Universität Bremen

Programmiersprachen WS 2021/22 Christoph Lüth

2. Übungsblatt

Ausgabe: 01.11.21

2.1 *While und If*

Für die Turing-Mächtigkeit einer Programmiersprache reichen neben veränderlichen Variablen entweder die Iteration, oder Fallunterscheidung und Rekursion.

Implementieren Sie in C, Python oder Java eine Fallunterscheidung mit Hilfe von `while`. (Sie werden dabei Hilfsvariablen brauchen.)

In Haskell stellt sich die Frage anders herum: Rekursion und Fallunterscheidung sind schon vorhanden, implementieren Sie also eine Funktion

```
while :: (a-> Bool)-> (a-> a)-> a-> a
```

mit den offensichtlichen Eigenschaften (welche schon eine valide Haskell-Definition darstellen).

2.2 *Funny Functions*

Betrachten Sie die folgenden — absichtlich etwas kryptische — wechselseitig rekursiven Funktionen `f` und `g` (Quellcode auf der Webseite, auch in Python erhältlich):

```
static int z= 0;

static int f(int x1, int y1)
{
  if (x1 == 0 && y1 == 0) {
      return z;
  } else if (x1 == 0) {
      z= z+1;
      return g(y1, y1);
  } else {
      x1= x1-1;
      z= z+1;
      return f(x1, y1);
  }
}
static int g(int x2, int y2)
{
    if (x2 == 0 && y2 == 0) {
      return z;
    } else if (y2 == 0) {
      x2= x2-1;
      z= z+1;
      return f(x2, x2);
    } else {
      y2= y2-1;
      z= z+1;
      return g(x2, y2);
    }
}
```

Skizzieren Sie für den Aufruf `f(3,2)` Lebensdauer und Sichtbarkeiten der Variablen `z`, `x1`, `y1`, `x2`, `y2` wie in der Vorlesung.

2.3 *Deklarationen*

Wie werden Deklarationen in C, Java, Python, Haskell gehandhabt? Sind sie

- sequentiell,
- kollateral,
- rekursiv,

oder eine Mischung daraus?