu lagi yang saya pelajari di praktikum week 6 ini saya mempelajari bagaimana membuat multiple-inheritance pada java yang mana java tidak mensupport penggunaan muliple-inheritance namun masih bisa dilakukan dengan menggunakan interface-implement.

REFERENSI

GeeksforGeeks. (29 September 2024). *Antarmuka dalam Java*. Diakses dari https://www.geeksforgeeks.org/interfaces-in-java/

GeeksforGeeks. (29 September 2024). *Override dalam Java*. Diakses dari https://www.geeksforgeeks.org/overriding-in-java/

LAMPIRAN

Link github : https://github.com/Radja-Restu-A/Object-orianted-programming/tree/main/Pertemuan%205%20(P)