

Imperative Programmierung

Aufgabenblatt 7

Hinweise:

Bearbeitungszeit: 2 Wochen, Abgabefrist: Siehe Juniorstud.IP

Ladet eure Lösungen der Aufgaben unter "Aufgaben" im Juniorstud.IP als PDF-Dokument bzw.

C-Quelltext oder als txt-Datei hoch. Aufkommende Fragen bitte im Forum stellen, damit diese dort für alle geklärt werden können. Bei verspäteter Abgabe wird je Tag 10% der zu erreichenden Gesamtpunktzahl abgezogen.

1. Die Fibonacci-Zahlen $fib(i)$ sind wie folgt definiert:

$$\begin{aligned}fib(0) &= 0 \\ fib(1) &= 1 \\ fib(n) &= fib(n-1) + fib(n-2)\end{aligned}$$

Schreibt auf dieser Basis ein rekursives Programm, das die ersten Fibonacci-Zahlen für $n = 0 \dots 50$ ausgibt. (Ihr müsst hier mit 64-bit Zahlen arbeiten: `double` oder `(long)longint`). Gebt an, wie lang euer Programm läuft.

5 Punkte

2. Schreibt ein Programm, das viel schneller arbeitet, als das Programm aus Aufgabe (1). (Ihr könnt euch zum Beispiel bereits berechnete Zahlen in einem Feld merken.)

10 Punkte

3. Betrachtet die folgende Funktionsdefinition:

```
double fnord(double v) {
    int c = getchar();
    if(c >= '0' && c <= '9') return fnord(v*10+(c-'0'));
    ungetc(c, stdin);
    return v;
}
```

Erläutert, was diese Funktion macht und wie sie es macht.

5 Punkte

4. Schreibt eine neue Version von `fnord`, die eine Schleife statt einer Rekursion nutzt.

5 Punkte

5. Fakultät, "beliebig" genau. Schreibt ein Programm, das ein iteratives Verfahren zur Berechnung der Fakultät verwendet und dabei $n!$ exakt ausrechnet.

Hinweis: Legt zur Speicherung des aktuellen Wertes von $n!$ ein `int`-Feld an, bei dem jedes Feldelement zum Beispiel 5 Dezimalstellen von $n!$ speichert und verwendet entsprechend das Verfahren der schriftlichen Multiplikation.

15 Punkte