Objektorientierte Programmierung

01 - Einführung

Alexander Stuckenholz

Version 2022-05-04

Inhalt

HOCHSCHULE HAMM-LIPPSTADT

Programmierparadigma

Im letzten Semester haben wir gelernt, mit der Programmiersprache C umzugehen.

- Neben C existieren aber noch sehr viele andere Programmiersprachen.
- Manche ähneln C, andere fühlen sich doch ganz anders an.

Programmiersprachen werden in zwei fundamentale **Programmierparadigmen** eingeteilt.

- Die **imperativen** und die **deklarativen** Programmiersprachen.
- Diese Paradigmen legen fest, wie Probleme mithilfe der Sprachen gelöst werden können.

Dabei geht es darum, welche Programmelemente die jeweilige Programmiersprache anbietet und wie diese zusammenwirken.

- Statische Elemente: Variablen, Funktionen, Objekte, Module, ...
- Dynamische Elemente: Kontrollfluss, Datenfluss, ...

Imperative Sprachen

HOCHSCHULE HAMM-LIPPSTADT

In der **imperativen Programmierung** wird mithilfe von Anweisungen genau dargelegt, wie ein Problem zu lösen ist.

- Die Programmiersprache C ist eine imperative Programmiersprache.

Imperative Programmiersprachen basieren auf einigen fundamentalen Bausteinen:

- **Deklarationen**: Einführung von Variablen, Funktionen etc.
- **Ausdrücke**: Ein Programmkonstrukt, welches zur Laufzeit ausgewertet wird und ein Ergebnis erzeugt.
- **Datentypen**: Festlegung der Bedeutung eines Speicherinhalts, des erlaubten Wertebereichs und der Operationen.
- **Anweisungen**: Ein einzelner Befehl bzw. Instruktion an den Rechner.
- **Kontrollstrukturen**: Spezielle Anweisungen, welche die Reihenfolge der Abarbeitung der Anweisungen verändern.

Deklarative Sprachen

HOCHSCHULE HAMM-LIPPSTADT

Deklarativen Programmiersprachen basieren auf einem ganz anderen Prinzip.

- Meist wird lediglich das Problem beschrieben, das was.
- Der Lösungsweg wird dann automatisch ermittelt.

Zu den deklarativen Programmiersprachen gehören z.B. SQL, Haskell, Erlang oder Prolog.

- Das folgende Beispiel in SQL gibt alle Windkraftanlagen im Umkreis von 10 km um Hamm aus.

```
1  select * from geo.renewable_power_plants where technology='Onshore' and
2    ST_distance(geom, ST_SetSRID(ST_Point(7.8398118, 51.6820189)::geography, 4326)) <= 10000
```

Dabei wird nur beschrieben, was gesucht wird.

- Nicht aber, wie diese Ergebnisse zu beschaffen sind.

Objektorientierung

Imperative und deklarative Sprachen basieren auf gänzlich unterschiedlichen Konzepten.

- Aber auch innerhalb der jeweiligen Paradigmen existieren viele unterschiedliche Strömungen.

Innerhalb der imperativen Sprachen ist z.B. die Idee der **Objektorientierung** entstanden.

- Die Objektorientierung hilft vor allem bei der Entwicklung großer und komplexer Systeme.
- Große Aufgaben können durch Objekte in handhabbare Teilprobleme zerlegt werden.
- Die Objektorientierung hilft daher dabei, die Erstellung von Software handhabbarer, wartbarer und testbarer zu machen.

Mit diesem Prinzip werden wir uns im Rest dieser Veranstaltung auseinandersetzen.

Abstrakte Datentypen

HOCHSCHULE HAMM-LIPPSTADT

In der Programmiersprache C können **abstrakte Datentypen** deklarieren werden.

- Durch Zusammensetzen mehrerer bestehender Datentypen können so neue Datentypen geschaffen werden, z.B. ein Kreis mit einem Radius:

```
1  struct Kreis {
2     double radius;
3  };
```

Wir können nun beliebig viele Variablen von diesem Datentyp erzeugen: `Kreis k1 = { 4.5 };`

- Allein mit dieser Variable `k1` können wir aber wenig anfangen.

Wir benötigen Funktionen, um z.B. den Umfang des Kreises zu berechnen.

- In C werden Funktionen getrennt von solchen Datentypen deklariert:

```
1  double berechneUmfang(Kreis k) {
2     return k.radius * 2 * M_PI ;
3  }
```

Objektorientierte Programmierung

HOCHSCHULE HAMM-LIPPSTADT

Daten und Funktionen bilden aber meist eine Einheit!

- Eine Variable vom Typ Kreis ist ohne passende Funktionen nutzlos.
- Diese Erkenntnis ist eine Grundidee der objektorientierten Programmierung.

Objekte verbinden Daten mit Funktionen.

- Im Gegensatz zu abstrakten Datentypen verbergen (schützen) Objekte ihre Daten.
- Öffentlich zugänglich sind nur Funktionen, die auf den Daten erlaubte Operationen zulassen.

Mithilfe der objektorientierten Programmierung lässt sich auch gut die Realität abbilden.

- Es ist nicht schwer, in Objekten zu denken!
- Alles ist ein Objekt!

Eigenschaften von Objekten

Objekte haben drei wichtige Eigenschaften (siehe [1, S. 223])

Jedes Objekt hat eine Identität.

- Objekte sind voneinander unterscheidbar.
- Die Identität tragen Objekte von ihrer Entstehung bis zu ihrem Ende.

Jedes Objekt hat einen Zustand.

- Der Zustand eines Objekts ergibt sich durch seine Daten.
- Für diese Daten ist das Objekt selbst verantwortlich.

Jedes Objekt zeigt ein Verhalten.

- Das Verhalten des Objekts sind die Funktionen, die das Objekt zur Nutzung anbietet.
- Die Interaktion mit dem Objekt ist ausschließlich über diese Funktionen möglich.

Klasse

Das folgende Beispiel realisiert das Konzept eines Kreises in C# als sog. **Klasse**:

```
1  class Kreis
2  {
3     private int radius;
4
5     public Kreis(int radius)
6     {
7        this.radius = radius;
8     }
9
10    public double BerechneUmfang()
11    {
12       return radius * 2 * Math.PI;
13    }
14 }
```

Die Deklaration der Klasse verbindet die Daten (Radius) mit den Funktionen (Umfang).

Objekte

Aus so einer Klasse können nun beliebig viele Objekte erzeugt werden.

- Jedes Objekt besitzt dann seine eigenen, individuellen Daten.
- Einen initialen Wert für den Radius übergeben wir bei der Erzeugung der Objekte:

```
1  Kreis k1 = new Kreis(4.5);
2  Kreis k2 = new Kreis(9);
```

Die Funktionen sind nun integraler Bestandteil der Objekte.

- Sie können die ureigenen Daten des Objektes nutzen, um Ergebnisse zu berechnen.
- Die Funktionen werden auf dem Objekt aufgerufen.

```
1  double umfang1 = k1.BerechneUmfang();
2  double umfang2 = k2.BerechneUmfang();
```

Strukturelement Objekt

HOCHSCHULE HAMM-LIPPSTADT

Die objektorientierte Programmierung verändert die Strukturierung der Programme.

- Dinge, die zusammen gehören, können nun auch zusammengebracht werden.
- Das Objekt ist ein neues Programmelement aus denen Systeme aufgebaut werden.

Objekte helfen dabei, komplexe Systeme zu konstruieren.

- Man kann sich zunächst überlegen, welche Objekte sinnvollerweise gebraucht werden.
- Danach kann man die Daten und Funktionen realisieren.
- Das hilft auch bei der Aufgabenverteilung im Team.

Die objektorientierte Programmierung setzt auf der imperativen Programmierung auf.

- Algorithmen werden noch immer auf dieselbe Art realisiert.
- Nur eben als Teil von Objekten.

Objektorientierte Prinzipien

HOCHSCHULE HAMM-LIPPSTADT

Objekte sind die Bausteine, aus denen objektorientierte Systeme bestehen.

- Die Objekte werden dabei auf Basis von vier Kernkonzepten entworfen:

Abstraktion

- Programmierer lieben Abstraktionen.
- Anstelle Programmcode zu duplizieren, werden z.B. Funktionen eingeführt.
- Mit Klassen und Schnittstellen existieren dazu ganz neue Möglichkeiten.

Datenkapselung

- Objekte verbergen die Details ihrer Implementierung und schützen ihre Daten.
- Über die Schnittstelle legt ein Objekt die sinnvolle Nutzung fest.

Objektorientierte Prinzipien

HOCHSCHULE HAMM-LIPPSTADT

Vererbung

- Vererbung erlaubt, neue Arten von Objekten aus bestehenden Definitionen abzuleiten.
- Durch die Bildung einer Vererbungshierarchie kann zusätzliche Abstraktion erzeugt werden.

Polymorphie

- Polymorphie meint, dass sich Objekte abhängig vom Kontext ihrer Verwendung unterschiedlich verhalten können.
- Sie sorgt u.a. dafür, dass objektorientierte Systeme flexibel erweitert werden können, ohne bestehenden Programmcode anpassen zu müssen.

Diese Konzepte zielen darauf ab, die Kohäsion zu erhöhen und die Koppelung zu reduzieren.

- Eine Voraussetzung, um evolvierbare Softwaresysteme herzustellen.
- Was das alles genau bedeutet, werden wir in dieser Veranstaltung lernen.

Zusammenfassung

Wir haben heute gelernt, ...

- welche fundamental unterschiedlichen Programmierparadigmen existieren.
- aus welcher Idee die objektorientierte Programmierung entstanden ist.
- was Objekte von abstrakten Datentypen unterscheidet.
- was die vier Kernkonzepte der Objektorientierung sind.

Aufgaben

Erstellen Sie in der Programmiersprache C einen abstrakten Datentypen, der einen mathematischen Bruch (Zähler und Nenner) umsetzt.

- Realisieren Sie eine Funktion zur Multiplikation zweier Variablen des Typs.

Welche Werte darf der Nenner eines Bruchs nicht annehmen?

- Können Sie verhindern, dass Variablen mit solch ungültigen Werten überhaupt erzeugt werden?

Quellen I

[1] Christian Ullenboom. *Java ist auch eine Insel: Das Standardwerk für Programmierer. Über 1.000 Seiten Java-Wissen. Mit vielen Beispielen und Übungen, aktuell zu Java 14*. 15. Edition. Rheinwerk Computing, 25. Juni 2020. 1246 S. ISBN: 978-3-8362-7737-2.