

Praktikum am Gymnasium im Schloß Wolfenbüttel

Thomas Wassong

Institut für Informatik
Seminar: Fachdidaktik der Informatik

17. Mai 2006

Übersicht

- 1 Eine Prozedur
- 2 Verzweigungen
- 3 Schleifen
 - Zählschleifen
 - Schleifen mit Bedingungen
- 4 rekursives Programmieren

Thomas Wassong Praktikum am Gymnasium im Schloß Wolfenbüttel 17. Mai 2006

Prozeduren in MuPAD

- Eine Prozedur hat immer den Rahmen `proc() begin ... end_proc;`
- In den Klammern von `proc()` werden die Übergabeparameter definiert.
- Mit `return` wird der Rückgabewert definiert. Die Prozedur wird danach beendet.
- Beispiel in MuPAD: Addition von zwei Zahlen

Thomas Wassong Praktikum am Gymnasium im Schloß Wolfenbüttel 17. Mai 2006

Aufgabe

Programmieren Sie eine Prozedur `mult_drei(a,b,c)`, die drei Zahlen miteinander multipliziert und das Ergebnis mit `return` zurückgibt.

Thomas Wassong Praktikum am Gymnasium im Schloß Wolfenbüttel 17. Mai 2006

Verzweigungen

- In MuPAD werden Verzweigungen mit `if (Bedingung) then ... end_if;` erklärt.
- Die Bedingung muss einen Wahrheitswert zurückgeben.
- Die Anweisungen nach dem `then` werden ausgeführt, wenn die Bedingung erfüllt ist.
- MuPAD-Beispiel: Maximum von zwei Zahlen.
- In MuPAD stellt auch einen `else`-Zweig zur Verfügung: `if (Bedingung) then ... else ... end_if;`
- Dieser wird betreten, wenn die Bedingung nicht erfüllt wird.
- MuPAD-Beispiel: Ist eine Zahl eine Primzahl?

Thomas Wassong Praktikum am Gymnasium im Schloß Wolfenbüttel 17. Mai 2006

Aufgabe

Programmieren Sie eine Prozedur `Betrag(a)`, die den Betrag einer Zahl zurückgibt. Nutzen Sie dabei eine Verzweigung.

Thomas Wassong Praktikum am Gymnasium im Schloß Wolfenbüttel 17. Mai 2006

Zählschleife

- In MuPAD wird die Zählschleife folgendermaßen definiert: `for zähler from start to ende do ... end_for;`
- Dabei wird die Schleife für jede ganzzahlige Zahl zwischen `start` und `ende` durchlaufen. Die erste Durchgang ist für `zähler=start`, der zweite für `zähler=start+1`, dann `zähler=start+2` usw. bis zum letzten Durchgang mit `zähler=ende`.
- MuPAD-Beispiel: $\sum_{i=1}^n i$.

Thomas Wassong Praktikum am Gymnasium im Schloß Wolfenbüttel 17. Mai 2006

Aufgabe

Programmieren Sie eine Prozedur `Fakultaet(n)`, die das Produkt der ersten `n` Zahlen berechnet. Dabei soll die Zahl `n` als Eingabeparameter übergeben werden. Nutzen Sie eine Zählschleife.

Thomas Wassong Praktikum am Gymnasium im Schloß Wolfenbüttel 17. Mai 2006

Aufgabe

Programmieren Sie eine Prozedur `Fib(n)`. Diese soll die n -te Fibonacci-Zahl berechnen. Die Fibonacci-Zahlen sind folgendermaßen definiert:

$$\begin{aligned}a_1 &:= 1, \\ a_2 &:= 1, \\ a_n &:= a_{n-1} + a_{n-2}.\end{aligned}$$

Schleifen mit Bedingungen

- Neben den Zählschleifen gibt es auch Schleifen, die solange eine bestimmte Bedingung erfüllt ist, sich wiederholen.
- In MuPAD sind durch eine `while`-Schleife definiert: `while Bedingung do ... end_while;`
- Dabei wird bei jedem Durchlauf (auch dem ersten) geprüft, ob die `Bedingung` erfüllt ist. Wenn dies der Fall ist, so werden die nachfolgenden Anweisungen bis `end_while` ausgeführt und erneut geprüft. Wenn dies nicht der Fall ist, so geht es mit den Anweisungen nach `end_while` weiter.
- MuPAD-Beispiel: euklidischer Algorithmus des ggT.

Aufgabe

Programmieren Sie eine Modulo-Funktion `Mod(a, b)`. Diese berechnet, was der Rest einer ganzzahligen Division $a \div b$ ist. Nutzen Sie dabei eine `while`-Schleife, in der solange b von a subtrahiert wird, bis $a < b$ gilt.

Einschub

Es sind alle programmierten Prozeduren schon in MuPAD (effizienter) programmiert!

rekursives Programmieren

- Bisher haben wir iterativ programmiert, das heißt wir konnten die Prozedur, das Programm von oben nach unten lesen.
- In einigen Fällen ist es jedoch naheliegender, rekursiv zu programmieren. Das bedeutet, dass sich eine Prozedur selbst aufruft.
- MuPAD-Beispiele: ggT und Fakultät.

Aufgabe

Programmieren Sie die Funktion `pot_mod(a, n, p)`. Diese berechnet die n -te Potenz von a modulo p mit folgender Formel:

$$a^n \bmod p = \begin{cases} a^0 \bmod p & \text{für } n=0 \\ (a^{n/2} \bmod p)^2 \bmod p & n \text{ gerade} \\ (a^{n-1} \bmod p) * a \bmod p & \text{sonst} \end{cases}$$

Nutzen Sie dabei die rekursive Programmierung. Tipp: mit `is(n, Type: :Even)` erfahren Sie, ob n gerade oder ungerade ist.

Aufgabe

Programmieren Sie eine rekursive Prozedur `Fib(n)`. Diese soll die n -te Fibonacci-Zahl berechnen. Die Fibonacci-Zahlen sind folgendermaßen definiert:

$$\begin{aligned}a_1 &:= 1, \\ a_2 &:= 1, \\ a_n &:= a_{n-1} + a_{n-2}.\end{aligned}$$