

Constantin Lazari, Marco Wettstein

23. September 2013

1. Ziel dieser Aufgabe ist, dass Sie sich untereinander austauschen, im Buch und Ihren Unterlagen nachschlagen und sich gedanklich mit der Definition eines Algorithmus auseinandersetzen.

- (a) Spontan und ohne Hilfsmittel: Schreiben Sie alleine auf, wie sie den Algorithmus-Begriff definieren würden.

Lösung:

Ein Algorithmus ist eine vollständige Reihe von Arbeitsanweisungen, die aus einem definierten Input einen definierten Output erzeugt.

- (b) Überlegen Sie sich welche Eigenschaften einen Algorithmus ausmachen.

Lösung:

Die Arbeitsanweisungen eines Algorithmus müssen dabei folgende Kriterien erfüllen:

1. Vollständigkeit: Die Anweisungen müssen jeden auszuführenden Schritt enthalten.
2. Unmissverständlichkeit: Jede Arbeitsanweisung muss eindeutig interpretierbar sein.
3. Machbarkeit: Jede Arbeitsanweisung muss ausführbar sein. Unmögliche Anweisungen sind unzulässig.

- (c) Diskutieren Sie Ihre Definition in der Gruppe.

Lösung:

Gefallene Stichworte:

1. Regelwerk mit Ein- und Ausgabe
2. Kochrezept
3. vollständige Anweisungen
4. Verfahren zur Problemlösung
5. Beschreibung von Funktionen
6. Befehlssequenz

- (d) Überarbeiten Sie Ihre erste Version des Begriffs Algorithmus und erstellen Sie ihre eigene Definition.

Lösung:

Ein Algorithmus ist eine endliche formale Beschreibung von Funktionen oder Operationen, die aus elementaren Anweisungen oder Konstrukten besteht.

- (e) Vergleichen Sie ihre Definition mit den Unterlagen vom Theorieunterricht

Lösung:

Ergänzende Eigenschaften:

- Vollständigkeit: fraglich, ob Vollständigkeit nicht eher eine Relation zwischen Spezifikation und Algorithmus ist
- Machbarkeit: diskutabel. Es gibt Algorithmen für Quantencomputer, die in dieser Form (noch) nicht existieren.

2. Entwerfen Sie einen Algorithmus zur Berechnung der Fibonacci-Zahlen

- (a) Stellen Sie den Algorithmus graphisch dar

Lösung:

Als Struktogramm:

Fibonacci Sequenz — Mathematischer Algorithmus

Einlesen von n (die n -te Fibonacci-Zahl)

J

$n > 2$

N

Gebe zurück:
Fibonacci Sequenz mit $n - 1$
+ Fibonacci Sequenz mit $n - 2$

Gebe zurück: 1

- (b) Geben Sie eine text-basierte Darstellung Ihres Algorithmus an.

Lösung:

Algorithmus: Fibonacci Sequenz (n):

1. Falls $n > 2$: Gebe Fibonacci Sequenz ($n - 1$) + Fibonacci Sequenz ($n - 2$) zurück.
2. Sonst: Gebe 1 zurück.

- (c) Diskutieren Sie die Vor- und Nachteile der verschiedenen Darstellungsformen.

Lösung:

Ein Vorteil der graphischen Darstellung ist, dass sie eben graphisch ist. D.h. man kann mit dem Finger die Instruktionen lang fahren, und sieht, unter welchen Bedingungen man wo landet.

Der Nachteil ist, dass eine Darstellung als Struktogramm beispielsweise bei verschachtelten Wenn-Dann-Bedingungen schnell sehr „hässlich“ und unübersichtlich werden kann. Andererseits sind verschachtelte Wenn-Dann-Bedingungen grundsätzlich sowieso zu vermeiden, weil dann der Algorithmus mehr als ein Problem gleichzeitig löst.

Der Vorteil der Text-Darstellung ist, dass sie sehr leicht lesbar ist. Ein Nachteil entsteht (wie bei der graphischen Darstellung) erst, wenn der Algorithmus sehr komplex wird und im Prinzip mehr als eine Aufgabe gleichzeitig erledigt. (Beispielsweise entscheiden, ob ein Wert grösser ist als ein anderer und anschliessend aufgrund dessen etwas unternimmt.)

3. Datenobjekte

- (a) Welche Datenobjekte haben Sie für Ihren Algorithmus eingesetzt?

Lösung:

1. Elementarer Datentyp: Integer
2. Strukturierter Datentyp: Funktion

- (b) Handelt es sich um elementare oder strukturierte?

Lösung:

Sowohl als auch (siehe Antwort a)

- (c) Konstanten oder Variablen?

Lösung:

Beides sind Konstanten, die ihren Wert während eines Durchlaufs nicht verändern.

- (d) Diskutieren Sie ihre Gedanken in der Gruppe.

Lösung:

Es stellt sich die Frage, ob eine Funktion ein Datentyp ist. Aus Sicht eines Computers in jedem Fall, da es sich auch bei Funktionen letztlich auch Daten (wenn auch komplexe und mit integrierter Logik) handelt.

4. Vom Algorithmus zum Programm

- (a) Implementieren Sie den Algorithmus. Jedoch nicht ihren eigenen, sondern den einer anderen Gruppe. Nutzen Sie hierzu die erarbeitete Darstellung des Algorithmus der anderen Gruppe.

Lösung:

```
/**
 * Calculate Fibonacci-Numbers
 */
public class FibonacciLoop {

    /**
     * Calculates the Fibonacci-Number
     * at the given position.
     * @param position
     *     Position of the Fibonacci-Number
     * @return
     *     The value of the Fibonacci-Number
     *     at the specified position.
     */
    public static int get(final int position) {
        int current = 1;
        int previous = 1;
        for(int index = 3; index <= position; ++index) {
            final int temp = current + previous;
            previous = current;
            current = temp;
        }
        return current;
    }
}
```

- (b) Falls Ihr Programm nicht oder nicht korrekt läuft, dann diskutieren Sie mit der Gruppe von welcher Sie den Algorithmus erhalten haben, was die Ursache der Fehler sein könnte.

Lösung:

Läuft korrekt für ganzzahlige Werte > 0 .

Problem: Werte < 1 geben 1 statt einer Fehlermeldung zurück.