

Informationsblatt zu Vorbereitungstest 1

Dieses Informationsblatt zeigt die Struktur und mögliche Inhalte einer Vorbereitungstestaufgabe. Es enthält die Textbausteine, die auch in der Datei `angabe.md` beim Vorbereitungstest zu finden sein werden. Konkret sind das:

- die Beschreibungen der Domänen der Aufgabenstellungen, wie sie beim Vorbereitungstest gegeben werden. In diesem Informationsblatt sind mehrere Domänenbeschreibungen enthalten. Beim eigentlichen Vorbereitungstest wird eine Aufgabenstellung zu einer dieser Domänen gegeben.
- allgemeine Angabetexte und Hinweise, die genau so auch in der Datei `angabe.md` beim Vorbereitungstest zu finden sein werden.

EP2 Test 1

Allgemeine Hinweise

- Die Lösung Ihrer Aufgabe muss im vorgegebenen Projekt und damit in den vorhandenen Dateien erfolgen.
- Sie dürfen zur Lösung dieser Aufgabe *nicht* auf das Java Collections Framework zurückgreifen.
- Verändern Sie keine der vorgegebenen Methodensignaturen.
- Implementieren Sie keine unnötigen Getter-/Setter-Methoden.
- Alle Objektvariablen und etwaige von Ihnen zusätzlich erstellte Methoden in vorgegebenen Klassen müssen `private` sein.
- Definieren Sie keine geschachtelten oder (anonymen) inneren Klassen, also keine Klassen, die in einer anderen Klasse, einem Objekt oder einer Methode stehen.
- Auch für teilweise korrekte Lösungen werden Punkte vergeben.

Information zu möglichen Domänen

Variante A

Rationale Zahlen (`rational numbers`) können als Bruchzahlen dargestellt werden, wobei jede Bruchzahl zwei Bestandteile hat, nämlich Zähler (`numerator`) und Nenner (`denominator`). Der Wert eines Bruches ergibt sich aus Zähler dividiert durch Nenner. Brüche kann man in gekürzter Form darstellen, indem man Zähler und Nenner durch deren größten gemeinsamen Teiler (`greatest common divisor - gcd`) dividiert. Der Wert des Bruches wird dadurch nicht verändert. Bruchzahlen mit gleichem Nenner werden addiert, indem ihre Zähler addiert werden. Um Bruchzahlen mit unterschiedlichen Nennern zu addieren, müssen sie zuerst auf gleichen Nenner gebracht werden, etwa indem sowohl Zähler als auch Nenner jeweils mit dem Nenner der anderen zu addierenden Zahl multipliziert werden. Bruchzahlen werden multipliziert, indem sowohl die Zähler als auch

die Nenner miteinander multipliziert werden. Bruchzahlen in einer Datenstruktur können beispielsweise nach ihrem Wert, aber auch nach ihrem Nenner sortiert sein.

Variante B

Ein Apfel (apple) ist eine Frucht, die auf einem Baum (tree) wächst. Die Süße (sweetness) eines Apfels hängt davon ab, mit wie viel Sonneneinstrahlung der Apfel versorgt wird, und die Sonneneinstrahlung hängt davon ab, wie hoch oben im Baum (height) der Apfel wächst. Der Wert für die Süße eines Apfels lässt sich über eine Formel aus der Höhe im Baum errechnen. Die Stellen, an denen Äpfel im Baum wachsen, hängen von der Reihenfolge ab, in der Bienen (bees) im Frühjahr die Blüten bestäuben. An jeder Verzweigung im Baum hängt im Frühjahr eine Blüte, aber Bienen bestäuben nur jede n-te Blüte, wobei n eine ganze Zahl größer 1 ist. Nur aus einer bestäubten Blüte bildet sich ein Apfel. Die Qualität des aus den Äpfeln eines Baums gewonnenen Apfelsafts (appleJuice) hängt von der Summe der Werte für die Süße aller Äpfel des Baums ab.

Variante C

Ein Kamel (camel) ist ein Wüstentier, dass Lasten (load) tragen kann. Wir nehmen an, dass jedes Kamel auch eine Stärke (strength) hat. Um eine Leistung erbringen zu können, benötigt ein Kamel Futter (food) und Wasser (water). Die Stärke eines Kamels kann davon abhängen, wie viel Futter und Wasser vorhanden ist. Die maximale Geschwindigkeit (maximal pace), mit der das Kamel sich fortbewegen kann, lässt sich über eine Formel aus load und strength berechnen.

Eine Karavane (caravan) ist eine Reihe sich hintereinander (im Gänsemarsch) fortbewegender Kamele. Alle Kamele der selben Karavane bewegen sich in der gleichen Geschwindigkeit (pace). Auch eine Karavane hat eine bestimmte maximale Geschwindigkeit, die sich aus der maximalen Geschwindigkeit des langsamsten Kamels in der Karavane ergibt. Die tatsächliche Geschwindigkeit einer Karavane kann aber auch kleiner sein, bis zum Stillstand (0). Eine Karavane führt eine bestimmte Menge an Futter und Wasser mit, mit der alle Kamele der Karavane versorgt werden.

Variante D

Ein Polynom (polynomial) ist ein mathematischer Ausdruck in einer Variablen 'x' in der Form $a \cdot x^0 + b \cdot x^1 + c \cdot x^2 + d \cdot x^3 + e \cdot x^4 \dots$. Es ist eine Summe mehrerer Monome (monomial), das sind Terme der Form $c \cdot x^d$, wobei 'c' der konstante Koeffizient, 'x' die Variable und 'd' der konstante Grad des Monoms ist. 'c' und 'd' können jeden Wert annehmen (auch einen negativen Wert oder 0). Da wir davon ausgehen, dass es in allen Polynomen nur ein und dieselbe Variable 'x' gibt, kann ein Monom als Paar (c, d) von zwei konstanten Werten (constants) dargestellt werden. Wir nehmen ausserdem an, dass diese zwei konstanten Werte ganze Zahlen sind (integer constants). Zum Beispiel kann das Monom $7 \cdot x^2$ durch das Paar (7,2) dargestellt werden. Werden zwei Polynome addiert, kann man im Ergebnis die Monome mit demselben Grad zusammenfassen, indem man deren Koeffizienten addiert. Beispiel: $(3 \cdot x^0 + 2 \cdot x^2 + 1 \cdot x^3) + (7 \cdot x^2 + 1 \cdot x^3 + 5 \cdot x^4) = 3 \cdot x^0 + (2+7) \cdot x^2 + (1+1) \cdot x^3 + 5 \cdot x^4 = 3 \cdot x^0 + 9 \cdot x^2 + 2 \cdot x^3 + 5 \cdot x^4$ Diese Gleichung lässt sich auch durch Paare ausdrücken:

$[(3,0),(2,2),(1,3)] + [(7,2),(1,3),(5,4)] = [(3,0),(9,2),(2,3),(5,4)]$ Um ein Produkt zweier Monome zu bilden, müssen deren Koeffizienten multipliziert und deren Grade addiert werden. Polynome 'p1' und 'p2' können multipliziert werden, indem man für jedes Monom aus 'p1' die Produkte mit jedem Monom aus 'p2' bildet und summiert.

Aufgabenstellung

Hier werden Sie eine an den Test angepasste Auflistung aller relevanten Dateien finden:

- ApplicationTest1.java
- MCTest1.java
- ...

Im Praxisteil des Tests sollen die hier aufgelisteten Dateien bearbeitet werden:

- ...

Teilaufgabe 1 (maximal 25 Punkte)

Absolvieren Sie den Multiple-Choice-Test. Um die MC Fragen zu beantworten, editieren Sie die Datei MCTest1.java und befolgen Sie die Anweisungen aus den Kommentaren in der Datei. Wenn Sie MCTest1.java ausführen, bekommen Sie eine Fehlermeldung falls inhaltlich relevante Textteile verändert wurden; ohne Fehler werden Ihre Antworten auf die Fragen aufgelistet.

Teilaufgabe 2 (maximal 75 Punkte)

Lösen Sie den Praxisteil. [Hier werden Sie weitere an den Praxisteil des Tests angepasste Fragestellungen und Hinweise finden.] Solve the practical programming part. [Here you will find an English translation of tasks and hints adapted to the practical part of the specific test.]

Vervollständigen Sie die Klassen in den oben aufgelisteten Dateien bitte wie in den Kommentaren beschrieben an den mit TODO gekennzeichneten Stellen. Alle anderen gegebenen Dateien (außer MCTest1.java und ApplicationTest1.java) sind bereits vollständig und dürfen nicht verändert werden.

Die Klasse ApplicationTest1 enthält Testfälle mit Sollausgaben als Kommentare. Neben diesen Testfällen müssen auch alle Kommentare in den gefragten Klassen zutreffen. Die Klasse ApplicationTest1 fließt nicht in die Beurteilung ein, Sie können diese Datei nach Belieben ändern.

Bei einigen Methoden sind bei den Beschreibungen der Parameter unter @param Vorbedingungen angegeben. Diese Vorbedingungen müssen innerhalb der Methode NICHT überprüft werden, sondern stellen Zusicherungen dar, auf die sich die Methode verlassen kann.