

Entwicklung eines Softwaresystems zur Carsharing Simulation

23. August 2017 | Nina Grebing | Institut für Bio- und Geowissenschaften

Inhalt

- 1 Vorstellung
- 2 Aufgabenanalyse
- 3 Algorithmus
- 4 Programmkonzept
- 5 Tests
- 6 Ausblick

Teil I: Vorstellung

23. August 2017 | Nina Grebing

Forschungszentrum Jülich - IBG-1

- Forschung in den Bereichen Energie, Umwelt, Medizin, Simulationswissenschaften und Nanotechnologie
- 10 Institute (über 60 Institutsbereiche)
- Über 5000 Mitarbeiter
- Institut für Bio- und Geowissenschaften

Praktische Arbeit

Aufgaben während der Ausbildungszeit

- Entwicklung eines Konzeptes zur Normalisierung einer Datenbank
- Entwurf und Implementierung von PostgreSQL-Datenbanken
- Konvertierung von MSSQLServer-Datenbanken nach PostgreSQL
- Entwicklung von Webanwendungen

Teil II: Aufgabenanalyse

23. August 2017 | Nina Grebing

Aufgabenstellung

- Simulation und Auswertung des Kundenverhaltens eines Carsharing Unternehmens
- Stadt in $m \times m$ Teilgebiete unterteilt
- Verlauf von Nachfrage und Abstellung der Autos für jedes Gebiet als monoton wachsende Polynome maximal 4. Grades, positiv auf dem Intervall $[0;24]$
- Einlesen der Daten aus einer Datei
- Ermitteln der Bedarfsänderungen für jedes Gebiet
 - Integrale der Polynome bilden
 - Ganzzahlübergänge durch Bisektion näherungsweise bestimmen

Aufgabenstellung

- Ausgabe der Ergebnisse in eine Datei
 - Liste aller Abstellungen und Nachfragen sortiert nach Uhrzeit (Abstellungen zuerst)
 - maximaler Bedarf und Endzustand

Präzisierungen

- Minimum 1 Gebiet
- Maximum 10 Gebiete
- Abbruchgenauigkeit der Bisektion: $e < 10^{-4}$

Teil III: Algorithmus

23. August 2017 | Nina Grebing

Programmablauf

- Einlesen der Daten

```
# AHausen
# m
1
# Polynome Nachfrage
0 0.2023761 -0.0287711 0.0016925 -0.0000352
# Polynome Abstellungen
0.434782 0 0 0 0
```

Programmablauf

- Einlesen der Daten
- Erzeugen der Simulation mit den Teilgebieten
 - Teilgebiete anlegen
 - Bestimmen der Polynome für Nachfragen und Abstellungen

Programmablauf

- Einlesen der Daten
- Erzeugen der Simulation mit den Teilgebieten
- Berechnung der Bedarfsänderungen für jedes Teilgebiet
- Nachfrage- und Abstellungs-Änderungen berechnen über Bisektion
- zeitlich sortierte Liste aller Änderungen für jedes Teilgebiet erstellen
- Bedarfsänderungen aller Gebiete in einer Liste aufsteigend nach Zeitpunkt sortieren

Programmablauf

- Einlesen der Daten
- Erzeugen der Simulation mit den Teilgebieten
- Berechnung der Bedarfsänderungen für jedes Teilgebiet
- Bestimmen des maximalen Bedarfs und des Endzustandes
 - maximal erreichten Bedarf bestimmen
 - End-Bedarf bestimmen

Programmablauf

- Einlesen der Daten
- Erzeugen der Simulation mit den Teilgebieten
- Berechnung der Bedarfsänderungen für jedes Teilgebiet
- Bestimmen des maximalen Bedarfs und des Endzustandes
- Ausgabe der Ergebnisse

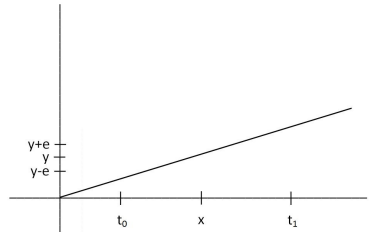
```
# AHausen
Abstellung in Q_11 zu t=2,30
Nachfrage in Q_11 zu t=3,76
Abstellung in Q_11 zu t=4,60
Nachfrage in Q_11 zu t=5,87
Abstellung in Q_11 zu t=6,90
Nachfrage in Q_11 zu t=7,86
Abstellung in Q_11 zu t=9,20
Nachfrage in Q_11 zu t=9,89
Abstellung in Q_11 zu t=11,50
Nachfrage in Q_11 zu t=11,96
Abstellung in Q_11 zu t=13,80
Nachfrage in Q_11 zu t=14,03
Nachfrage in Q_11 zu t=16,06
Abstellung in Q_11 zu t=16,10
Nachfrage in Q_11 zu t=18,06
Abstellung in Q_11 zu t=18,40
Nachfrage in Q_11 zu t=20,17
Abstellung in Q_11 zu t=20,70
Abstellung in Q_11 zu t=23,00
Nachfrage in Q_11 zu t=23,43
Endzustand des Tages:
0
Maximaler Bedarf:
1
```

Berechnung der Bedarfsänderungen

- Berechnung für jedes Teilgebiet
 - Berechnung für Nachfragen und Abstellung
 - Nullstellenproblem (nächster ganzzahliger Wert wird vom Funktionswert subtrahiert)
 - Bisektion im Intervall $[t_0, t_1]$
 - t_0 , beginnend mit 0, Stelle der letzten Änderung
 - $t_1 = 24$
 - Änderungen der Nachfragen und Abstellungen zusammenfügen und zeitlich sortieren
- Änderungen aller Teilgebiete zusammenfügen

Bisektion

- bestimme y = Funktionswert an der Stelle x - nächster ganzzahliger Wert
- $x = t_0 + \frac{t_1 - t_0}{2}$
- Abbruch bei:
 - $|y| < e = 10^{-4}$
 - gebe x zurück
- ansonsten Bisektion erneut durchführen
 - $y < 0$: Bisektion für $t_0 = x, t_1 = t_1$
 - $y > 0$: Bisektion für $t_0 = t_0, t_1 = x$



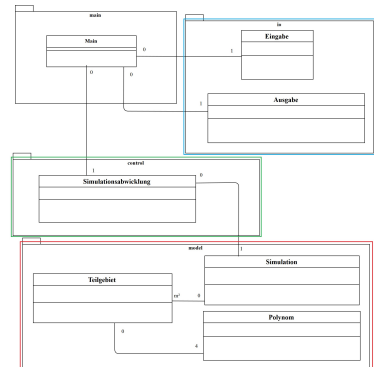
Teil IV: Programmkonzept

23. August 2017 | Nina Grebing

Klassenübersicht

MVC

- **Model:** Teilgebiet, Simulation, Polynom
- **View:** Eingabe, Ausgabe
- **Controller:** Simulationsabwicklung



Klassenübersicht

Teilgebiet

speichert:

- die Polynome für Nachfrage und Abstellung
- die Bedarfsänderungen
- den Endzustand
- den maximalen Bedarf

Teilgebiet
- nachfrage: Polynom - nachfrageIntegral: Polynom - abstellung: Polynom - abstellungIntegral: Polynom - änderungen: float[][] - endzustand: int - maxBedarf: int - bedarf: float[][]
+ Teilgebiet(nachfrage: Polynom, abstellung: Polynom)

Klassenübersicht

Simulationsabwicklung

berechnet

- Integrale
- Bedarf
- Endzustand
- maximalen Bedarf

Simulationsabwicklung
- simulation: Simulation
+ Simulationsabwicklung(simulation:Simulation) - bestimmeIntegrale() - bestimmeÄnderungen() - bestimmeÄnderungen(Teilgebiet t) - fügeÄnderungenZusammen(n: float[][], a: float[][]): float[][] - bisektion(funktion: Polynom, a: float, b: float): float - berechneBedarf() - berechneBedarf(Teilgebiet t): float [][] - bestimmeEndzustand() - bestimmeMaxBedarf() + änderungenToString(): String + endzustandToString(): String + maxBedarfToString(): String + beschreibungToString(): String

Teil V: Tests

23. August 2017 | Nina Grebing

Teststrategie

White-Box-Tests

- Test gegen die Implementierung
- alle Programmzweige werden durchlaufen

Testklassen

- Normalfälle
- Sonderfälle
- Fehlerfälle

Testklassen

Normalfälle:

- $2 \leq m \leq 10$
- mindestens eine Nachfrage oder Abstellung in der gesamten Simulation
- höchstens 75 Nachfragen und 75 Abstellungen pro Teilgebiet

Sonderfälle:

- $m = 1$
- Nachfragen und Abstellungen für ein Teilgebiet gleich
- alle Teilgebiete sind äquivalent

Testklassen

Fehlerfälle:

- $m = 0$
- $m > 10$
- Anzahl von Nachfragen/Abstellung für ein Teilgebiet > 75
- keine Bedarfsänderungen in der gesamten Simulation
- erste Zeile der Eingabedatei keine Kommentarzeile
- kein Zahlenwert für m
- kein Zahlenwert für Koeffizienten eines Polynoms
- zu wenig Polynome
- zu viele Nicht-Kommentarzeilen
- zu wenig Koeffizienten für ein Polynom
- zu viele Koeffizienten für ein Polynom

Teil VI: Ausblick

23. August 2017 | Nina Grebing

Erweiterungsmöglichkeiten

- Stadtgebiet auch als Rechteck ermöglichen
- Polynome höheren Grades als 4 zulassen
- Verschiebung der Autos bestimmen und ausgeben
 - möglichst wenige Verschiebungen
 - nach Möglichkeit den Endzustand aller Teilgebiete zu 0 ändern
- Parallelisierung
 - Integrale und Bedarfsänderungen der Gebiete können unabhängig voneinander berechnet werden

Vielen Dank für Ihre Aufmerksamkeit!

Teil VII: Zusatzinformationen

23. August 2017 | Nina Grebing

Sequenzdiagramm

