

Aufgabe 08: Geometrische Figuren

Nach einer kurzen, minimalen Einführung zum Thema *Generalisierung* (Vererbung) soll anhand dieser Aufgabe das Verständnis erweitert und vertieft werden.

Auftrag

Sie sollen die Grundlage einer Applikation zum Zeichnen einfacher geometrischer Figuren erstellen. Dieses Programm wird nach der Fertigstellung *keine* eigentliche Ausgabe bzw. Darstellung realisieren, sondern lediglich den konzeptionellen Teil beinhalten.

Für die Umsetzung der Applikation soll das Erstellen, Positionieren, Verschieben und Rotieren einfacher Figuren wie beispielsweise Kreis, Rechteck oder Dreieck implementiert werden. Als gute Programmiererin bzw. guter Programmierer erkennen Sie natürlich sofort, dass Sie hier das Konzept der Generalisierung anwenden können.

Hinweis zum Koordinatensystem

In der Programmierung ist es üblich, dass der Ursprung des Koordinatensystems (0,0) sich in der oberen, linken Ecke des Bildschirms bzw. des Fensters befindet. Die x-Achse erstreckt sich dabei von links nach rechts (analog zur x-Achse in der Mathematik). Anders als traditionelle Koordinatensysteme in der Mathematik, wächst die y-Achse von oben nach unten. Der höchste y-Wert befindet sich folglich am unteren Ende des Bildschirms bzw. des Fensters.

Beachten Sie für die Umsetzung dieser Aufgabe als neues Projekt (C#) die nachfolgenden Anforderungen:

1. Damit Sie alle Figuren auf der Zeichenfläche platzieren können, sollen diese eine X- und Y-Koordinate (`float`) haben.
2. Werte für diese beiden Koordinaten können, müssen jedoch nicht zwingend, beim Erzeugen der jeweiligen Objekte angegeben werden.
3. Wenn keine Werte für die X- und Y-Koordinate angegeben werden, wird jede Figur grundsätzlich an der Position [`100.0` , `100.0`] gezeichnet.
4. Die spezifischen Merkmale von Kreis und Rechteck sollen ebenfalls beim Erzeugen von Objekten der jeweiligen Klasse angegeben werden *können* (nicht zwingend).
5. Bei der Zuweisung von Werten zu figurspezifischen Merkmalen sollen nur positive Werte angenommen werden.
6. Durch den Aufruf einer Methode mit dem Namen `PrintPosition` soll die Position (obere, linke Ecke) aller Figuren anhand der Koordinaten auf die Konsole geschrieben werden.
7. Für Kreise soll bei der Ausgabe der Position (`PrintPosition`) nun der Mittelpunkt statt die obere linke Ecke auf die Konsole geschrieben werden.
8. Für Rechtecke soll beim Aufruf der Methode `PrintPosition` *zusätzlich* die Position der unteren, rechten Ecke auf die Konsole geschrieben werden.
9. Implementieren Sie eine Methode, welche es ermöglicht, Figuren um ihren Mittelpunkt um 90° im Uhrzeigersinn rotieren zu können.
10. Für alle Figuren soll ein Skalierungsfaktor (`_scaleFactor`) vom Datentyp `int` gespeichert werden können.
11. Der Skalierungsfaktor (gleicher Bezeichner: `_scaleFactor`) für Dreiecke soll genauer, d.h. mit dem Datentyp `double` gespeichert werden können.
12. Die Fläche jeder Figur soll durch einen Aufruf der Methode `PrintArea` auf die Konsole geschrieben werden können. Dabei wird die berechnete Fläche (aufgrund der figurspezifischen Daten) vor der Ausgabe jeweils mit dem Skalierungsfaktor multipliziert.

Arbeitsform

Es steht den Lernenden frei, ob die Umsetzung der Applikation in Einzel- oder Zweierarbeit erfolgt. Die Arbeitsphase der Lernenden wird in regelmässigen Abständen durch Inputs und mögliche Lösungen der Lehrperson unterbrochen.

Zeit

Für die Bearbeitung dieser Aufgabe stehen Ihnen während der Unterrichtszeit rund 2 ganze Lektionen zur Verfügung (inkl. Inputs durch die Lehrperson).

Ergebnissicherung

Erstellen Sie für die Bearbeitung dieser Aufgabe ein neues Konsolenprojekt (`dotnet new console`). Die Zwischenschritte, welche die Lehrperson anlässlich der Inputs präsentiert, werden Ihnen auf einem separaten Branch in diesem Repository zur Verfügung gestellt.

Auswertung

Im Verlauf der Bearbeitung dieser Aufgabe wird die mögliche Umsetzung verschiedener Anforderungen im Plenum besprochen. Dabei wird die Lehrperson entweder eine eigene, mögliche Implementation zeigen oder verschiedene Ansätze der Lernenden als Beispiele mit der Klasse besprechen.