Handlungsorientierte Beschreibung von toDec(source: str, zahl: str)

Um Zahlen aus den Zahlensystemen binär, oktal oder hexadezimal, in das dezimale System umzuwandeln, hat der Autor die Funktion toDec(source, zahl) geschrieben. Um die Konvertierung von hexadezimalen Zahlen zu erleichtern, hat er ein Dictionary erstellt, in dem den Buchstabensymbolen des hexadezimalen Systems ihre jeweiligen dezimalen Werte für die weitere Berechnung zugeordnet werden,

HEXADECIMAL\_CONVERT\_FROM = {"A":10, "B":11, "C":12, "D":13, "E":14, "F":15}.

Die Funktion hat zwei Parameter, welche beide als Typ String übergeben werden. Der erste Parameter, source, übergibt die Information, aus welchem Zahlensystem umgerechnet werden soll: “b“ für binär, “o“ für oktal und “h“ für hexadezimal. Der zweite Parameter, zahl, übergibt der Funktion die umzurechnende Zahl. Die Übergabe der Zahl als String erlaubt dem Autor im späteren Verlauf der Funktion einfachere Vorgänge zu verwenden.

Als erstes überprüft die Funktion, ob die Zahl 0 übergeben wurde. Da diese Zahl in jedem Zahlensystem dieselbe ist, wird in diesem Fall einfach 0 zurückgegeben und keine Berechnungen finden statt.

Danach wird die lokale Ergebnisvariable result initialisiert und auf 0 gesetzt.

Da in Zahlensystemen nach Stellenwert von rechts nach links in aufsteigender Potenz der Basis dargestellt wird, geht der Autor auch bei der Berechnung von rechts nach links. Um trotzdem in folgenden Schleifen mit aufsteigenden Indizes zu arbeiten, wird die übergebene Zahl im nächsten Schritt mit list(zahl) in eine Liste konvertiert und gespeichert, und dann hat der Autor die Liste mit zahl[::-1] von rechts nach links umgedreht.

Im nächsten Schritt unterscheidet der Autor mithilfe einer Fallunterscheidung zwischen den drei verschiedenen Zahlensystemen. Welcher Fall gewählt wird, ist durch den Parameter source im Funktionsaufruf bereits vorgegeben.

Wenn der Parameter source “b“ ist, wird der Fall für das Umrechnen von binären Zahlen ausgewählt. Hierfür iteriert der Autor mit einer for-Schleife durch alle Indizes der Liste zahl, entnimmt mit int(zahl[i]) den entsprechenden Wert als Integer, multipliziert diesen mit der Basis 2 und dem Exponenten i der Laufvariable, entsprechend dem Stellenwert. Das Ergebnis dieser Iteration addiert der Autor dann zu der lokalen Variable result. Nachdem alle Iterationen abgeschlossen sind hat der Autor alle Ergebnisse zu dem endgültigen Ergebnis, der entsprechenden Dezimalen Zahl, aufaddiert, und gibt diese mit return result zurück.

Wenn der Parameter source “o“ ist, wird der Fall für das Umrechnen von oktalen Zahlen ausgewählt. Der Vorgang ist hier fast gleich, der Autor muss nur in der Berechnung die Basis 2 durch die Basis 8 ersetzen. Dann wird wieder das aufaddierte Ergebnis mit return result zurückgegeben.

Eine Besonderheit besteht, wenn der Parameter source “h“ ist, also das Umrechnen von hexadezimalen Zahlen ausgewählt wurde. Bevor hier die gleiche Umrechnungsfunktion wie im Fall binär oder oktal benutzt werden kann, indem die Basis wieder entsprechend auf 16 geändert wird, müssen zuerst Buchstabensymbole aus der hexadezimalen Darstellung in ihre entsprechenden Zahlen im dezimalen System geändert werden, damit der Autor mit diesen Zahlen rechnen kann. Dafür iteriert der Autor mit einer for-Schleife durch jedes Element der Liste zahl und überprüft mit der Methode isdigit(), ob es sich um eine Zahl oder einen Buchstaben handelt. Wenn es sich um einen Buchstaben handelt, so wird dieser zuerst mit der capitalize() Methode groß geschrieben. Der Autor vermeidet damit Fehlermeldungen in der folgenden Zuordnung von Buchstaben zu Zahl mithilfe des am Anfang definierten Dictionaries HEXADECIMAL\_CONVERT\_FROM. Das Listenelement wird dann mit der entsprechenden Zahl aus dem Dictionary überschrieben. Jetzt enthält die Liste nur noch Zahlen, und der Autor kann die gleiche Umrechnungsfunktion mit der Basis 16 anwenden und das aufaddierte Ergebnis wieder mit return result zurückgegeben.