

Aufgabe 2.1

a)

Ein LOOP-Programm besteht aus einer endlichen Aneinanderreihung von Anweisungen. Wertzuweisungen sind offensichtlich in endlicher Zeit ausführbar. LOOP-Anweisungen überprüfen die übergebene Variable und wiederholen das jeweilige Teilprogramm entsprechend oft. Da Variablen nur endliche Werte haben können, kann das Teilprogramm auf diese Weise auch nur endlich oft wiederholt werden, es muss also in endlicher Zeit ausführbar sein. Da also alle Anweisungen eines LOOP-Programmes in endlicher Zeit ausführbar sind, muss jedes LOOP-Programm in endlicher Zeit terminieren.

b)

```
P ::=
  x0 := 1;
  WHILE x0 ≠ 0 DO x0 := x0 + 1 END
```

c)

Sei p eine natürliche Zahl.

```
LOOP:
P ::=
  x0 := p;
  LOOP x0 DO Q END
```

```
WHILE:
P ::=
  x0 := p;
  WHILE x0 ≠ 0 DO Q; x0 := x0 - 1 END
```

di)

```
P ::=
  result := y + 0;
  x1 := x + 0;
  WHILE x1 ≠ 0 DO
    x1 := x1 - 1;
    result := result + 1;
  END
```

Ergebnis: *result*

dii)

```
P ::=
  result := 0;
  xMult := x + 0;
  WHILE xMult ≠ 0 DO
    yAdd := y + 0;
    xMult := xMult - 1;
    WHILE yAdd ≠ 0 DO
      yAdd := yAdd - 1;
      result := result + 1;
    END
  END
END
```

Ergebnis: *result*

diii)

```
P ::=
  result := 0;
  WHILE x ≠ 0 DO
    result := 1;
    yPot := y + 0;
    WHILE yPot ≠ 0 DO
      yPot := yPot - 1;
      xMult := x - 1;
      tempRes := result;
      WHILE xMult ≠ 0 DO
        xMult := xMult - 1;
        resAdd := tempRes + 0;
        WHILE xMult ≠ 0 DO
          resAdd := resAdd - 1;
          result := result + 1;
        END
      END
    END
  END
END
```

Ergebnis: *result*