Übung Java ShapeSwing Poly Interface

Ausgangslage

Mit dieser Übung nutzen wir nun die Polymorphie umd die Lösung zu optimieren. Zur Zeit werden die Shape Instanzen in der Methode paintComponent() immer wieder neu instanziiert und dann gezeichnet (draw). Dabei handelt es sich immer um die gleichen Shape Objekte. Die paintComponent() Methode wird je nach Nutzung vielfach aufgerufen. Es handelt sich hier also um eine schlechte Lösung hinsichtlich dem Memory Usage und der Performance. Dies werden wir nun mittels Polymorphie optimieren als Refactoring. Neu sollen alle Shape Objekt in einer Liste innerhalb der Klasse ShapeSwing geführt werden. Im Konstruktor werden die Shape Objekte instanziiert und initialisiert und der Liste hinzugefügt. In der Methode paintComponent() wird die Liste traversiert und gezeichnet (draw).In einem letzten Schritt lagern wir die Shape-Methoden getArea() und draw() in das Interface IShape aus.Das folgende Klassendiagramm zeigt die Interfaces und Klassen:

Vorgehen

Lösen Sie bitte die Aufgabe wie folgt:Öffnen Sie die Klasse ShapeSwing der Übung ShapeSwing Inheritance.Definieren Sie das Attribute shapeList wie folgt:private List<IShape> shapeList;Programmieren Sie den Default Konstruktor ShapeSwing und kopieren Sie den Inhalt der paintComponent() Methode. Entfernen Sie den Aufruf der draw-Methode und fügen Sie anstelle solcher die Shape Instanz in die shapeList, welche zuvor mit einer ArrayList zu instanzieren ist.Programmieren Sie die leere paintComponent() Methode aus indem Sie die shapeList mit einer forEach Schleife traversieren und jedes Shape Objekt via draw() zeichnen.Extrahieren Sie aus der Klasse Shape das Interface IShape gemäss dem Klassendiagramm. Ersetzen Sie den shapeList Generic Typ mit IShape.Das gezeichnete Bild sollte immer noch genau gleich gezeichnet werden.

Lösung

Eine mögliche Lösung finden Sie hier

Kontakt

Simtech AG
Finkenweg 23
3110 Münsingen
Schweiz

Impressum

Das Copyright für sämtliche Inhalte dieser Website liegt bei Simtech AG, Schweiz.
Beachten Sie auch unsere Hinweise zum Urheberrecht, Datenschutz und Haftungsauschluss.
Jeder Hinweis auf Fehler nehmen wir gerne entgegen.

Copyright

2024 Simtech AG, All rights reserved, Powered by stack.ch written in Golang by Daniel Schmutz

https://www.simtech-ag.ch/Deutschland Java 11 Schulung Kurs 8 Einführung Übungen OO