



1. Übungsaufgabe

Complang
Puntigam
Über mich
Research
Lehre
LVAs 2015 W
PK
OOP
1. Aufgabe
Typsysteme
LVAs 2015 S
frühere Lehre
Links

Thema:

Aufbau der Zusammenarbeit in Gruppe

Termine:

Ausgabe: 14.10.2015

Abgabe: 21.10.2015, 12:00 Uhr

Abgabeverzeichnis:

Aufgabe1-3

Programmaufruf:

java Test

Grundlage:

Erstes Kapitel des Skriptums

Aufgabe

Welche Aufgabe zu lösen ist:

Nach dem ausgesprochen heißen und trockenen Sommer beschließt die Gemeinde Hinterwaldgstetten, einen Beitrag zum Klimaschutz leisten zu wollen. Konkret soll der seit Jahren brach liegende Gemeindegrund, der damals noch unter dem ehemaligen Bürgermeister für einen nie realisierten Sport- und Freizeitpark angeschafft worden war, in das angrenzende Waldgebiet integriert werden. Das Grundstück soll mit Bäumen bepflanzt werden, die möglichst viel Kohlendioxid aus der Luft für einen möglichst langen Zeitraum binden. Es ist jedoch noch nicht klar, wie man dieses Ziel am besten erreicht: Bei Nutzung als Energiewald wird durch schnellwachsende Bäume (wie Pappel, Weide und Robinie) rasch viel Kohlendioxid gebunden, aber beim Verbrennen des Holzes wird ein Großteil dieses Kohlendioxids bald wieder freigesetzt; jedoch ersetzt man fossile Energie. Bei Gewinnung von Konstruktions-, Faser- und Möbelholz (z.B. Fichte und Buche) bleibt ein bedeutender Teil des Kohlendioxids für einen längeren Zeitraum gebunden; jedoch wird das Kohlendioxid durch den langsameren Holzzuwachs nicht so rasch gebunden, und es besteht die Gefahr, dass ein Großteil des Holzes durch Umwelteinflüsse entwertet und dadurch auch nur durch Verbrennen nutzbar wird. Man kann auch auf eine intensive Nutzung verzichten und das Kohlendioxid bis zur natürlichen Verrottung des Holzes gebunden lassen. Natürlich ist auch eine Kombination verschiedener Strategien vorstellbar. So bleiben abgestorbene Baumteile sowie Wurzelstöcke und kleinere Äste üblicherweise auch bei intensiver Nutzung als Totholz bis zur natürlichen Verrottung im Wald.

Unterstützen Sie die Gemeinde Hinterwaldgstetten, indem Sie ein Java-Programm zur Simulation der Waldnutzung über viele Jahre entwickeln. Das Programm soll folgende Parameter verwalten (das heißt, jeden dieser Parameter anzeigen und verändern lassen):

- durchschnittlicher Holzzuwachs in Kubikmetern pro Jahr (dabei wird Kohlendioxid gebunden),
- durchschnittliche Erntemenge an Holz in Kubikmetern pro Jahr,
- durchschnittlicher Anfall von im Wald verbleibendem Totholz in Kubikmetern pro Jahr,
- durchschnittliche endgültige Verwertung (durch Verbrennen oder Verrotten) des geernteten Holzes in Prozent pro Jahr gemessen an der Menge des noch nicht endgültig verwerteten geernteten Holzes (dabei wird Kohlendioxid frei),
- durchschnittliche Verrottung von Totholz in Prozent pro Jahr gemessen am im Wald liegenden Totholz (dabei wird Kohlendioxid frei),
- alle in einem Simulationslauf benötigten Anfangswerte an Holzmengen in Kubikmetern.

Ein Kubikmeter an Holz entspricht einer Tonne an gebundenem Kohlendioxid. In einem Simulationslauf für eine gegebene Zahl an Jahren ist aus diesen Parametern eine Tabelle mit jährlich zumindest folgenden Daten zu erzeugen:

- Gesamtmenge des sich im Wald befindlichen lebenden Holzes (also aufsummierter Holzzuwachs abzüglich geerntetem Holz und Totholz)
- Gesamtmenge des im Wald liegenden (noch nicht verrotteten) Totholzes,
- Gesamtmenge des geernteten Holzes, das noch im Umlauf ist (also noch nicht endgültig verwertet wurde),
- Gesamtmenge des bereits endgültig verwerteten geernteten Holzes (zur Abschätzung der Menge an eingesparter fossiler Energie),
- Gesamtmenge des derzeit gebundenen Kohlendioxids in Tonnen.

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.

Wie die Aufgabe zu lösen ist:

Der Programmcode Ihrer Lösung soll möglichst einfach sein und keine unnötige Funktionalität haben. Der Code soll auch wiederverwendbar sein, da die nächste Aufgabe voraussichtlich 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 bei der Lösung der Aufgabe besonders 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 (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.

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 internen 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 Aufgabenstellungen.

Was Sie bei der Lösung der Aufgabe beachten sollen:

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 Lösungen Punkte abgezogen werden, weil Sie sich nicht an die Vorgaben gehalten haben. „Fleißaufgaben“ sind in diesem Fall also unerwünscht. Unterschiedliche Aufgaben haben unterschiedliche Schwerpunkte. Verlassen Sie sich nicht darauf, dass die nächste Aufgabe nach demselben Schema beurteilt wird wie die vorangegangene, sondern richten Sie sich 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 werden Sie vermutlich auch ohne Skriptum noch leicht lösen können, spätere Aufgaben aber kaum mehr. Als Hilfestellung sind in jeder Aufgabenstellung Teile des Skriptums genannt, in denen die relevantesten Themen behandelt werden – bei komplizierten Themen oft nur wenige Seiten.

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.

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

Ihr(e) Tutor(in) wird Ihnen in Kürze eine Mail schreiben, in der er oder sie sich vorstellt und um Informationen über Sie bittet. Geben Sie ihm oder ihr diese Information möglichst bald, damit der Tutor bzw. die Tutorin 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 um so einfacher sind sie zu lösen.