TUGAS 3. Inheritance

TUGAS 3. Latihan Inheritance

Kerjakan soal-soal berikut ini. (Soal diambil dari buku Java How to Program 10th edition 2015, halaman 394)

9.8 (Quadrilateral Inheritance Hierarchy) Write an inheritance hierarchy for classes Quadrilateral, Trapezoid, Parallelogram, Rectangle and Square. Use Quadrilateral as the superclass of the hierarchy. Create and use a Point class to represent the points in each shape. Make the hierarchy as deep (i.e., as many levels) as possible. Specify the instance variables and methods for each class. The private instance variables of Quadrilateral should be the x-y coordinate pairs for the four endpoints of the Quadrilateral. Write a program that instantiates objects of your classes and outputs each object's area (except Quadrilateral).

9.14 (Employee Hierarchy) In this chapter, you studied an inheritance hierarchy in which class BasePlusCommissionEmployee inherited from class CommissionEmployee. However, not all types of employees are CommissionEmployees. In this exercise, you'll create a more general Employee superclass that factors out the attributes and behaviors in class CommissionEmployee that are common to all Employees. The common attributes and behaviors for all Employees are firstName, lastName, socialSecurityNumber, getFirstName, getLastName, getSocialSecurityNumber and a portion of method toString. Create a new superclass Employee that contains these instance variables and methods and a constructor. Next, rewrite class CommissionEmployee from Section 9.4.5 as a subclass of Employee. Class CommissionEmployee should contain only the instance variables and methods that are not declared in superclass Employee. Class CommissionEmployee's constructor should invoke class Employee's constructor and CommissionEmployee's toString method should invoke Employee's toString method. Once you've completed these modifications, run the CommissionEmployeeTest and BasePlusCommissionEmployeeTest apps using these new classes to ensure that the apps still display the same results for a CommissionEmployee object and BasePlusCommissionEmployee object, respectively.

9.15 (Creating a New Subclass of Employee) Other types of Employees might include SalariedEmployees who get paid a fixed weekly salary, PieceWorkers who get paid by the number of pieces they produce or HourlyEmployees who get paid an hourly wage with time-and-a-half—1.5 times the hourly wage—for hours worked over 40 hours. Create class HourlyEmployee that inherits from class Employee (Exercise 9.14) and has instance variable hours (a double) that represents the hours worked, instance variable wage (a double) that represents the wages per hour, a constructor that takes as arguments a first name, a last name, a social security number, an hourly wage and the number of hours worked, set and get methods for manipulating the hours and wage, an earnings method to calculate an HourlyEmployee's earnings based on the hours worked and a toString method that returns the HourlyEmployee's String representation. Method setWage should ensure that wage is nonnegative, and setHours should ensure that the value of hours is between 0 and 168 (the total number of hours in a week). Use class HourlyEmployee in a test program that's similar to the one in Fig. 9.5.

Petunjuk Pengerjaan

1. Masing-masing soal dikerjakan dalam satu project (satu project satu soal).
2. Hapus folder bin di masing-masing project
3. Kompres seluruh folder project menjadi satu file menggunakan format ZIP

   (Gunakan hanya format ZIP, jangan gunakan format lain seperti RAR, 7z, dll).

4. Beri nama file dengan format sebagai berikut:

   TUGAS3_PBO_KELAS_NRP.
   Contoh: TUGAS3_PBO_H_051119100xxx.zip

5. Kirimkan melalui myITS Classroom sebelum batas waktu (keterlambatan dikenai pinalti). Jika tidak bisa login harap segera hubungi asisten

Happy Coding!