

# **Übung 9**

**Ausgabe:** Di, 19.12.2000

**Abgabe in** Mi, 10.01.2001  
**den Gruppen:** Do, 11.01.2001

**Besprechung in** Mi, 17.01.2001  
**den Gruppen:** Do, 18.01.2001

**Aufgabe 9.1:** Aufwandsabschätzung (5 Punkte)

Gegeben sind die folgenden Funktionen EMPotenz und SQPotenz, welche eine Potenz  $b^n$  durch einfaches Multiplizieren beziehungsweise sukzessive Quadratbildung (siehe auch Aufgabe 5.2) berechnen:

```

PROCEDURE EMPotenz(b: REAL;
    n: CARDINAL): REAL =
VAR erg: REAL := 1.0;

BEGIN
    FOR i := 1 TO n DO
        erg := erg * b;
    END;

    RETURN erg;
END EMPotenz;
```

```

PROCEDURE SQPotenz(b: REAL;
    n: CARDINAL): REAL =
VAR rekErg: REAL;

BEGIN
    IF n = 0 THEN
        RETURN 1.0;
    ELSIF n MOD 2 = 0 THEN
        rekErg := SQPotenz(b, n DIV 2);
        RETURN rekErg * rekErg;
    ELSE
        RETURN b * SQPotenz(b, n - 1);
    END;
END SQPotenz;
```

Schätzen Sie jeweils den Aufwand (Anzahl der notwendigen Multiplikationen) in Abhängigkeit des Exponenten  $n$  für beide Funktionen. Bestimmen Sie für die Funktion SQPotenz eine untere und eine obere Schranke für die Anzahl der benötigten Multiplikationen im günstigsten beziehungsweise schlechtesten Fall. Vergleichen Sie die Ergebnisse: Um welchen Faktor weniger Multiplikationen werden für  $n = 1024$  (bzw.  $n = 1023$ ) bei sukzessiver Quadratbildung gegenüber der einfachen Multiplikation benötigt?

**HINWEIS:** Die Operationen DIV und MOD seien ohne Multiplikation realisiert und tragen daher nichts zum Aufwand bei.

**Aufgabe 9.2:** Dateien (6 Punkte)

Erweitern Sie das Programm Termine aus der Musterlösung zu Aufgabe 6.2 so, dass die eingegebenen Termine in einer Datei gespeichert und später wieder gelesen werden. Implementieren Sie zu diesem Zweck geeignete Prozeduren, um die Daten des Terminkalenders einerseits in eine Datei zu schreiben und andererseits wieder zu lesen. Erweitern Sie anschließend das Programm so, dass es mit den implementierten Prozeduren zu Beginn die Termine aus einer lokalen Datei liest und am Ende des Programms wieder in dieser Datei speichert. Geben Sie das vollständige Programm und eine Datei mit mindestens drei bereits gespeicherten Terminen als Lösung ab.

**HINWEIS:** Verwenden Sie die Operationen des Moduls SIO, um die Termindaten zu speichern und zu lesen, indem Sie die Aus- bzw. Eingabe auf eine sequentielle (Text-)Datei, wie in der Globalübung vorgestellt, umstellen.

### Aufgabe 9.3: Testen (9 Punkte)

Führen Sie einen systematischen Test des Programms aus Aufgabe 6.3 durch. Verwenden Sie dazu die speziell instrumentierte Implementierung des Programms `Meldeamt`, welche zusammen mit dem Übungsblatt auf den WWW-Seiten bereitgestellt wird.

- a) Entwerfen Sie geeignete Blackbox-Testfälle, um die Eingabe eines Gemeldeten zu testen. Verwenden Sie als Spezifikation die Aufgabenstellung aus Übungsblatt 6 in Kombination mit folgenden Anforderungen:
- sämtliche Namen seien vom Typ `TEXT` und dürfen, wenn eine Eingabe zwingend ist, nicht leer sein
  - ein Tagesdatum muss laut dem gregorianischen Kalender gültig sein, z.B. der 30.02.2000 ist kein gültiges Datum. Die Werte können jedoch beliebig, auch ohne führende Null, eingegeben werden, solange sie im Wertebereich 01.01.1800 bis 31.12.3000 liegen.
  - Hausnummern sind auf den Bereich 1 bis 1000 beschränkt
  - Postleitzahlen sind auf den Bereich 0 bis 99999 beschränkt
  - ist der Familienstand verheiratet, fragt das Programm zusätzlich nach den Namen des Ehepartners, während bei Familienstand ledig, verwitwet oder geschieden nichts weiter passiert

Versuchen Sie Eingabeüberdeckung zu erreichen. Bestimmen Sie dazu sinnvolle Äquivalenzklassen für die einzugebenden Datenwerte und ermitteln Sie mit Hilfe einer einfachen Grenzwertanalyse dieser Äquivalenzklassen passende Testfälle. Definieren Sie zu jedem Testfall das Soll-Resultat. Nummerieren Sie sowohl die gefundenen Äquivalenzklassen als auch die Testfälle eindeutig und erstellen Sie eine Tabelle, welche für jeden Testfall die durch ihn überdeckten Äquivalenzklassen angibt.

HINWEIS: Nur die Dateneingabe soll getestet werden, die Menüsteuerung des Programms und die Datenausgabe (außer für Vergleichszwecke) brauchen nicht weiter berücksichtigt werden. Der Typ `TEXT` sei nicht begrenzt in seiner Größe.

- b) Führen Sie die instrumentierte Version des Programms `Meldeamt` mit den in Teil a) entwickelten Testfällen aus und protokollieren Sie die Testergebnisse. Wo gibt es Abweichungen zwischen Soll- und Ist-Resultat? Wieviel Prozent der Zweige werden dabei überdeckt?
- c) Zeichnen Sie das Flussdiagramm für die Prozedur `FamilienstandEingeben()` des Programms `Meldeamt`. Ermitteln Sie anschließend geeignete Testfälle, die für eine Zweigüberdeckung (C1 Test) der Prozedur notwendig sind. Geben Sie jeweils den erwarteten Rückgabewert der Prozedur als Soll-Resultat an.
- d) Zeichnen Sie das Flussdiagramm für die Prozedur `PersonEingeben()` des Programms `Meldeamt`. Ermitteln Sie anschließend Testfälle, die für eine eingeschränkte Pfadüberdeckung der Prozedur notwendig sind. Geben Sie jeweils den erwarteten Rückgabewert der Prozedur als Soll-Resultat an.