引数

出典: フリー百科事典『ウィキペディア(Wikipedia)』

引数(ひきすう、いんすう[1]、*Parameter*)とは、コンピュータープログラミングで用いられる用語で、**仮引数**と**実引数**の総称である。仮引数はサブルーチンやメソッド等のプロシージャー定義の部分で定義されるもので、実引数はプロシージャーを呼び出すときに指定するものである。実行時には、実引数の値を仮引数が受け取る。「引数」の読み方は、変則的に湯桶読みして「ひきすう」とするのが一般的である。数学分野で因数との取違えを防ぐためといった理由もある(数学では、parameterに一般に「媒介変数」の訳語をあてるが、「引数」を用いることもある)。

目次

仮引数

仮引数(かりひきすう、かびきすう、parameter、formal parameter (argument))とは、プロシージャーで定義される変数のうち、実行時に呼び出し元から渡される(実引数の)値を受けるものをいう。例としてC言語系言語における定義を挙げる:

```
int sum(int addend1, int addend2)
{
    return addend1 + addend2;
}
```

上の定義では、

- int 型の仮引数 addend1
- int 型の仮引数 addend2

2つを伴った関数 sum を定義している。定義の中で addend1 と addend2 が変数のように使用されていることに注目されたい。

実引数

実引数(じつひきすう、argument、actual argument (parameter))とは、プロシージャーを呼び出す際に渡す値のことで、プロシージャーの挙動(動作や結果)に作用する。変数やリテラルを含む式を指定できる。C言語系言語において前に示した例中の関数 sum を用いた例を挙げる:

```
sum(123, 456);
```

上の文は、

- 仮引数 addend1 に対応する実引数 123
- 仮引数 addend2 に対応する実引数 456

2つを関数 sum に渡している[2]。

値渡し

値渡し(あたいわたし、call by value)は右辺値を渡す方法で、実引数として変数を渡したとしても、その値のみが渡される。もちろん即値や複雑な式を渡すこともでき、式の評価結果が渡される。その仕組みとしては、独立した新たな変数が関数内に用意され、元の値がコピーされる。そのため変数を渡したとしても、元の変数が変更されるという事はない。

これは「関数が副作用を持たない」という観点から、計算を中心とする言語では望ましい動作といえる。またそもそも代入概念のない関数型言語では、引数は必ず値で渡されると考えられる(ただし、代入が存在しない以上コピーをとる必要もない)。

値渡しを採用した言語としてはC言語、ML、APL、Scheme、Java等が挙げられる。

ポインタ渡し

C言語やC++のポインタ変数が保持する値は変数に対する参照(メモリアドレス)であり、後述の参照渡しの参照と似た性質を持つ。このため、ポインタ変数を値渡しすると、値渡しでありながら参照渡しと似たような効果を得ることができる。このため、ポインタ(=メモリアドレス)を値渡しする事を単なる値渡しと区別して俗に**ポインタ渡し**などと呼ぶ事もある。

変数渡し

変数渡し(へんすうわたし、call by variable)は、変数そのもの(左辺値)を渡す方法で、この場合は仮引数に対する操作がそのまま実引数(渡された変数)に影響する。

参照渡し(さんしょうわたし、call by reference)はその実装手段の一つで、変数に対する参照(アドレス情報)を渡す方法である。(これは言語側が勝手に行う。C言語のように明示的にアドレス演算子を使うものは参照渡しとは呼ばない。) その他、値渡しと同じようにコピーを渡しておいて、関数/サブルーチンからのリターン時に元の変数に変更結果をコピーしなおす方法もある。PL/Iでは、どちらの方法で実装しても良いと規定されている。

原始的な言語であるFORTRANは機械語のアドレス操作を反映した参照渡ししか持たなかった。これは特にcall by indexと呼ばれている。 他に変数渡しをサポートする言語としては、Pascal、Perl、C++、C#、Quick BASIC等の構造化BASICなどが挙げられる。

なお変数渡しの関数・サブルーチンに、実引数として変数以外(右辺値)を渡した場合にどうなるかは、言語によって異なる。そのような操作が禁止されており、エラーが発生する言語(Pascal、C++等)、テンポラリな変数を作成し、リターン時にそれを捨ててしまうため、値渡しと同じことができる言語(Quick BASIC等)、「未定義の動作」をひきおこし、何が起こるか全く予測がつかない言語(FORTRAN等)がある。

値渡しによる引数の変更

C言語は値渡しのみをサポートするが、変数のポインタ(メモリアドレス)を取得することが可能であるため、変数へのポインタを値で渡す事で元の変数を変更できる。オフセット計算により配列や構造体の一部分を参照するコードも容易に記述できる。

しかしこれは、実際の変数領域を逸脱した部分をも参照できるので、あくまでも値渡しによる参照渡しのエミュレートである。参照渡しをサポートする言語でも内部的には同様の操作を行っているが、それは何らかの意味で言語の保護下にある参照となる。

参照の値渡し

参照渡しで言うところの「参照」と呼ばれているものと、特定の言語で「参照」と呼ばれているものが必ずしも同じでない事には注意が必要。例えば、Javaは参照型を扱うための『Javaの「参照」』を持つが、これはPascal等のポインタ相当で、『参照渡しの「参照」』とは概念が違うため、『Javaの「参照」』を渡しても参照渡しであるとは言えない。C言語の「ポインタの値渡し」と同じである[3]。これは、Javaの参照型と似た参照型と、Javaのプリミティブ型に近い値型を持つC#を見ると理解しやすいだろう。C#では、特に指定しなければ参照型も値型も値渡しされるが、引数に `ref` もしくは `out` を使用する事によって参照渡しにする事ができる。『C#の値型』を渡すから値渡し、『C#の参照型』を渡すから参照渡しとはならない。

例:Pascalの変数渡し

Pascalの手続き(procedure)や関数(function)では、原始型(integer, realなど) の値渡しと変数渡しのどちらでも行える。変数渡しの場合は手続き・関数の引数にvarを付ける。

```
{ 手続き swap 内で a,b の値を入れ替える。
  sampleの i,j は変数渡しされ、aとi、bとjは同じアドレスを指して
  いるので、i,jの値は入れ替わる。 }

procedure swap(var a,b:integer);  { var をつけると変数渡し }
  var tmp:integer;
  begin
    tmp := a; a := b; b := tmp
  end;
```

```
procedure sample();
  var i, j:integer;
  begin
    i := 5;
    j := 10;
    swap(i, j);
    ... { iは10, jは5になる }
  end;
```

遅延評価

詳細は「遅延評価」を参照

Haskellなどの遅延評価型関数言語に見られる形態で、値が実際に必要になるまで計算を行わない方法。概念上は、計算方法を遅延した**thunk**と呼ばれるオブジェクトが渡っていると考えられる。

名前渡し

ALGOLで採用されていた特徴的な機能の一つである。名前渡しでは値でも参照でもなく、式がそのまま渡される。基本的には参照渡しのように振る舞うが、式を参照するごとに値を計算して取り出す事が特徴である。C言語のプリプロセッサのマクロ展開と似ているが、引数と、ローカル変数が衝突しないように配慮はされる。 次のような例は名前渡しに特徴的な動作と言われる。

```
swap(x,y) {
 tmp = x;
 x = y;
 y = tmp;
}
```

この例に対し、x=i, y=a[i]という"式"を渡すとする。仮にi=2だったとすると、

tmp = x;
　x=i=2 なのでtmpは2になる。
x = y;
　xはiを渡されているのでiがyの値になる。yはa[i]だから、iはaの2番目の値になる。
y = tmp;
　yはaのi番目の値だが、前手順によりiはa[2]になっている。従ってy=a[a[2]]になる。

このような複雑さもあって、ALGOL以外で名前渡しが採用された事例はほとんどない。

脚注

1. ^ 引数;引き数 (http://www.edrdg.org/jmdictdb/cgi-bin/entr.py?svc=jmdict&sid=&e=15967) JMdictDB - Japanese Dictionary Database
2. ^ ちなみに、上の文の結果は 579 である
3. ^ JavaHouse-Brewers の議論 (http://java-house.jp/ml/archive/j-h-b/026214.html#body)

関連項目

- 変数 (プログラミング)
- 多重定義
- 評価戦略
- 呼出規約

この項目は、コンピュータに関連した**書きかけの項目**です。この項目を加筆・訂正(https://ja.wikipedia.org/w/index.php?title=%E5%BC%95%E6%95%B0&action=edit)などしてくださる協力者を求めています(PJ:コンピュータ/P:コンピュータ)。