



Haskell言語超入門、参照透過性のある関数の価値

竹花 忠

Copyright (C) 2019 TAKEHANA TADASHI

著作日時: 2019.10.07月. 08:09:00 著作者、竹花 忠

Haskell言語、超入門、参照透過性のある関数の価値:

純粋に単純に返り値を返すだけの関数であることの価値:

関数から関数の返り値だけを得ることを目的として関数を使用したい。返り値を返して寄越すこと以外には、なにも変化を生じさせて欲しくない。そのような目的で、そのような用途で、関数を使用したい。

それに正にぴったり応えてくれるのが、参照透過性のある関数である。

返り値を返す以外には、外部に一切の変化を残さず、関数とその処理を終えれば、その関数内部のデータもすべて消滅する。返り値を返すこと以外は一切なんの痕跡も残さない、参照透過性のある関数は。

なので、安心して、返り値の獲得だけに注目して、返り値の受け渡しだけに集中して、それ以外に変化を生じさせている可能性に対して一切配慮する必要なしに、その関数を使用してゆける。

それが参照透過性のある関数の価値である。

その関数が返す返り値だけに、着目して・注目して、それ以外には一切変化を生じさせないことが確信できて保証されて、返り値だけに、着目して・注目して、その関数を使いまわしてゆける。

そのことによって、返り値をもたらしている以外に、ほかのどこかをどうにかさせてしまっていることがない。外部には何の影響も与えていないことが保証されていて安心して使用してゆける、使いまわしてゆける。

それが、参照透過性のある関数の価値である。

つまり、思考を単純に保ったまま思考を進めてゆけることになるのが、参照透過性のある関数の価値である。

参照透過性のない関数であったなら、返り値が得られたことのほかに、変数Aを5に書き換え、変数Bは1カウントアップし、というようなことを伴う。

すると、それが必要な状況でその関数を使用する時には、変数Aの内容を事前に保存しておいて、事後に書き戻すことをしなければ、という、配慮・処理、が必要になるし、変数Bは事後に1カウントダウンさせる、配慮・処理、が必要になる。

参照透過性のない関数だと、関数を使用する際に、考えるべきことが、増大してしまう・膨れ上がってしまう。つまり、思考が安易に複雑化させられていってしまう。

参照透過性のある関数はそれを食い止めてくれる。参照透過性のある関数だと、返り値がもたらされること以外には、外部状況・周辺状況、対して、一切配慮しないで・一切心配しないで、もたらされる返り値だけに、注目・着目、して、その関数を自由に使用してゆける。

それが参照透過性のある関数の価値である。

