

Einführung in die wissenschaftliche Programmierung

Übungsblatt 6

1.) Numerische Quadratur

Die Numerische Quadratur bezeichnet die näherungsweise Berechnung von bestimmten Integralen, d.h. Integralen der Form

$$\int_a^b f(x) dx.$$

Viele Quadraturregeln lassen sich als gewichtete Summe

$$I(f) = \sum_{i=1}^n w_i f(x_i), \quad (1)$$

schreiben. Zwei bekannte Quadraturregeln dieser Form sind die zusammengesetzte Rechtecksregel I_R und die zusammengesetzte Trapezregel I_T

$$I_R(f) = \frac{b-a}{n} \sum_{i=0}^{n-1} f\left(a + \left(i + \frac{1}{2}\right) \frac{b-a}{n}\right),$$
$$I_T(f) = \frac{b-a}{n-1} \left(\frac{f(a) + f(b)}{2} + \sum_{i=1}^{n-2} f\left(a + i \frac{b-a}{n-1}\right) \right).$$

Implementieren Sie eine Klasse `Integration` mit einer Methode `integrate`, welche die allgemeine Summe in (1) implementiert. Im Konstruktor von `Integration` soll dann eine Funktion `constructMethod` aufgerufen werden. Diese macht in der Klasse `Integration` erstmal nichts. Leiten Sie dann zwei Klassen `Midpoint` und `Trapezoidal` ab. Diese sollen nun die Methode `constructMethod` implementieren, wo die Gewichte w_i und die Stützstellen x_i von I_R bzw. I_T konstruiert werden. Welches Konzept der Objektorientierten Programmierung haben Sie angewendet?

Testen Sie Ihr Programm mit einem Polynom und der `polyval` Funktion.

2.) Bibliothek

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.
- In dieser besonderen Bibliothek kann man sogar eine Volltextsuche durch alle Bücher vornehmen.
- Ein Buch hat immer einen Autor und einen Titel.
- Ein Buch kann man schreiben und lesen. Man kann einzelne Seiten herausreißen und eine neue Seiten (hinten) anfügen.
- Eine Seite kann man lesen und schreiben. Sie ist definiert über ihren Inhalt.

Implementieren Sie diese Bibliothek in Python. Diese soll alle genannten Funktionen unterstützen.

3.) Formen

Es sollen Klassen zur Darstellung von Formen (Rechteck, Quadrat, Ellipse, Kreis, Dreieck, ...) entwickelt werden. Wir sind nur am Flächeninhalt der Formen interessiert. Jede Form hat einen Flächeninhalt. Dieser wird jedoch je nach Form anders berechnet. "Addiert" man zwei Formen, so addieren sich einfach deren Flächeninhalte, d.h. das Resultat der Addition ist eine Zahl.

Implementieren Sie geeignete Klassen um ein Rechteck, einen Kreis und ein Quadrat darzustellen. Welches grundlegende Konzept der Objektorientierten Programmierung bietet sich hier an? Wie unterscheidet es sich von dem Konzept von Aufgabe 2?