

2. Programmieraufgabe

Objektorientierte Programmietechniken

LVA-Nr. 185.A01
2017/2018 W
TU Wien

Kontext

Hinter den Träumen von Max und Irene (1. Programmieraufgabe) steckt der Klimawandel. Die vielbeschworene Energiewende scheint unausweichlich, Trinkwasser könnte sogar bei uns knapp werden, und zunehmend auftretende katastrophale Naturereignisse zerstören immer wieder Versorgungseinrichtungen. Wir müssen handeln. Die Frage ist: „Wie?“ Je nach Weltanschauung sehen viele in den Ideen hinter den Träumen von Max und Irene (von Privathäusern und -gärten ausgedehnt auf Betriebe, Dörfer, Städte und Regionen) den einzig gangbaren Weg, während andere solche Ideen als Schrebergartenmentalität abtun, dahinter soziale Ungerechtigkeiten vermuten und radikale Lösungen fordern.

Betrachten wir die Energiewende allgemeiner: In wenigen Jahren bzw. Jahrzehnten soll die gesamte Energieversorgung auf erneuerbare Energiequellen umgestellt sein. Der Gesamtenergieverbrauch soll durch technische Verbesserungen sinken und ein Großteil der nötigen Energie durch nachhaltig erzeugte Elektrizität abgedeckt werden, sodass der restliche Bedarf aus nachwachsenden biologischen Rohstoffen gewonnen werden kann. Theoretisch ist die gesamte Elektrizität leicht durch konsequente Nutzung von Wasserkraft, Windenergie, Sonnenstrahlung (z.B. Photovoltaik), Erdwärme, etc. erzeugbar. Allerdings ist ein Großteil derart erzeugter Energie meist nicht zu der Zeit und an dem Ort verfügbar, wo sie benötigt wird. Gewaltige Investitionen in Speichersysteme (einschließlich konventioneller Kraftwerke, die ständig nur für den Notfall bereitstehen) und extrem leistungsfähige Leitungsnetze sind nötig. Neben hohen Kosten verhindern auch Umweltbedenken, politische Widerstände und Verteilungskämpfe auf internationaler Ebene den raschen Ausbau. Zudem sind solche Systeme anfällig für Naturkatastrophen und Anschläge und müssen entsprechend geschützt und redundant ausgelegt werden. Auch andere Versorgungssysteme (Trinkwasser, Fernwärme, Abwasserreinigung, etc.) und insbesondere Verkehrssysteme werden aufgrund ihrer Energieabhängigkeit zunehmend komplexer.

Ideen hinter den Träumen von Max und Irene reduzieren die Belastungen der Versorgungssysteme nur dann, wenn die lokalen Speichersysteme tatsächlich ausreichende Kapazitäten aufweisen. Andernfalls kommt es zu Zusatzbelastungen, weil ein Überschuss oder Zusatzbedarf fast überall in derselben Gegend gleichzeitig auftritt und somit die sehr teuren überregionalen Speichersysteme und Leitungsnetze verstärkt beansprucht werden. Es besteht also die Gefahr, dass Max und Irene die Kosten, die sie bei kleinerer Auslegung ihrer Speicher einsparen, auf die Allgemeinheit abwälzen, insbesondere auf ärmere Bevölkerungsschichten, die keine eigenen Versorgungsanlagen betreiben. Die richtige Dimensionierung lokaler Speichersysteme ist von überregionaler Bedeutung. Entscheidende Daten dazu kommen aus Simulationen wie in der 1. Programmieraufgabe.

Themen:

Aufwandsabschätzung,
Programmiereffizienz,
Untertypbeziehungen,
dynamisches Binden

Ausgabe:

18. 10. 2017

Abgabe (Deadline):

25. 10. 2017, 12:00 Uhr

Abgabeverzeichnis:

Aufgabe1-3

Programmaufruf:

java Test

Grundlage:

Skriptum, Schwerpunkt
auf Abschnitt 2.1

Welche Aufgabe zu lösen ist

Es zeigt sich, dass die Lösung der 1. Programmieraufgabe für den praktischen Einsatz zu viele Faktoren unberücksichtigt lässt, keine zuverlässigen Ergebnisse liefert und sehr aufwendig zu bedienen ist. Führen Sie das Programm durch Verbesserungen in vielen Bereichen an die praktische Verwendbarkeit heran. Hier sind einige Vorschläge:

das sind Vorschläge,
nicht verpflichtend

- Das Programm soll in der Lage sein, Versorgungssysteme auf allen Ebenen zu simulieren, von kleinen privaten über regionale und nationale bis zu ganz großen internationalen.
- Die Annahme einer einzigen Quelle und eines einzigen Speichers reicht nicht. Tatsächlich existieren verschiedenartige Quellen und Speicher parallel, die auf unterschiedliche Weise miteinander verbunden sind. Verbindungen können dynamisch konfigurierbar sein.
- Die Steuerung der simulierten Systeme, vor allem der Verbraucher, kann äußerst komplex sein. Diese Algorithmen sind auch in der Simulation zu verwenden um realistische Ergebnisse zu erhalten.
- Das angenommene Wetter beeinflusst Ergebnisse stark. Statt auf Zufallszahlen sollte die Simulation auf real gemessenen Wetterdaten vergangener Jahre aufbauen. Ebenso existieren Aufzeichnungen über reale Verbrauchsverläufe.
- Transportverluste, der Schwund von Speicherinhalten (auch ohne Verbrauch) sowie die Alterung (z.B. Maximalkapazität eines Speichers wird im Laufe der Zeit kleiner) sind zu berücksichtigen.
- Die Möglichkeit technischer Gebrechen wie der Ausfall einer Verbindung oder eines Speichers (mit Verlust des Speicherinhalts) ist zu berücksichtigen. Gleiches gilt für typische Bedienungsfehler (wie ein im Winter offen gelassene Fenster oder ein nicht abgedrehter Wasserhahn), die sich drastisch auf den Verbrauch auswirken.
- Ein hierarchischer Aufbau von Simulationen soll möglich sein: Ergebnisse einer Simulation auf einer Ebene sind als Eingangsdaten für eine Simulation auf einer höheren Ebene verwendbar. Rückkopplungen sind ebenfalls möglich, wobei Ergebnisse auf einer höheren Ebene Eingaben auf einer niedrigeren Ebene darstellen.
- Ergebnisse von Simulationsläufen sind im Detail darzustellen, wobei die genaue Form der Darstellung konfigurierbar ist. Daneben sind auch überblicksartige Darstellungen mit verschiedenen Diagrammen und statistischen Zusammenfassungen sinnvoll.
- Es wird nicht jeder Simulationslauf separat gestartet, sondern ganze Reihen von Simulationsläufen, die sich in bestimmten Parametern unterscheiden, werden auf einmal gestartet und Unterschiede in den Ergebnissen einander gegenübergestellt.
- Die Interpretation der Ergebnisse von Simulationsläufen erfolgt bis zu einem gewissen Grad automatisch, sodass Auffälligkeiten bereits in der Darstellung der Ergebnisse herausgestrichen werden.

- Obwohl ganz unterschiedliche Systeme simulierbar sein sollen, ist es wahrscheinlich sinnvoll, für bestimmte wichtige Systeme (Stromversorgung, Wasserversorgung, etc.) Spezialisierungen zu verwenden, sodass systemspezifische Elemente einfacher integrierbar sind und auf das System bezogene Begriffe verwendet werden.
- Eine eingebettete Sprache zur Spezifikation von neuen simulierten Systemen kann die Verwendbarkeit des Programms verbessern.
- In der Simulation sollen nicht nur die Flüsse von Quellen über Speicher zu Verbrauchern dargestellt werden, sondern (zur Untersuchung der Kostenwahrheit) auch die damit verbundenen Kosten.
- Ausfallsicherheit ist ein wichtiges Thema. In speziellen Simulationsläufen werden automatisch an zufälligen Stellen Teilausfälle eingestreut um ihre Auswirkungen aufzuzeigen.

Bestimmen Sie selbst den Funktionsumfang. Konzentrieren Sie sich auf den Kern des Programms (ohne Benutzerschnittstelle) sowie das Testprogramm. Der Kern soll möglichst viel der nötigen (vorgeschlagenen oder selbst gefundenen) Funktionalität abdecken, aber (möglicherweise abgesehen von der Ausgabe von Diagrammen und Ähnlichem) keinerlei Eingabe von der Tastatur oder Ausgabe auf den Bildschirm machen – klare Trennung von Programmkern und Tests.

Das Testprogramm soll mittels `java Test` von Aufgabe1-3 aufrufbar sein und die selbst gewählte Funktionalität überprüfen. Tests sollen ohne Benutzerinteraktion ablaufen, sodass Aufrufer keine Testfälle auswählen oder Testdaten eintippen müssen. Ergebnisse sollen (in nachvollziehbarer und verständlicher Form) am Bildschirm ausgegeben werden.

Neben dem Testprogramm soll die Klasse `Test.java` als Kommentar die Grobstruktur und den (geplanten) Funktionsumfang der Lösung zusammenfassen sowie eine kurze, aber verständliche Beschreibung der Aufteilung der Arbeiten auf die einzelnen Gruppenmitglieder enthalten.

Funktionsumfang
selbst wählen

im richtigen Verzeichnis
testen

Struktur,
Funktionsumfang,
Aufgabenaufteilung
beschreiben

Wie die Aufgabe zu lösen ist

Die oben aufgezählten Vorschläge zur Verbesserung der Simulation sind als Anhaltspunkte gedacht. Sie können Punkte weglassen, abändern oder durch andere sinnvoll erscheinende Verbesserungen ersetzen.

Eine der größten Schwierigkeiten dieser Aufgabe besteht in der richtigen Abschätzung des Umfangs der Arbeiten, die Sie bis zum Abgabetermin fertigstellen können. Planen Sie entsprechend Ihren Vorkenntnissen und Fähigkeiten möglichst viel ein, das Sie in der vorgesehenen Zeit auch zum Abschluss bringen können – das heißt, so viel Sie können, aber nicht mehr als Sie können. Als groben Anhaltspunkt sollten Sie (jedes Gruppenmitglied) etwa sechs bis sieben mit intensiver Arbeit gefüllte Stunden in die Lösung dieser Aufgabe fließen lassen. Versuchen Sie so effizient wie möglich zu arbeiten und sehr rasch zu einer brauchbaren Lösung zu kommen. Ignorieren Sie Details, die Ihnen als unwichtig erscheinen. Bedenken Sie, dass Sie alle wichtigen Teile Ihrer Lösung durch Testfälle überprüfen sollen und planen Sie die Zeit für die Entwicklung der Testfälle und für die Fehlerbeseitigung ein.

Erstellen Sie ein Konzept, von dem Sie annehmen, dass Sie die Arbeiten zeitlich schaffen. Schicken Sie es frühzeitig Ihrer Tutorin oder Ihrem Tutor. Sie werden so bald wie möglich erfahren, ob das Konzept hinreichend ist oder ob Änderungen vorgeschlagen werden.

Konzept an Tutor(in)

Einer der Schwerpunkte dieser Aufgabe ist der Umgang mit Untertypbeziehungen. Planen Sie die Verwendung von Untertypbeziehungen zusammen mit dynamischem Binden ein.

Untertypbeziehungen

Wie bei jeder der ersten drei Aufgaben weist Ihr(e) Tutor(in) Sie möglicherweise nach der Abgabe auf ganz konkrete schwerwiegende Fehler hin und bittet um Beseitigung. Dies wirkt sich nur dann auf Ihre Beurteilung aus, wenn Sie der Bitte nicht nachkommen. Dennoch sollten Sie selbst (auch ohne entsprechendes Feedback) ein Gefühl für die Qualität Ihrer Lösungen entwickeln. Aufgrund der großen Zahl erwarteter schwerwiegender Fehler können Tutor(inn)en nicht auf jede kleine Nachlässigkeit in den abgegebenen Lösungen eingehen.

Warum die Aufgabe diese Form hat

Sie sollen möglichst große Freiheit bei der Lösung der Aufgabe haben und selbst die Verantwortung für alles übernehmen. Niemand schreibt Ihnen vor, wie die Aufgabenstellung genau zu verstehen ist.

Diese Aufgabe stellt hohe Anforderungen an jedes Gruppenmitglied und die Zusammenarbeit in der Gruppe – eine Nagelprobe für das Funktionieren der Gruppe und zum Aufdecken möglicher Schwachstellen.

Untertypbeziehungen sind ein schwieriges, aber für die objektorientierte Programmierung sehr wichtiges Thema. Nutzen Sie die Gelegenheit, bei der Lösung der Aufgabe Erfahrungen damit zu sammeln. Fehler, die Sie dabei machen, wirken sich nicht sofort auf Ihre Beurteilung aus.