

Angewandte Mathematik und Programmierung

Einführung in das Konzept der objektorientierten Anwendungen zu
wissenschaftlichen Rechnens mit C++ und Matlab

SS2013

Fomuso Ekellem



Inhalt

- **Übungsbesprechung**
- **Objekt orientierte Programmierung(Wiederholung)**
 - Wie ist OOP entstanden
 - Wozu dient OOP
 - Die wichtigsten Grundlagen
 - Klassen
- **Klassenkonzept (Wiederholung)**
 - Was ist eine Klasse
 - Die Klasse in C++
 - Deklaration, Definition und Zugriff
 - Zugriffsrechte(Geheimnisprinzip)



Objekt orientierte Programmierung

Wie ist OOP entstanden

- Ursprünglich dienten Computer ausschließlich dazu, numerische Aufgaben zu lösen. Ihre Aufgabe war es Rechenaufgaben zu lösen, deren Umfang und Komplexität die Grenzen menschlichen Rechenvermögens erreichte oder gar überschritt.
- Ein typisches Beispiel für diese Verwendung ist das Berechnen von Logarithmentafeln, eine Aufgabe, für die viele der ersten Computer konstruiert wurden. schlichen Rechenvermögens erreichte oder gar überschritt.
- Diese wurden für viele Berechnungen im militärischen und ingenieurwissenschaftlichen Bereich benötigt und die von Hand erstellten Tafeln waren zumeist sehr fehlerbehaftet.
- Für diese numerischen Zwecke war ein Programmierkonzept völlig ausreichend, das auf der einen Seite einen Satz von Daten vorsah (gegebenen und berechneten) und auf der anderen Seite Funktionen, um diese zu ändern.
- Hinzu kam, daß die Programme noch relativ klein waren und nur für eine bestimmte Nutzungszeit, den sogenannten Software-Lebenszyklus, entworfen wurden.



Objekt orientierte Programmierung

Wie ist OOP entstanden

- Mit steigenden Anforderungen an die Computer wuchs aber der Umfang der Programme sehr stark.
- Auch beschränkten sich die Anwendungen nicht mehr nur auf das Lösen numerischer Probleme, sondern der Computer vollzog einen Wandel zu einem universellen Arbeitsgerät, bis hin zu Multimedia, Künstlicher Intelligenz und computervermittelter Kommunikation.
- Diese Entwicklung erforderte ein Umdenken auch bei dem Programmierkonzept. Die Komplexität der Software erforderte nicht nur eine größere Übersicht bei der Programmierung, sondern es war auch nicht mehr praktikabel, Software jedesmal von Grund auf neu zu programmieren, um dann ein Produkt zu haben.
- Es wurde nötig, Software in kleine Sinneinheiten aufzuteilen, die immer wieder verwendet werden können. Man strukturierte statt dessen die Programmteile in Funktionseinheiten, die die für sie benötigten Daten gleich mit enthielten, womit das Konzept der OOP geboren war.



Objekt orientierte Programmierung

Wozu dient OOP

- Die grundlegende Idee der OOP ist es, durch Bildung von Sinneinheiten (Objekten) einen besser strukturierten Programmcode zu erzeugen.
- Diese Sinneinheiten bestehen zum einen aus Daten (Elementen), zum anderen aus Funktionen (Methoden).
- Dadurch entsteht eine leicht definierbare Schnittstelle zu den Objekten, die die Wiederverwendung erleichtert. Auch eine nachträgliche Änderung ist leicht durchführbar, solange sie die bestehende Schnittstelle nicht verändert (eine Erweiterung ist natürlich möglich).
- Durch das Vererbungskonzept (Unterklassen erben alle Eigenschaften der Oberklasse) ist es zusätzlich möglich, bestehende Objekte leicht um neue Funktionen zu erweitern bzw. Variationen bestehender Objekte zu erzeugen, was „Polymorphismus“ („Vielgestaltigkeit“) genannt wird.
- Die OOP versucht, sich an der menschlichen Denkweise zu orientieren. Die grundlegende Idee der OOP lautet wie folgt:
 - Die Welt besteht aus agierenden **Objekten**, welche in **Klassen** zusammengefasst werden können



Objekt orientierte Programmierung

Die wichtigsten Grundlagen

- Beginnen wir mit der Wiederholung des Grundlagenbegriffs **“Objekt”**
 - Jedes Objekt hat zur Identifikation einen eindeutigen **Namen**.
 - Die Daten eines Objekts werden als Attribute bzw. als Eigenschaften des Objektes bezeichnet
 - Die Prozesse werden als dynamische Eigenschaften, Verhalten oder Methoden des Objektes bezeichnet.
 - Attribute und Methoden bilden innerhalb des Objektes eine Einheit.
- **Attribute(Datentypen):**
 - Attribute beschreiben die Daten eines Objektes und werden durch ihren Namen und ihren Typ definiert.
 - Für die Attribute stehen die Standardtypen oder primitivtypen, wie “integer”, “float”, “char”, „float“, „bool“ und viel mehr, in der Programmiersprache C++ zur Verfügung.
 - Attribute können jedoch auch als Aufzählungstypen, komplexe Datenstruktur oder selbst als Objekt definiert werden.
- **Methodentypen:**
 - Methoden bezeichnen Funktionen, die von einem Objekt ausgeführt werden. Sie werden auch als Operationen bezeichnet.



Objekt orientierte Programmierung

Die wichtigsten Grundlagen

■ Es stehen

Objektoperationen oder Objektmethoden zur Verfügung.

- Diese werden stets auf ein bereits existierendes Objekt angewendet.
- In unserem Beispiel sind die Operationen “zahle aus” und “überweise” definiert.

Ein weiterer Operationstyp sind die

- **Konstruktoroperationen. (Konstruktoren)**

■ Objektdiagramme

- Objekte und deren Beziehungen untereinander werden in Objektdiagrammen dargestellt.



Objekt orientierte Programmierung

Klassen

- Eine Klasse definiert für eine Sammlung von gleichartigen Objekten die zugehörigen Attribute, Methoden und Beziehungen.
- Der Klassenname sollte im gesamten Betrachtungsraum eindeutig sein. Die Klasse besitzt die Fähigkeit neue Objekte zu erzeugen. Jedes erzeugte Objekt gehört genau einer Klasse an.



Klassenkonzept

Klausur Vorbereitung: Machen Sie sich Gedanken über diese!

- ☐ Was ist eine Klasse?
- ☐ Die Klasse in C++?
- ☐ Deklaration, Definition und Zugriff?
- ☐ Zugriffsrechte(Geheimnisprinzip)?