NAME

Vector

SYNOPSIS

use Vector;

use Vector qw(:all);

DESCRIPTION

Vector class provides the following methods:

new, AddValues, Copy, GetLength, GetMagnitude, GetNumOfNonZeroValues, GetPercentOfNonZeroValues, GetSize, GetValue, GetValues, GetX, GetXYZ, GetY, GetZ, IsVector, Normalize, SetValue, SetValuePrintFormat, SetX, SetXYZ, SetY, SetZ, StringifyVector, IsVector

The following functions are available:

IsVector, SetValuePrintFormat UnitXVector, UnitYVector, UnitZVector, UnitVector, ZeroVector

The following operators are overloaded:

```
"" 0+ bool
@{}
+ - * / %
x .
== != < <= > >=
neg
abs exp log sqrt cos sin
```

FUNCTIONS

new

```
$NewVector = new Vector();
$NewVector = new Vector(@Values);
$NewVector = new Vector(\@Values);
$NewVector = new Vector($AnotherVector);
```

Creates a new Vector object containing *Values* and returns NewVector object. In case no *Values* are specified, an empty Vector is created.

AddValues

```
$Vector->AddValues(@Values);
$Vector->AddValues(\@Values);
$Vector->AddValues($AnotherVector);
```

Adds values to *Vector* using an array, reference to an array or another vector and returns *Vector*.

Copy

```
$NewVector = $Vector->Copy();
```

Creates a copy of *Vector* and returns *NewVector*.

GetLength

```
$Length = $Vector->GetLength();
```

Returns *Lengh* of a 3D *Vector* corresponding to its dot product.

GetMagnitude

```
$Length = $Vector->GetMagnitude();
```

Returns *Lengh* of a 3D *Vector* corresponding to its dot product.

GetNumOfNonZeroValues

```
$Value = $Vector->GetNumOfNonZeroValues();
```

Returns number of non-zero values in *Vector*.

GetPercentOfNonZeroValues

```
$Value = $Vector->GetPercentOfNonZeroValues();
```

Returns percent of non-zero values in *Vector*.

GetSize

```
$Size = $Vector->GetSize();
```

Returns size of a *Vector* corresponding to number of its values.

GetValue

```
$Value = $Vector->GetValues($Index);
```

Returns vector `Value` specified using *Index* starting at 0.

GetValues

```
@Values = $Vector->GetValues();
$ValuesRef = $Vector->GetValues();
```

Returns an array or a reference to an array containing all *Vector* values.

GetX

```
$X = $Vector->GetX();
```

Returns X value of a 3D *Vector*

GetXYZ

```
@XYZValues = $Vector->GetXYZ();
$XYZValuesRef = $Vector->GetXYZ();
```

Returns XYZ values of a 3D *Vector* as an array or a reference to an array containing the values.

GetY

```
$Y = $Vector->GetY();
```

Returns Y value of a 3D *Vector*.

GetZ

```
$Z = $Vector->GetZ();
```

Returns Z value of a 3D *Vector*.

IsVector

```
$Status = Vector::IsVector($Object);
```

Returns 1 or 0 based on whether *Object* is a `Vector` object.

Normalize

```
$Vector->Normalize();
```

Normalizes a 3D *Vector* by dividing its values by the length and returns *Vector*.

SetValue

```
$Vector->SetValue($Index, $Value);
```

Sets a *Vector* value specified by *Index* to *Value* and returns *Vector*.

SetValuePrintFormat

```
$Vector->SetValuePrintFormat($ValuePrintFormat);
Vector::SetValuePrintFormat($ValuePrintFormat);
```

Sets format for printing vector values for a specified *Vector* or the whole class. Default format: *%g*.

SetX

```
$Vector->SetX($Value);
```

Sets X value of a 3D vector to *Value* and returns *Vector*.

SetXYZ

```
$Vector->SetXYZ(@Values);
$Vector->SetXYZ(\@Values);
$Vector->SetXYZ($AnotherVector);
```

Sets XYZ values of a 3D vector and returns *Vector*.

SetY

```
$Vector->SetY($Value);
```

Sets Y value of a 3D vector to *Value* and returns *Vector*.

SetZ

```
$Vector->SetZ($Value);
```

Sets Z value of a 3D vector to *Value* and returns *Vector*.

StringifyVector

```
$String = $Vector->StringifyVector();
```

Returns a string containing information about *Vector* object.

UnitVector

```
$UnitVector = UnitVector([$Size]);
$UnitVector = Vector::UnitVector([$Size]);
```

Returns a UnitVector of *Size*. Default size: *3*.

UnitXVector

```
$UnitXVector = UnitXVector();
```

Returns a 3D UnitXVector.

UnitYVector

```
$UnitYVector = UnitYVector();
```

Returns a 3D UnitYVector.

UnitZVector

```
$UnitZVector = UnitZVector();
```

Returns a 3D UnitZVector.

ZeroVector

```
$UnitVector = ZeroVector([$Size]);
$UnitVector = Vector::ZeroVector([$Size]);
```

Returns a ZeroVector of *Size*. Default size: *3*.

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

BitVector.pm

COPYRIGHT