

Tugas 7 : Polymorphism
Pemrograman Berorientasi Objek



Nama : Agista Diva Briliani

NIM : 221511038

Prodi : D3 Teknik Informatika

Kelas : 2B JTK

PROGRAM STUDI DIII TEKNIK INFORMATIKA
POLITEKNIK NEGERI BANDUNG

Hasil Akhir Program	Keterangan
<p>Studi Kasus 1 :</p> <p>- Commission.java</p> <pre>11 public class Commission extends Staff { 12 13 private double total_sales; 14 private double commission_rate; 15 16 public Commission(String sName, String sAddress, String sPhone, String sSocialNumber, double rate, double commission_rate) { 17 super(sName, sAddress, sPhone, sSocialNumber, rate); 18 this.commission_rate = commission_rate; 19 } 20 21 22 public void addSales(double totalSales) { 23 this.total_sales = totalSales; 24 } 25 26 public double pay() { 27 double payment = super.pay() + (total_sales * commission_rate); 28 total_sales = 0; 29 return payment; 30 } 31 32 public String toString() { 33 String result = super.toString(); 34 result += "\nTotal Sales: " + total_sales; 35 return result; 36 } 37 } 38 39 }</pre> <p>- Staff.java</p> <pre>11 public class Staff { 12 staff[member] staffList; 13 14 public Staff() { 15 staffList = new Staff[member]; 16 17 staffList[0] = new Executive ("Norm", "123 Main Lane", "555-8374", "123-45-6789", 2403.07); 18 staffList[1] = new Employee ("Cliff", "456 Off 1 Ave", "555-8374", "987-65-4321", 1246.5); 19 staffList[2] = new Employee ("Mindy", "789 Off Road", "555-8374", "101-23-3456", 1180.23); 20 staffList[3] = new Hourly ("John", "4567 E.C. Ave.", "555-8374", "987-65-4321", 10.55); 21 staffList[4] = new Volunteer ("Norm", "123 Main Lane", "555-8374", "123-45-6789", 0); 22 staffList[5] = new Volunteer ("Mindy", "4567 Off Road", "555-8374", "987-65-4321", 0); 23 24 staffList[6] = new Commission ("Agista", "Bandung", "1234567890", "012-345-6789", 7.25, 400.0, 35); 25 staffList[7] = new Commission ("Agista", "Bandung", "1234567890", "012-345-6789", 8.75, 950.0, 40); 26 27 ((Executive) staffList[0]).searchName ("Norm"); 28 ((Hourly) staffList[3]).addHours (1000.0); 29 30 ((Commission) staffList[6]).addSales (totalSales); 31 ((Commission) staffList[7]).addSales (totalSales); 32 ((Commission) staffList[7]).addSales (totalSales); 33 } 34 35 public void payday() { 36 double amount; 37 for (int count = 0; count < staffList.length; count++) { 38 amount = staffList[count].pay(); 39 System.out.println(staffList[count]); 40 } 41 } 42 }</pre>	<p>Output :</p> <pre>----- Name: Norm Address: 987 Suds Blvd. Phone: 555-8374 Thanks! ----- Name: Cliff Address: 321 Duds Lane Phone: 555-7282 Thanks! ----- Name: Agista Address: Bandung Phone: 1234567890 Social Security Number: 012-345-6789 Current hours: 35 Total Sales: 400.0 Paid: 333.75 ----- Name: Agista Address: Bandung Phone: 1234567890 Social Security Number: 012-345-6789 Current hours: 40 Total Sales: 950.0 Paid: 492.5 ----- BUILD SUCCESS Total time: 1.706 s Finished at: 2023-10-20T22:24:32+07:00 -----</pre> <p>1. Menambahkan class baru yaitu commission. 2. Update class Staff.</p> <p>Letak Polymorphism : Ketika memanggil methods ‘pay’ pada objek yang disimpan dalam array ‘staffList’ dalam methods ‘payday’ di class Staff.</p>
<p>Studi Kasus 2 :</p> <p>- Shape.java</p> <pre>12 abstract class Shape { 13 14 private String shapeName; 15 16 public Shape(String shapeName) { 17 super(); 18 this.shapeName = shapeName; 19 } 20 21 public abstract double area(); 22 23 public String toString() { 24 return "Shape Name : " + shapeName; 25 } 26 } 27</pre> <p>- Rectangle.java</p> <pre>11 public class Rectangle extends Shape{ 12 13 private double length; 14 private double width; 15 16 public Rectangle(double length, double width) { 17 super(shapeName: "Rectangle"); 18 this.length = length; 19 this.width = width; 20 } 21 22 public double area() { 23 return length * width; 24 } 25 26 public String toString() { 27 return super.toString() + " of length " + length + " and width " + width; 28 } 29 }</pre>	<pre>-----< com.mycompany:PaintThings >----- Building PaintThings 1.0-SNAPSHOT -----[jar]----- --- exec-maven-plugin:3.0.0:exec (default-cli) @ PaintThings --- Computing amount for Shape Name : Rectangle of length 20.0 and width 35.0 Computing amount for Shape Name : Sphere of radius 15.0 Computing amount for Shape Name : Cylinder of radius 10.0 and height 30.0 Number of gallons of paint needed... Deck 2 Big Ball 8.1 Tank 26.9 ----- BUILD SUCCESS Total time: 1.359 s Finished at: 2023-10-20T22:27:05+07:00 -----</pre> <p>1. Membuat abstract class shape. 2. Membuat class Rectangle dan Cylinder. 3. Menyesuaikan file paintThings dengan apa yang diperintahkan.</p> <p>Letak Polymorphism : Dalam abstract class Shape yang memiliki methods abstract area(). Dengan karakteristik bentuk geometris masing masing. Polymorphism ini memungkinkan untuk menghitung jumlah cat yang dibutuhkan untuk berbagai bentuk geometris tanpa perlu mengubah implementasi methods amount.</p>

- Cylinder.java

```
11 public class Cylinder extends Shape {
12
13     private double radius;
14     private double height;
15
16     public Cylinder(double radius, double height) {
17
18         super(shapeName: "Cylinder");
19         this.radius = radius;
20         this.height = height;
21     }
22
23     public double area() {
24
25         return Math.PI * (Math.pow(radius, 2)) * height;
26     }
27
28     public String toString() {
29
30         return super.toString() + " of radius " + radius + " and height " + height;
31     }
32 }
```

- PaintThings.java

```
12 import java.text.DecimalFormat;
13
14 public class PaintThings {
15
16     public static void main(String[] args) {
17
18         final double COVERAGE = 350;
19         Paint paint = new Paint(cov: COVERAGE);
20
21         Rectangle deck = new Rectangle(length: 20, width: 35);
22         Sphere bigBall = new Sphere(r: 15);
23         Cylinder tank = new Cylinder(radius: 10, height: 30);
24
25         double deckAmt = paint.amount(s: deck);
26         double ballAmt = paint.amount(s: bigBall);
27         double tankAmt = paint.amount(s: tank);
28
29         DecimalFormat fmt = new DecimalFormat(pattern: "0.##");
30         System.out.println("\nNumber of gallons of paint needed...");
31         System.out.println("Deck " + fmt.format(number: deckAmt));
32         System.out.println("Big Ball " + fmt.format(number: ballAmt));
33         System.out.println("Tank " + fmt.format(number: tankAmt));
34     }
35 }
36 }
```

Studi Kasus 3 :

- String.java

```
11 import java.util.Scanner;
12
13 public class Strings {
14
15     public static void main(String[] args) {
16
17         Comparable[] intList;
18         int size;
19
20         Scanner scan = new Scanner(source: System.in);
21
22         System.out.println("\nHow many word do you want to sort? ");
23         size = scan.nextInt();
24         intList = new Comparable[size];
25
26         System.out.println("\nEnter the word...");
27         for (int i = 0; i < size; i++)
28             intList[i] = scan.nextLine();
29         Sorting.selectionSort(list: intList);
30         System.out.println("\nYour word in sorted order...");
31         for (int i = 0; i < size; i++)
32             System.out.println(intList[i] + " ");
33         System.out.println();
34     }
35 }
36 }
```

- Sorting.java

```
11 public class Sorting {
12
13     public static void selectionSort(Comparable[] list) {
14         int min;
15         Comparable temp;
16
17         for (int index = 0; index < list.length - 1; index++) {
18             min = index;
19             for (int scan = index + 1; scan < list.length; scan++)
20                 if (list[scan].compareTo(list[min]) < 0)
21                     min = scan;
22
23             temp = list[min];
24             list[min] = list[index];
25             list[index] = temp;
26         }
27     }
28
29     public static void insertionSort(Comparable[] list) {
30         for (int index = 1; index < list.length; index++) {
31             Comparable key = list[index];
32             int position = index;
33
34             while (position > 0 && key.compareTo(list[position - 1]) < 0) {
35                 list[position] = list[position - 1];
36                 position--;
37             }
38             list[position] = key;
39         }
40     }
41 }
42 }
```

Output :

```
-----< com.mycompany:WeeklySales >-----
Building WeeklySales 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ WeeklySales ---

Ranking of Sales for the week

Taylor, Harry: 7300
Adams, Andy: 5000
Duck, Daffy: 4935
Jones, Jane: 3000
Jones, James: 3000
Black, Jane: 3000
Smith, Walt: 3000
Doe, Jim: 2850
Walter, Dick: 2800
Trump, Don: 1570

BUILD SUCCESS

Total time: 1.345 s
Finished at: 2023-10-20T22:28:12+07:00
```

1. Mengcompile dan membetulkan fungsi yang kurang tepat pada class number.
2. Modif InsertionSort.
3. Membuat class string, untuk membaca array objek String dan mengurutkannya.
4. Menggunakan nama pegawai penjualan.

<div><div>- WeeklySales.java</div><pre>12 public class WeeklySales { 13 14 public static void main(String[] args) { 15 16 SalesPerson[] salesStaff = new SalesPerson[10]; 17 18 salesStaff[0] = new SalesPerson(first: "Jane", last: "Jones", sales: 3000); 19 salesStaff[1] = new SalesPerson(first: "Daffy", last: "Duck", sales: 4935); 20 salesStaff[2] = new SalesPerson(first: "James", last: "Jones", sales: 3000); 21 salesStaff[3] = new SalesPerson(first: "Dick", last: "Walter", sales: 2800); 22 salesStaff[4] = new SalesPerson(first: "Don", last: "Trump", sales: 1570); 23 salesStaff[5] = new SalesPerson(first: "Jane", last: "Black", sales: 3000); 24 salesStaff[6] = new SalesPerson(first: "Harry", last: "Taylor", sales: 7300); 25 salesStaff[7] = new SalesPerson(first: "Andy", last: "Adams", sales: 5000); 26 salesStaff[8] = new SalesPerson(first: "Jim", last: "Doe", sales: 2850); 27 salesStaff[9] = new SalesPerson(first: "Walt", last: "Smith", sales: 3000); 28 29 Sorting.insertionSort(list: salesStaff); 30 31 System.out.println(s: "\nRanking of Sales for the week\n"); 32 33 for (SalesPerson s: salesStaff) 34 System.out.println(s: s); 35 } 36 }</pre></div>	<p>Letak Polymorphism :</p> <p>Dalam penggunaan interface Comparable yang digunakan untuk mendefine methods ‘compareTo’.</p>
<p>Teman yang membantu :</p> <ul style="list-style-type: none">- Mahesya Setia Nugraha	
<p>Source Code : https://github.com/agistadivab/PBO.git</p>	