

# 1. Programmieraufgabe

**Objektorientierte  
Programmiertechniken**

LVA-Nr. 185.A01

2017/2018 W

TU Wien

## Kontext: Stromversorgung

Max träumt von einem Haus mit weitgehend autonomer Stromversorgung: Die Photovoltaikanlage am Dach speist Strom in einen Stromspeicher (aufladbare Batterien), und alle Stromverbraucher im Haus beziehen ihren Strom aus dem Speicher. Nur wenn der Speicher leer ist wird Strom aus dem öffentlichen Stromnetz bezogen, und nur wenn der Speicher voll ist wird überschüssiger Strom in das Stromnetz gespeist. Manche Stromverbraucher wie die Lüftungsanlage und Wärmepumpe (für Heizung und Warmwasserbereitung) richten sich nach dem Stromangebot: Sie schalten sich vorwiegend dann ein, wenn der Speicher zu einem gewissen Grad gefüllt ist und die Photovoltaikanlage genügend Strom produziert.

Photovoltaikanlage und Stromspeicher sind teuer und sollen so dimensioniert sein, dass möglichst selten Strom über das öffentliche Stromnetz fließt (weder in die eine noch in die andere Richtung) und der Speicher dennoch möglichst klein bleibt.

## Kontext: Gartengießanlage

Irene träumt von einem Garten mit automatischer Gießanlage: Regenwasser, das auf das Hausdach fällt, wird in einem großen Tank gesammelt. Zu fixen Zeiten und nach Bedarf pumpen kleine elektrische Pumpen über unterirdische Schläuche Wasser aus dem Tank zu den einzelnen Gartenbeeten. Der Bedarf orientiert sich an der über Sensoren ermittelten Wetterlage: Bei Regen braucht nicht gegossen werden, bei Sonne muss gegossen werden, und bei hohen Temperaturen muss verstärkt gegossen werden. Ist der Tank leer, muss Trinkwasser aus der öffentlichen Wasserversorgung bezogen werden, und bei vollem Tank wird überschüssiges Regenwasser über die Kanalisation entsorgt.

Der Tank braucht viel Platz und soll möglichst klein dimensioniert sein, aber dennoch groß genug, damit nur selten Trinkwasser zum Gießen herangezogen werden muss.

## Welche Aufgabe zu lösen ist

Entwickeln Sie ein Java-Programm zur Simulation von Anlagen wie jenen aus den beiden oben beschriebenen Kontexten. Das heißt, mit nur einem Programm soll es möglich sein,

- die Stromversorgung so zu simulieren, dass über Simulationsläufe nach einer guten Dimensionierung der Photovoltaikanlage und des Stromspeichers gesucht werden kann,
- die Gartengießanlage so zu simulieren, dass nach einer guten Dimensionierung des Regenwassertanks gesucht werden kann.

Gemeinsamkeiten in den beiden Kontexten sollen ausgenutzt werden, und wesentliche Unterschiede in den Kontexten müssen durch Eingabedaten darstellbar sein.

## Themen:

Aufbau der Zusammenarbeit in der Gruppe,  
Einrichten einer Arbeitsumgebung,  
Datenabstraktion,  
Klassenzusammenhalt,  
Objektkopplung

## Ausgabe:

11. 10. 2017

## Abgabe (Deadline):

18. 10. 2017, 12:00 Uhr

## Abgabeverzeichnis:

Aufgabe1-3

Hochladen ins Git-Repository mittels push

## Programmaufruf:

java Test

## Grundlage:

Kapitel 1 und Anhang A  
des Skriptums

Konkret sind folgende Systemteile zu simulieren:

- Eine *Quelle* erzeugt unabhängig von Verbrauchern irgendetwas, wobei Zeitpunkte und Mengen der Erzeugung zufällig sind, die Wahrscheinlichkeit aber vorgegebenen Mustern entspricht (etwa nachts nichts und tagsüber gleichverteilt zwischen 10% und 100% eines festgelegten Maximums, oder unabhängig von der Tageszeit zu 90% aller Stunden nichts und in den restlichen 10% gleichverteilt zwischen 0% und 100% eines festgelegten Maximums). Die aktuelle Erzeugungsmenge ist jederzeit abfragbar.
- Ein *Speicher* mit festgelegter Kapazität nimmt das von der Quelle Erzeugte auf, sofern noch Platz ist. Auf Anforderung gibt der Speicher etwas ab, sofern noch etwas vorhanden ist. Überschüsse und Fehlbestände werden protokolliert. Der aktuelle Füllstand ist jederzeit abfragbar.
- Eine Menge von *Verbrauchern* fordert zufallsgesteuert oder zu fixen Zeiten wiederholt etwas vom Speicher an, wobei der Zufall (wie bei der Quelle) Wahrscheinlichkeitsmustern entspricht. Zusätzlich können Verbrauchsanforderungen davon abhängen, wie voll der Speicher ist und wie viel die Quelle gerade erzeugt.

Machen Sie mehrere Simulationsläufe (entsprechend beider Kontexte) und stellen Sie fest, wie häufig und in welcher Menge Überschüsse und Fehlbestände protokolliert wurden.

Schreiben Sie nur den Kern eines solchen Programms (ohne Benutzerschnittstelle) sowie ein Testprogramm. Der Kern soll die oben beschriebene Funktionalität aufweisen, aber keinerlei Eingabe von der Tastatur oder Ausgabe auf den Bildschirm (und auch nicht von/auf eine Datei) machen. Daher müssen alle benötigten bzw. erzeugten Daten über Parameter und Rückgabewerte von Methoden übergeben werden.

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

Neben dem Testprogramm soll die Klasse `Test.java` als Kommentar eine kurze, aber verständliche Beschreibung der Aufteilung der Arbeiten auf die einzelnen Gruppenmitglieder enthalten — wer hat was gemacht.

nur Programmkern, keine Benutzerschnittstelle

Ausführbarkeit in richtigem Verzeichnis prüfen, package-Anweisungen nur wo Pakete verwendet

Arbeitsaufteilung beschreiben

## Wie die Aufgabe zu lösen ist

Der Programmcode Ihrer Lösung soll möglichst einfach sein und keine unnötige Funktionalität haben. Der Code soll wiederverwendbar sein, da die nächste Aufgabe auf Teilen davon aufbaut. Vermeiden Sie jedoch Vorgriffe, das heißt, schreiben Sie keine Programmteile aufgrund der Vermutung, dass diese Teile in der nächsten Aufgabe verlangt sein könnten.

Achten Sie auf Datenabstraktion: Alles, was kaum trennbar miteinander verbunden ist, soll in einem Objekt gekapselt sein, leicht voneinander trennbare Einheiten sollen zu verschiedenen Objekten gehören. Es soll

einfach halten

Datenabstraktion

in Ihrem Programm mehrere, voneinander möglichst unabhängige Objekte geben (auch wenn sich das aufgrund der Aufgabenbeschreibung nicht von alleine ergibt). Auf Daten soll nur über dafür vorgesehene Methoden zugegriffen werden. Unnötige Zugriffe und unnötige Zugreifbarkeit von Daten und Methoden sind zu vermeiden. Achten Sie auf hohen Klassenzusammenhalt und schwache Objektkopplung.

Klassenzusammenhalt,  
Objektkopplung

Diese Aufgabe hilft auch den Tutor(inn)en bei der Einschätzung Ihrer Kenntnisse sowie der Zusammenarbeit in der Gruppe. Bitte sorgen Sie in Ihrem eigenen Interesse dafür, dass Sie diese Aufgabe in der Gruppe lösen (ohne Abschreiben) und jedes Gruppenmitglied etwa in gleichem Maße mitarbeitet. Sonst könnten Sie bei einer Fehleinstufung wertvolle Zeit verlieren. Scheuen Sie sich bitte nicht, Ihre(n) Tutor(in) um Hilfe zu bitten, falls Sie bei der Lösung der Aufgabe Probleme haben oder keine brauchbare Zusammenarbeit in der Gruppe zustandekommt.

### Warum die Aufgabe diese Form hat

Der Umfang der Aufgabe ist so gewählt, dass die Programmierung bei guter Organisation nicht viel Zeit in Anspruch nehmen sollte. Nutzen Sie die Gelegenheit um die Aufgabenteilung und interne Abläufe innerhalb der Gruppe zu organisieren. Auf eine genaue Spezifikation der Aufgabenstellung wird aus folgenden Gründen bewusst verzichtet:

- Sie sollen in der Gruppe diskutieren, wie Sie die Aufgabe verstehen und welche Lösungswege geeignet erscheinen.
- Sie sollen sich von Anfang an daran gewöhnen, dass Aufgaben in der Regel nicht vollständig spezifiziert sind, aber trotzdem bestimmte Vorgaben eingehalten werden müssen.
- Sie sollen sich eine eigene brauchbare Faktorisierung überlegen und dabei von Merkmalen wie Datenkapselung, Klassenzusammenhalt und Objektkopplung (statt starrer Vorgaben) leiten lassen.
- Sie sollen auch die Verantwortung für die Korrektheit Ihrer Lösung (so wie Sie sie selbst verstehen) übernehmen, indem Sie entsprechende Tests durchführen.

### Allgemeine Informationen zur Übung

Folgende Informationen betreffen diese und auch alle weiteren Aufgaben.

### Was bei der Lösung der Aufgabe zu beachten ist

Unter der Überschrift „Wie die Aufgabe zu lösen ist“ finden Sie einerseits Hinweise darauf, wie Sie die Lösung der Aufgabe vereinfachen können und welche Fallen Sie umgehen sollen, erfahren andererseits aber auch, welche Aspekte bei der Beurteilung als besonders wichtig betrachtet werden. Bei der ersten Aufgabe kommt es beispielsweise besonders auf Datenabstraktion, Klassenzusammenhalt, Objektkopplung und die Einfachheit der Lösung an. Das heißt, in späteren Aufgaben können Ihnen bei solchen Hinweisen auch für unnötig komplizierte oder umfangreiche

Schwerpunkte beachten

Lösungen Punkte abgezogen werden, weil Sie sich nicht an die Vorgaben gehalten haben. „Fleißaufgaben“ sind unerwünscht. Unterschiedliche Aufgaben haben unterschiedliche Schwerpunkte. Die nächste Aufgabe wird nicht nach dem gleichen Schema beurteilt wie die vorige. Richten Sie sich daher nach der jeweiligen Aufgabenstellung.

Ein häufiger Fehler besteht darin, eine Aufgabe nur nach Gefühl zu lösen ohne zu verstehen, worauf es ankommt. Meist bezieht sich die Aufgabe auf ein Thema, das kurz zuvor in der Vorlesung behandelt wurde. Versuchen Sie, eine Beziehung zwischen der Aufgabenstellung und dem Vorlesungsstoff herzustellen. Achten Sie besonders darauf, Fachbegriffe (wie Datenabstraktion, Klassenzusammenhalt und Objektkopplung) nicht nur umgangssprachlich zu interpretieren, sondern verwenden Sie diese Begriffe so wie in der Vorlesung und im Skriptum beschrieben. Die ersten Aufgaben sind vermutlich auch ohne Skriptum leicht lösbar, spätere aber kaum. Als Hilfestellung sind in jeder Aufgabenstellung Teile des Skriptums genannt, in denen die relevantesten Themen behandelt werden – bei komplizierten Themen oft nur wenige Seiten.

strukturiert vorgehen

Versuchen Sie nicht, Teile der Aufgabenstellung durch Tricks oder Spitzfindigkeiten zu umgehen. Beispielsweise gibt es immer wieder Lösungsversuche, in denen die Test-Klasse nur den String „Tests erfolgreich“ ausgibt statt tatsächlich Tests durchzuführen. Solche Versuche werden durch händische Beurteilungen mit hoher Wahrscheinlichkeit erkannt. Spätere Aufgaben enthalten oft Schwierigkeiten, die mit Allgemeinwissen alleine oder über aufgabenbezogene Internet-Recherchen kaum zu lösen sind. Gerade in solchen Fällen ist davon abzuraten, die Schwierigkeiten durch Tricks zu umgehen. Hinweise zur richtigen Lösung lassen sich im Skriptum und auf den Vorlesungsfolien finden.

keine Spitzfindigkeiten

## Was Ihr(e) Tutor(in) von Ihnen wissen möchte

Ihr(e) Tutor(in) wird Ihnen in Kürze eine Mail schreiben, in der sie oder er sich vorstellt und um Informationen über Sie bittet. Geben Sie ihr oder ihm diese Informationen möglichst bald, damit die Tutorin bzw. der Tutor die für Sie am besten geeignete Form der Betreuung wählen kann. Unabhängig von der Form der Betreuung kann natürlich jedes Gruppenmitglied jederzeit konkrete Fragen an den Tutor oder die Tutorin richten. Scheuen Sie sich bitte nicht, sich auch mit organisatorischen oder gruppeninternen Problemen, die Sie möglicherweise nicht selbst lösen können, an den Tutor oder die Tutorin zu wenden. Je früher Tutor(inn)en von Problemen erfahren, desto einfacher können Lösungen gefunden werden.