

<u>ファイル名</u>	randsub.vb
<u>VBバージョン</u>	Visual Basic .Net のみ

## クラス名

Ran3

## 数学

Knuth の減算法により一様乱数を生成する。数学的（統計的）な性質はあまり明らかにされていない。.Net Framework の System.Random クラスも同減算法を用いてるが、ここに示すものもアルゴリズムは同一である。内部的に用いている定数は異なるので、得られる乱数列は同じにはならない。（MSEED の値を 161803398 にすれば同一の乱数列が得られるが、MBIG の値との関係で、あまり適切ではないと思われる。）

しかし、.Net Framework の 1.x と 2.0 で System.Random クラスの実装は大幅に異なっており、ここに示すものは両者の特徴を併せ持ったものである。内部的に生成される 31 bit 整数の乱数列に直接基づく一様乱数を得る関数・メソッド（NextRaw、NextDoubleRaw、NextBytesRaw）、この整数の乱数を 2 つ組み合わせ、52 bit 仮数部すべてを埋めた Double 型浮動小数点数の一様乱数を内部的に生成する Protected Overridable な関数（Sample）、この Double 型の浮動小数点数を基に、一様乱数を得る関数・メソッド（Next、NextDouble、NextBytes）を備える。

派生クラスを作り、Sample メソッドを Override して異なった分布の乱数を得られるようにすれば、これに基づく Next、NextDouble、NextBytes の関数・メソッドも連動して分布が変化する。

## キーワード

乱数、減算法、Knuth、.Net Framework、System.Random

<u>メソッド名</u>	New
<u>引数</u>	なし
<u>説明</u>	デフォルトのコンストラクタ。内蔵時計の時刻に依存した値で初期化する。

<u>メソッド名</u>	New
<u>引数</u>	ByVal seed As Integer
<u>説明</u>	初期値を指定できるコンストラクタ。

<u>メソッド名</u>	Initialize
<u>引数</u>	なし
<u>説明</u>	すでに作成されているインスタンスを、内蔵時計の時刻に依存した値で再初期化する。

<u>メソッド名</u>	Initialize
<u>引数</u>	ByVal seed As Integer
<u>説明</u>	すでに作成されているインスタンスを初期値を指定して再初期化する。

<u>関数名</u>	NextRaw
<u>戻り型</u>	Integer
<u>引数</u>	なし
<u>説明</u>	0 ～ Integer.MaxValue の範囲で整数の一樣乱数を生成して返す。内部的に最も基となる乱数である。下限値と上限値について数学的に精密な説明は得られていないが、本メソッドを用いた経験によれば、0 は発生するが、Integer.MaxValue は発生しない。Integer.MaxValue-1 は発生する。したがって、[0, Integer.MaxValue) を返る値の範囲と考えるべきである。このことは、.Net Framework 2.0 の System.Random クラスの Next メソッドの説明と一致している。

<u>関数名</u>	NextDoubleRaw
<u>戻り型</u>	Double
<u>引数</u>	なし
<u>説明</u>	[0.0, 1.0) の範囲で Double 型浮動小数点数の一樣乱数を生成して返す。NextRaw メソッドを 1 回呼び出しで生成される 31 bit 長の乱数を Double 型浮動小数点数の仮数部上位 bit に格納するだけなので、得られる 2 つの値が最も接近した場合でも、その差はせいぜい $1/\text{Integer.MaxValue}$ (約 $4.7 \times 10^{-10}$ ) であり、これ以上細かくなならない。

Double 型浮動小数点数の仮数部をすべて用いる乱数を得たければ、後述の NextDouble メソッドを用いること。

<u>メソッド名</u>	NextBytesRaw
<u>引数</u>	ByVal b() As Byte
<u>説明</u>	引数に与えた Byte 型配列のすべての要素に [0, 255] の範囲で整数の一樣乱数を格納する。各々の要素にのために、それぞれ NextRaw メソッドを呼び出して 31 bit 長の乱数を生成し上位 8 bit を切り出す。残りの 23 bit は捨てられる。.Net Framework 2.0 の System.Random クラスの NextBytes メソッドでは、下位 8 bit を用いるようであるが、一般的に上位 bit を用いることが望ましいと思われる。

引数の配列 b() が Nothing の場合には、何もせずに戻る。

<u>関数名</u>	NextDouble
<u>戻り型</u>	Double
<u>引数</u>	なし
<u>説明</u>	[0.0, 1.0) の範囲で Double 型浮動小数点数の一樣乱数を生成して返す。 NextRaw メソッドを 2 回呼び出し、それぞれ、生成される 31 bit 長の乱数の上位 26 bit を取り出し、Double 型浮動小数点数の仮数部 52 bit に格納する。この操作は、後述の Protected Overridable な Sample 関数に実装されている。

このため、本クラスを基に派生クラスを作る場合には、Sample 関数を Override することにより本関数で得られる関数の分布を変化させることができる。

<u>関数名</u>	Next
<u>戻り型</u>	Integer
<u>引数</u>	なし
<u>説明</u>	[0, Integer.MaxValue) の範囲で整数の一樣乱数を生成して返す。 Sample 関数を 1 回呼び出し、その結果に Integer.MaxValue を乗じ、小数部を切り捨てた結果である。

したがって、本クラスを基に派生クラスを作る場合には、Sample 関数を Override することにより本関数で得られる関数の分布を変化させることができる。本関数で返される値を [0, Integer.MaxValue) の範囲にしたければ、Override した Sample 関数も、分布は異なってもよいが、[0.0, 1.0) の範囲で値を返さなければならない。

<u>関数名</u>	Next
<u>戻り型</u>	Integer
<u>引数</u>	Byval maxValue As Integer
<u>説明</u>	[0,maxValue) の範囲（ただし maxValue >= 0）で整数の一樣乱数を生成して返す。Sample 関数を 1 回呼び出し、その結果に maxValue を乗じ、小数部を切り捨てた結果である。

したがって、本クラスを基に派生クラスを作る場合には、Sample 関数を Override することにより本関数で得られる関数の分布を変化させることができる。本関数で返される値を [0,maxValue) の範囲にしたければ、Override した Sample 関数も、分布は異なってもよいが、[0.0, 1.0) の範囲で値を返さなければならない。

maxValue が負の場合 System.ArgumentOutOfRangeException が発生する。

<u>関数名</u>	Next
<u>戻り型</u>	Integer
<u>引数</u>	ByVal minValue As Integer, ByVal maxValue As Integer
<u>説明</u>	<p>[minValue,maxValue) の範囲（ただし minValue ≤ maxValue。この条件を満たせば、minValue や maxValue は負でもよい。）で整数の一樣乱数を生成して返す。Sample 関数を 1 回呼び出し、その結果に maxValue と minValue の差（Double 型で求められる）を乗じ、小数部を切り捨て、改めて minValue を加えた結果である。</p> <p>したがって、本クラスを基に派生クラスを作る場合には、Sample 関数を Override することにより本関数で得られる関数の分布を変化させることができる。本関数で返される値を [0, Integer.MaxValue) の範囲にしたければ、Override した Sample 関数も、分布は異なってもよいが、[0.0, 1.0) の範囲で値を返さなければならない。</p> <p>maxValue が minValue より小さい場合、System.ArgumentOutOfRangeException が発生する。</p>
<u>メソッド名</u>	NextBytes
<u>引数</u>	ByVal b() As Byte
<u>説明</u>	<p>引数に与えた Byte 型配列のすべての要素に [0, 255] の範囲で整数の一樣らんすうを格納する。各々の要素にのために、それぞれ Next（引数なし）メソッドを呼び出して 31 bit 長の乱数を生成し上位 8 bit を切り出す。残りの 23 bit は捨てられる。</p> <p>間接的に Sample メソッドから返される値に依存しているので、本クラスを基に派生クラスを作る場合には、Sample 関数を Override することにより本関数で得られる関数の分布を変化させることができる。本関数で返される要素の値を [0, 255] の範囲にしたければ、Override した Sample 関数も、分布は異なってもよいが、[0.0, 1.0) の範囲で値を返さなければならない。</p> <p>引数の配列 b() が Nothing の場合には、何もせずに戻る。</p>
<u>関数名</u>	Sample
<u>保護等</u>	Protected Overridable
<u>戻り型</u>	Double
<u>引数</u>	なし
<u>説明</u>	<p>[0.0, 1.0) の範囲で Double 型浮動小数点数の一樣乱数を生成して返す。NextRaw メソッドを 2 回呼び出し、それぞれ、生成される 31 bit 長の乱数の上位 26 bit を取り出し、Double 型浮動小数点数の仮数部 52 bit に格納する。</p> <p>派生クラスを作り本関数を Override すれば、上述の Next、NextDouble、NextBytes の各関数・メソッドで得られる乱数の分布を変更することができる。しかし、Override しても、返す値の範囲は [0.0, 1.0) でなければならない。</p>

プロパティ名  
戻り型  
説明

Seed

Integer

初期化に用いた初期値を返す。この値を用いて再度初期化すれば、改めて同じ乱数系列を生成することができる。