

Einführung in die wissenschaftliche Programmierung

Übungsblatt 6

1.) Formen

Es sollen Klassen zur Darstellung von Formen entwickelt werden. Wir betrachten zunächst drei Formen: Kreis, Rechteck, und Quadrat. Für jede Form muss eine Klasse implementiert werden, die folgende Klassenvariablen definiert:

- `Kreis` hat einen Radius `R`
- `Rechteck` hat Länge `length` und Breite `width`
- `Quadrat` hat Länge `length`

Zusätzlich müssen alle Klassen die Variable `A` definieren, in der der Flächeinhalt der Form gespeichert wird, sowie drei Methoden:

- i) `computeArea`, die den Flächeinhalt der Form berechnet aber *nicht zurückgibt*
- ii) `getArea`, gibt den Flächeinhalt der Form zurück
- iii) `__add__`, die dem Benutzer ermöglicht, den `+` Operator zu verwenden, um zwei Objekte einer bestimmten Form zu „addieren“: addiert man zwei Formen, so addieren sich einfach deren Flächeinhalte, d.h. das Resultat der Addition ist eine Zahl.

Die Methode `computeArea` muss im Konstruktor (`__init__`) aufgerufen werden.

Vererbung

Nun sollen die drei Klassen von einer Basisklasse `Shape` erben. *Vererbung* ermöglicht uns, *Code-Duplikation* zu vermeiden, was als schlechte Programmierpraxis gilt. An welchen Stellen Ihres Codes wird Code dupliziert? Wie kann sich die Basisklasse `Shape` hierzu als nützlich erweisen?

2.) Numerische Quadratur revisited, revisited (Hands-on)

Auf Blatt 5 haben Sie die Klassen `Midpoint` und `Trapezoidal` implementiert. Deren Methode `integrate` ist ein weiteres Beispiel von Code-Duplikation. Schreiben Sie eine Basisklasse `QuadratureRule`, von der `Midpoint` und `Trapezoidal` ableiten. Diese neue Klasse soll alle Methoden implementieren, die alle Quadraturregeln gemeinsam haben.

Fortgeschritten

Erweitern Sie Ihren Code mit einer dritten Quadraturregel `MonteCarlo`, die definiert ist durch

$$I_{\text{MC}}(f) = \frac{b-a}{n} \sum_{i=1}^n f(a + \xi_i(b-a)), \quad (1)$$

wobei $\xi_i \in [0, 1)$ eine gleichverteilte Zufallsvariable ist. Wie kann man Formel (1) interpretieren?

3.) Bibliothek (Hausaufgabe)

Überlegen Sie sich vernünftige Klassen, um eine Bibliothek darzustellen. Beachten Sie folgende Punkte:

- Eine Bibliothek besteht aus Büchern und ein Buch besteht aus Seiten.
- In einer Bibliothek kann man Bücher hinzufügen, entfernen und suchen.
- Ein Buch hat immer einen Autor und einen Titel.
- Ein Buch kann man schreiben und lesen. Man kann einzelne Seiten herausreißen und neue Seiten (hinten) anfügen.
- Eine Seite kann man lesen und schreiben. Sie ist definiert über ihren Inhalt.

Implementieren Sie drei Klassen, die die oben genannten Funktionen unterstützen, d.h.

- i) die Klasse `Seite` hat eine Klassenvariable `text` (String) und zwei Methoden `read` (gibt `text` aus) und `write` (legt die Variable `text` fest)
- ii) die Klasse `Buch` hat drei Klassenvariablen `autor` (String), `titel` (String), und `seiten` (Liste). Die Klasse implementiert die Methoden `getAutor`, `getTitel`, `getSeitenzahl`, `neueSeite` (erweitert `seiten` um eine (leere) Seite), `schreibeSeite` (überschreibt den Text einer bestimmten Seite), und `lesen` (liest alle Seiten)
- iii) die Klasse `Bibliothek` hat eine Klassenvariable `buecher` (Liste) und drei Methoden `neuesBuch` (fügt ein neues `Buch` zur Liste `buecher` hinzu), `entferneBuch` (entfernt ein `Buch` aus `buecher`), und `getBuch` (gibt ein bestimmtes Objekt in `buecher` zurück).