Software and Tools for Computational Engineering

RWTH AACHEN UNIVERSITY

Einführung in die Programmierung mit C++

Übungsblatt 10

Klassenhierarchie

Sebastian Christodoulou, Alexander Fleming

Informatik 12:
Software and Tools for Computational Engineering (STCE)

RWTH Aachen

1. Implementiere die folgenden Klassen und erarbeite eine entsprechende Klassenhierarchie, sodass die Hauptroutine in `main.cpp` ausführbar ist.
   - function: Die abstrakte Basis-Klasse function besitzt eine rein virtuelle **public** Routine eval zur Berechnung des Funktionswertes. Außerdem soll eine rein virtuelle **public** Routine print deklariert werden, die die mathematische Funktion auf der Konsole ausgibt.
   - integral: In dieser abstrakten Basis-Klasse sollen die Parameter für die Integration gespeichert werden (linke und rechte Integrationsgrenze, sowie die Anzahl der Auswertepunkte n). Diese können nur über den Konstruktor an die Objekte übergeben werden. Außerdem soll eine öffentlich verfügbare, rein virtuelle Funktion print deklariert werden, die einen Zeiger auf ein Objekt vom Typ function übergeben bekommt, sowie eine rein virtuelle **protected** Routine computeArea deklariert werden, die ebenfalls einen Zeiger auf ein Objekt vom Typ function übergeben bekommt.
     In den abgeleiteten Klassen (weiter unten) soll später print den Integralwert der übergebenen function für die Integralparameter auf der Kommandozeile ausgeben. Dies soll geschehen, indem print die funktion computeArea aufruft, die diesen Integralwert gemäß Integralparameter bestimmt.

- riemann: Diese Klasse ist eine Spezialisierung von integral. Sie implementiert die Reckteckregel. Dem Konstruktor sollen die zur Integration benötigten Parameter übergeben werden.
- montecarlo: Diese Klasse ist eine Spezialisierung von integral. Dem Konstruktor sollen die zur Integration benötigten Parameter übergeben werden.Außerdem soll der Zufallszahlengenerator initialisiert werden.

  Für die Algorithmen zur Integration per Rechteckregel und Monte-Carlo, siehe Lösung zur Übung 6.
- polynom: Diese Klasse ist eine Spezialisierung von function. Ein Objekt der Klasse polynom stellt die Funktion $f(x) = \sum_{i=0}^{d} p_i x^i$ dar. Neben einer Variable für den Grad des Polynoms, werden in einem std::vector die Konstanten für die benötigten Terme gespeichert. Die spezialisierte Version der Routine print sollte z.B. für ein Polynom 2. Ordnung folgende Ausgabe liefern:

  ```
  f(x) = p0 + p1 * x^1 + p2 * x^2
  ```

- exponential: Diese Klasse ist eine Spezialisierung von function. Ein Objekt der Klasse exponential stellt die Funktion $f(x) = a * exp(b * x) + c$ dar. Die Konstanten a, b und c sollen in der Klasse gespeichert werden. Die spezialisierte Version der Routine print sollte folgende Ausgabe liefern:
  `f(x) = a * exp( b * x) + c`

2. Füge in allen Destruktoren Konsolen-Ausgaben ein, um festzustellen, in welcher Reihenfolge die Destruktoren aufgerufen wurden. Auf der Konsole soll nach der Ausführung folgende Ausgabe sichtbar sein:

```
Integral in [0,4] von f(x) = 3 + 1.2 * x^1 + 0.7 * x^2:  36.5333
Integral in [0,4] von f(x) = 3 + 1.2 * x^1 + 0.7 * x^2:  35.36
Integral in [0,4] von f(x) = 0.2 * exp(0.8 * x) + 2.4:  15.4831
Integral in [0,4] von f(x) = 0.2 * exp(0.8 * x) + 2.4:  15.488
```

   gefolgt von den Ausgaben der Destruktoren.
   In der oben gezeigten Ausgabe wird ein `polynom` und `exponential` Objekt mit den angezeigten Parametern jeweils zuerst von einem `riemann`, dann von `montecralo` Objekt in deren `print` Routine aufgerufen. Die Zahlen der Monte Carlo Integration können leicht abweichen.

Abgabe:

- integrals.hpp
- functions.hpp
- main.cpp