ESライブラリ＆＆
ゲームプログラミング

○　数学基礎入門 - 端数処理　○

端数処理

端数処理とは、数値の「ある位」以下を端数として、なくしてしまう処理のことです。たとえば、「ある位」を小数点として端数処理を行うとします。3.103の場合、小数点以下がなくなるので、３または４になります。端数をなくすだけではなく、端数があったことを反映させるため、前後の数に増減させる方法もあります。よって、3.103を４にする場合もあります。どの方法を採用するのかは、そのときの状況に合わせます。

端数処理は丸め(round)とも呼ばれ、よく使われる処理としては、「切り捨て」「切り上げ」「四捨五入」があります。代表的なものが、小数以下をなくしたり、小数第１位や２位で四捨五入する、というものです。

切り捨て

ある位以下をなくしてしまう処理、つまり「切り捨てる」処理です。指定された位以下はなくなってしまうため、基本的には、数が減る処理といえます。

ここで、単純に小数を切り捨てて整数のみを取り出すことを考えてみます。「3.103」という数の場合、小数以下を切り捨てると「３」という整数になり、0.103減りました。負の数「-3.103」という数では、小数以下を単純に捨てると「－３」になり、0.103ほど「増えて」しまいます。数を「捨てた」のに増えたということです。単純な切り捨てを行うと、正の数では減り、負の数では増えます。

切り捨て処理で、数が減るように処理することを数学では「床関数」(floor function)と呼びます。床関数では、正の数では単純な切り捨て、負の数では単純な切り上げが行われます。つまり「3.103」は「３」に、「-3.103」は「－４」になります。

なお、最初に出てきた「単純な切り捨て」のことを「切り落とし」(truncate)と呼びます。

切り上げ

ある位以下が０でない場合、(端数を切り捨てた後)次の数へ増やす、というものです。少しでもはみ出れば次の数に「切り上がる」ため、数が増える処理といえます。

ここで、小数以下を単純に整数へ切り上げることを考えてみます。「3.103」という数の場合、小数以下は切り上げられるので「４」という整数になり、数が増えます。負の数「-3.103」という数では、単純に切り上げると「－４」になり、数が「減って」しまいます。単純な切り上げを行うと、正の数では増え、負の数では減ります。

切り上げ処理で、数が増えるように処理することを数学では「天井関数」(ceiling function)と呼びます。天井関数では、正の数では単純な切り上げ、負の数では切り落としが行われます。つまり、「3.103」は「４」に、「-3.103」は「－３」になります。

四捨五入

端数の最初の位が５以上かによって、「切り捨て」または「切り上げ」を選択するものです。名前のとおり、４以下なら切り捨て、５以上なら切り上げます。

四捨五入が提供されていない環境では、代替処理として「端数に５を足したあと、端数を切り捨てる」ことがよく行われます。このとき、負の数では絶対値が減ってしまうことに注意する必要があります。たとえば、-5.655という数の小数以下を四捨五入する場合、本来は「－６」にするべきですが、端数に５を足す方法では、「-5.655 + 0.5 = -5.155」の小数以下が切り捨てられて「－５」になります。また、-3.103では、「-3.103 + 0.5 = -2.603」となり、そのまま切り捨てると「－２」になってしまいます。

端数をランダムに発生させた数を四捨五入した場合、端数を「切り捨てる」確率と「切り上げる」確率は同じと思われがちですが、実は、若干数が増える傾向が現れます。端数によって、以下のように処理が行われているからです。

０…変わらず。端数はないので「切り捨て」も「切り上げ」も同じ数になる
１…切り捨て(減る)
２…切り捨て(減る)
３…切り捨て(減る)
４…切り捨て(減る)
５…切り上げ(増える)
６…切り上げ(増える)
７…切り上げ(増える)
８…切り上げ(増える)
９…切り上げ(増える)

端数が１桁しかない場合、上のように「変わらず」が１０％、切り捨てが４０％、切り上げが５０％となっており、切り上げが行われる確率が一番高くなっています。そのため、絶対値が増えていく傾向が現れるのです。無理数を含めた場合でも、端数以下がぴったり０になる場合もあるため、１０％とまではいかないまでも、「変わらない」確率が存在するため、若干「切り上げる」確率が高くなります。

最近接偶数への丸め

最近接偶数への丸めとは、端数が５より小さいなら切り捨て、５より大きい場合は切り上げを行うのは四捨五入と同じですが、５ちょうどの場合は、切り捨て・切り上げのうち、結果が偶数になる方を選択する、というものです。
たとえば、小数以下を最近接偶数への丸めで処理すると、以下のようになります。

５．５ → ６
６．５ → ６(四捨五入では ７)
－５．５ → －６
－６．５ → －６(四捨五入では－７)

最近接偶数への丸めでは、端数によって以下のように処理されます。

０…変わらず
１…切り捨て
２…切り捨て
３…切り捨て
４…切り捨て
５…切り捨て または 切り上げ(残る数が偶数なら切り捨て、残る数が奇数なら切り上げ)
６…切り上げ
７…切り上げ
８…切り上げ
９…切り上げ

四捨五入と「５」以外は同じです。ランダムに数を発生させた場合、偶数の確率と奇数の確率は同じなため、端数が「５」の場合の「切り捨て」と「切り上げ」の確率は同じです。つまり、「変わらず」は１０％、「切り捨て」は４５％、「切り上げ」は４５％になります。切り上がる確率が多かったために起こった四捨五入の問題点「若干数が増える傾向」が解消されています。JISでも四捨五入より望ましい、とされています。

端数処理の関数・メソッド

プログラミング言語の中には、端数処理の関数を提供しているものがあります。プログラミング言語以外でも、表計算ソフトやデータベース言語でもサポートされています。代表的なプログラミング言語の端数処理には、以下のものがあります。

・Ｃ／Ｃ＋＋

ceil 関数	もとの数以上の最小の整数へ、小数点以下を切り上げる(天井関数)
floor関数	もとの数以下の最大の整数へ、小数点以下を切り捨てる(床関数)

・Ｃ言語(C99)

nearbyint関数	あらかじめ指定された丸め方向に従って整数値に丸める
rint 関数	あらかじめ指定された丸め方向に従って整数値に丸める
round関数	四捨五入
trunc関数	絶対値がもとの数を超えないように、小数点以下を切り捨てる

・Ｊａｖａ

Math.ceil メソッド	もとの数以上の最小の整数へ、小数点以下を切り上げる(天井関数)
Math.floorメソッド	もとの数以下の最大の整数へ、小数点以下を切り捨てる(床関数)
Math.roundメソッド	四捨五入
Math.rint メソッド	最も近い整数値

・ＨＬＳＬ

ceil 関数	もとの数以上の最小の整数へ、小数点以下を切り上げる(天井関数)
floor関数	もとの数以下の最大の整数へ、小数点以下を切り捨てる(床関数)

floatやdoubleなどの実数型をintやlongなどの整数型へ型変換することによっても端数処理を行うことができます。ほとんどの場合は単純な切り捨てとなりますが、環境によっては四捨五入されることもあります。また、有効桁数や表現できる数値の範囲が異なるため、コンパイル時に警告が出ることがあります。

課　　題

最近接偶数への丸めを行うround関数、絶対値がもとの数を超えないように小数点以下を切り捨てる(切り落としを行う)trunc関数を作成しましょう。