プログラム言語論課題 3

2019 年 7 月 5 日

櫻井彰人

1. Call-by-name を用いると（上手に使えば）プログラムが簡潔に記述できることが知られています（しかし、副作用の多さからあまり用いられていません)。Jensen's device はその一つです。ところで、Call-by-name がなくとも Jensen's device が実装できることが知られています。その例（いずれも調和数（harmonic number）を求めるプログラムです）が http://rosettacode.org/wiki/Jensen's_Device に記されています。
(1-1) C のマクロ機能を使うと容易に（Algol 同様に）書けることはすぐ分かります。C そのものでも tricky な工夫をすることにより可能なことが示されています。これらが目的通り動作する理由を説明して下さい。
(1-2) プログラム例がいろいろな言語で書かれています。このうち、C, C# 以外の言語を取り上げ、それが目的通り動作する理由を説明して下さい。取り上げる言語はなんでもよいのですが、プログラム例が C とあまり変わらない言語は避けて下さい。

2. プログラミング言語の型概念（そして抽象データ型）について、800 字程度にまとめてください。型概念に関する解説は本一冊でも足りないでしょうから、800 字にまとめるなど正気の沙汰ではありません。が、しかし、ある観点（自分で考えた、選んだ観点で結構です。例えば歴史的観点や抽象化の観点）からまとめて下さい。
また、オブジェクト指向のオブジェクト（クラス）との関係を、やはり、800 字程度にまとめて書いて下さい（ですから、この問題前半では、オブジェクト指向との関係については、原則的には、書かないでください)。

次の 3. と 4. については、どちらか一問に答えて下されば結構です。両方に答えて下さるのも、もちろん歓迎！

3. Ruby で書かれた次のプログラムは階乗を計算するメソッドです。どうして階乗が計算できるかを説明してください。

(1) `def factorial(n,f=1) if n<=1 then f else factorial(n-1,n*f) end end`

(2) `def factorial(n) (1..n).inject{|x,y| x*y} end`

(3) `def factorial(n) eval( [*(1..n)].join("*") ) end`

４. 自分の興味をもったプログラミング言語を一つとりあげ、それについて調べ、その構文的特長、意味的特長（何か特別な意味をもった要素を実装しているなど)、設計意図を、特に他の言語との違いを強調して、報告して下さい。なお、マイナーな言語を対象として下さい。何がマイナーか（メジャーか）は皆さんの判断に任せます。この課題の目的は、実際の未知のプログラミング言語について調べ、その分析を試み、その結果を報告することです。

提出期限は、7 月 29 日(月曜日）午前 8 時です。できれば、TeX または MsWord で作成してください。そして、これを、**圧縮**してメールにて（アドレスはこれまで同様）送って下さい。