

1. Programmieraufgabe

Kontext

Photovoltaik (kurz PV) stellt eine relativ einfache Technik zur Stromerzeugung dar, die auf dem besten Weg dazu ist, integraler Bestandteil der meisten Gebäude und überbauten Flächen zu werden. Früher wurden PV-Paneele einfach nur etwa 45 Grad geneigt nach Süden ausgerichtet, um die maximale Sonneneinstrahlung pro Fläche aufnehmen zu können und damit den größtmöglichen Kilowattpeak (kWp, Maximalleistung unter Standardbedingungen, etwa 0,2 kWp pro m²) zu erzielen. Heute reicht dieses Kriterium nicht mehr, unter anderem deshalb:

- Es geht darum, Strom dann zu produzieren, wenn er gebraucht wird, nicht wenn er am einfachsten erzeugbar ist. Häufig ist der Bedarf am Vor- und Nachmittag größer als zu Mittag. Im Herbst und Winter ist ein Ertrag natürlich wertvoller als im Frühling und Sommer.
- Eine PV-Anlage enthält teure Komponenten wie einen Stromwandler, die auf die Maximalleistung des Systems ausgelegt sein müssen. Diese Komponenten sind billiger und haben geringere Leistungsverluste, wenn nicht alle Paneele zur gleichen Zeit ihre Maximalleistung liefern, sondern zu unterschiedlichen Tageszeiten.
- Wenn eine um 45 Grad Richtung Süd geneigte Fläche 100 % (ca. 200 kWh pro m² und Jahr) liefert, liefert eine um 45 Grad Richtung Süd-Ost oder Süd-West geneigte Fläche 95 %, bei 45 Grad nach Westen oder Osten 75 %, eine komplett horizontale Fläche etwa 90 %, eine vertikale Fläche nach Süden ausgerichtet etwa 70 %, vertikal nach Süd-Ost oder Süd-West 65 % und vertikal nach Westen oder Osten 50 %. Das sind durchschnittlich über ein Jahr gerechnete Erträge, die nicht berücksichtigen, dass Erträge zu manchen Zeiten wertvoller sind als zu anderen. Bei großem Energiebedarf sind auch 50 % Ertrag sehr wertvoll.
- Horizontale Flächen können gleichmäßige Erträge liefern (90 %). Allerdings verschmutzen solche Flächen schnell und werden nicht durch Niederschläge gereinigt. Ein Reinigungseffekt setzt bei etwa 15 bis 20 Grad Neigung ein, ist bei etwa 25 Grad ideal und ab etwa 50 bis 60 Grad kaum mehr gegeben. Ein mit 25 Grad nach West-Ost orientiertes Dach ist günstig, weil beide Dachhälften für PV nutzbar sind. Ein nach Nord-Süd orientiertes Dach kann nur die Südseite nutzen und sollte für mehr Fläche steiler sein.
- Vertikale Flächen Richtung Süden können wegen nötiger Fenster (dienen der passiven Nutzung der Sonnenenergie) zu maximal 50 %, nach Süd-West oder Süd-Ost zu maximal 65 % und nach Westen oder Osten zu maximal 80 % für PV genutzt werden.

Objektorientierte
Programmiertechniken

LVA-Nr. 185.A01
2022/2023 W
TU Wien

Themen:

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

Ausgabe:

10. 10. 2022

Abgabe (Deadline):

17. 10. 2022, 14:00 Uhr

Abgabeverzeichnis:

Aufgabe1-3

Hochladen ins Git-
Repository mittels push

Programmaufruf:

java Test

Grundlage:

Kapitel 1 und Anhang A
des Skriptums

Welche Aufgabe zu lösen ist

Simulationsläufe: Entwickeln Sie ein Java-Programm, das den Betrieb einer PV-Anlage in mehreren Simulationsläufen mit unterschiedlichen Annahmen über je ein Jahr simuliert und dabei einmal pro Stunde den aktuellen Zustand ermittelt. Diese Zustände sollen in folgende (jährliche) Kennzahlen zusammengefasst werden:

- Kilowattpeak (kWp), die unter Annahme optimaler Witterungsverhältnisse erreichbare Maximalleistung zu irgendeinem Zeitpunkt,
- gesamter Jahresertrag der PV-Anlage (in kWh),
- Summe der im Jahr nicht selbst verbrauchten und daher in das Stromnetz gespeisten Energie (in kWh),
- Summe der im Jahr zusätzlich zur Bedarfsdeckung aus dem Stromnetz bezogenen Energie (in kWh).

Berechnen Sie über ein Jahr stündlich den Sonnenstand (für einen von Ihnen gewählten Ort in Europa¹) und lassen Sie die Simulation für jede Kombination folgender in der Simulation angenommener Modelle laufen:

- drei verschiedene Modelle von Gebäuden zusammengesetzt aus verschiedenen orientierten und zu einem bestimmten Anteil für PV genutzten Flächen (bzw. vereinfacht jeweils eine Menge von Flächen bestimmter Größe und Orientierung, aber beliebiger Form),
- drei verschiedene Witterungsverläufe über ein Jahr (stündliche Annahmen über den Anteil des Lichts, das aufgrund der aktuellen Wetterlage zu den PV-Paneelelen gelangt),
- drei verschiedene Szenarien für den Stromverbrauch (stündliche Annahmen über den aktuellen Stromverbrauch).

Berücksichtigen Sie, dass der Energieertrag davon abhängt, in welchem Winkel Licht auf die Fläche strahlt (maximaler Ertrag bei senkrechter Bestrahlung, deutlich weniger Ertrag bei flachem Winkel). Wählen Sie Annahmen so, dass Ergebnisse sich nicht sehr weit von den unter „Kontext“ genannten Zahlen entfernen.

Testen, Aufgabenaufteilung: Testen Sie Ihr Programm sorgfältig. Nach dem Programmstart sollen ohne Benutzerinteraktion (das heißt, ohne Eingabe über die Tastatur oder Maus) die Simulationsergebnisse möglichst übersichtlich angezeigt werden. Das Programm soll (nach neuerlicher Übersetzung aller vorhandenen .java-Dateien) mittels `java Test` vom Verzeichnis `Aufgabe1-3` aus aufrufbar sein.

Neben Programmtext soll die Datei `Test.java` als Kommentar eine kurze, aber verständliche Beschreibung der Aufteilung der Arbeiten auf die einzelnen Gruppenmitglieder enthalten – wer hat was gemacht. Beschreibungen wie die folgende reichen nicht: „Alle haben mitgearbeitet.“

Programmname „Test“ aus gutem Grund gewählt

im richtigen Verzeichnis ausführbar?

Arbeitsaufteilung kurz, verständlich beschreiben

¹Es ist nicht schwer, Formeln zur Berechnung des Sonnenstands im Internet zu finden. Benutzen Sie eine möglichst einfache Formel, auch wenn sie nur Näherungswerte liefert. Bitte binden Sie kein fertiges Programm zur Berechnung des Sonnenstands oder zur Ermittlung von Wetterdaten ein. Es besteht eine Einschränkung auf Europa, damit die oben genannten Zahlen einigermaßen gut zutreffen.

Wie die Aufgabe zu lösen ist

Der Programmtext Ihrer Lösung soll möglichst einfach sein und keine unnötige Funktionalität haben. Er 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.

einfach halten

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 in Ihrem Programm mehrere, voneinander möglichst unabhängige Objekte geben. 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.

Datenabstraktion

Szenarien für den Witterungsverlauf sowie den Stromverbrauch können mithilfe von Zufallszahlen generiert werden. Allerdings kommt es darauf an, dass die Ergebnisse einigermaßen realitätsnah sind. Beim Stromverbrauch bietet es sich an, fixe Vorgaben für den täglichen und jahreszeitlichen Verlauf zu machen und diese über Zufallszahlen zu modifizieren. Gebäudemodelle sind vermutlich einfacher per Hand zu erstellen als über Zufallszahlen (als kleine Mengen von Flächen, die in Ihrer Vorstellung Teile von Gebäudeoberflächen sind).

Klassenzusammenhalt,
Objektkopplung

Diese Aufgabe hilft Ihren Tutor_innen, Ihre Kenntnisse sowie die Zusammenarbeit in der Gruppe einzuschätzen. Bitte sorgen Sie in Ihrem eigenen Interesse dafür, dass jedes Gruppenmitglied etwa in gleichem Maße mitarbeitet. Sonst könnten Sie bei einer Fehleinschätzung wertvolle Zeit verlieren. Scheuen Sie sich bitte nicht, Ihre Tutorin oder Ihren Tutor um Hilfe zu bitten, falls Sie bei der Lösung der Aufgabe Probleme haben oder keine brauchbare Zusammenarbeit in der Gruppe zustande kommt.

Warum die Aufgabe diese Form hat

Umfang und Schwierigkeitsgrad der Aufgabe wurden so gewählt, dass die eigentliche Programmierung bei guter Organisation und entsprechendem Vorwissen nicht zu viel Zeit in Anspruch nimmt, aber eine Einarbeitung in neue Themen nötig ist und Diskussionsbedarf entsteht. Nutzen Sie die Gelegenheit, um die Aufgabenverteilung und interne Abläufe innerhalb der Gruppe zu organisieren. Details der Aufgabe bleiben bewusst offen:

- Sie sollen in der Gruppe diskutieren, wie Sie die Aufgabe verstehen und welche Lösungswege geeignet erscheinen.
- Sie sollen sich daran gewöhnen, dass Aufgaben nicht vollständig spezifiziert sind, aber trotzdem 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 Hinweise darauf, wie Sie die Lösung der Aufgabe vereinfachen können und welche Fallen Sie umgehen sollen, erfahren aber auch, welche Aspekte bei der Beurteilung als wichtig betrachtet werden. In der ersten Aufgabe kommt es beispielsweise 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 Lösungen Punkte abgezogen werden, weil Sie sich nicht an die Vorgaben gehalten haben. 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.

Schwerpunkte beachten

Ein häufiger Fehler besteht darin, eine Aufgabe nach Gefühl zu lösen ohne zu verstehen, worauf es ankommt. Meist bezieht sich die Aufgabe auf ein Thema, das zuvor theoretisch behandelt wurde. Versuchen Sie, eine Beziehung zwischen der Aufgabenstellung und dem OOP-Stoff herzustellen. Achten Sie darauf, Fachbegriffe (wie Datenabstraktion, Klassenzusammenhalt und Objektkopplung) nicht nur umgangssprachlich zu interpretieren, sondern verwenden Sie diese Begriffe so wie im Skriptum beschrieben. Die ersten Aufgaben sind vermutlich auch ohne Skriptum 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 Web-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

Ihre Lösung bestehend aus .java-Dateien muss am Tag der Abgabe um 14:00 Uhr pünktlich im Git-Repository Ihrer Gruppe stehen. Übersetzte .class-Dateien sollen nicht ins Repository gestellt werden, da sie die Zusammenarbeit in der Gruppe erschweren und vor der Beurteilung ohnehin neu generiert werden. Zugangsinformationen zum Repository erhalten Sie in Kürze. Informationen zum Umgang mit dem Repository finden Sie im Anhang des Skriptums. Es wird empfohlen, rechtzeitig vor der Deadline die Lösung auf dem Übungsrechner auszuprobieren und die Daten dabei durch pull aus dem Repository zu laden. So können Sie typische Fehler bei der Abgabe (z.B. auf push vergessen, Lösung im falschen Verzeichnis, falsche package- und include-Anweisungen, Klassen aus Nicht-Standard-Paketen verwendet und nicht mit abgegeben)

ausprobieren

sowie Inkompatibilitäten aufgrund unterschiedlicher Versionen und Betriebssystemeinstellungen (z. B. Dateinamen mit Umlauten sowie neueste Sprach-Features nicht unterstützt) erkennen und beseitigen.

Was Ihr_e Tutor_in von Ihnen wissen möchte

Ihre Tutorin oder Ihr Tutor wird Ihnen in Kürze eine Mail schreiben, um sich vorzustellen und um Informationen über Sie zu bitten. Geben Sie diese Informationen möglichst bald, damit die für Sie am besten geeignete Form der Betreuung gewählt werden kann. Unabhängig von der Form der Betreuung kann natürlich jedes Gruppenmitglied jederzeit konkrete Fragen an Tutor_innen richten. Scheuen Sie sich bitte nicht, sich auch mit organisatorischen oder gruppeninternen Problemen, die Sie möglicherweise nicht selbst lösen können, an Tutor_innen zu wenden. Früh im Semester sind Probleme meist einfacher zu lösen als im bereits weit fortgeschrittenen Semester.