

показателей, включающих в себя, в частности материальные и временные затраты на проектирование.

Поскольку принимаемые предположения могут не оправдаться, часто требуется повторное выполнение проектных процедур предыдущих этапов после выполнения проектных процедур последующих этапов.

Такие повторения обеспечивают последовательное приближение к оптимальным результатам и обуславливают итерационный характер проектирования. Следовательно, итерационность нужно относить к важным принципам проектирования сложных объектов. На практике обычно сочетают восходящее и нисходящее проектирование. Например, восходящее проектирование имеет место на всех тех иерархических уровнях, на которых используются унифицированные элементы. Очевидно, что унифицированные элементы, ориентированные на применение в ряде различных систем определенного класса, разрабатываются раньше, чем та или иная конкретная система этого класса. Достоинство нисходящего проектирования состоит в том, что оно позволяет разработчикам сосредоточиться на основных для данного проблемах и отложить принятие всех тех решений, которые не должны приниматься на данном этапе проектирования. Нисходящее проектирование требует с самого начала ставить и решать наиболее фундаментальные задачи, откладывая частные вопросы для последующего рассмотрения.

УДК 271

Рудакова В.О.

ОРГАНИЗАЦИЯ МАТЕМАТИЧЕСКИХ ВЫЧИСЛЕНИЙ СРЕДСТВАМИ C#

БНТУ, Минск

Научный руководитель Дробыш А.А.

Для организации математических вычислений в языке C# создан специальный статический класс Math, обладающий набором полей и методов для выполнения основных математических операций.

Полями класса Math являются: π – число π ; e – число e . Далее представлены основные методы класса Math.

Методы определения модуля и получения знака – Abs, Sign.

Для вычисления абсолютного значения (модуля) используется метод `Abs()` класс `Math`. Возвращаемым значением этого метода является абсолютное значение переданного числа. Для получения знака числа используется метод `Sign()` класса `Math`. Возвращает значение -1 для отрицательного числа, 0 для нуля, 1 для положительного числа.

В качестве аргумента этим методам передается число. целого типа со знаком `—` `sbyte`, `short`, `int`, `ulong` или вещественного типа – `float`, `double`, `decimal`.

Методы определения минимума и максимума – `Min`, `Max`.

Методы `Min()` и `Max()` возвращают соответственно минимальное или максимальное значение из двух чисел, переданных этим методам в качестве аргументов. Эти методы работают со всеми базовыми числовыми типами данных, но при условии, что типы двух передаваемых аргументов совпадают.

Методы округления.

Для округления вещественного числа по правилам арифметики используются методы:

`Round(double Число, int КоличествоРазрядов)`, `Round(decimal Число, int КоличествоРазрядов)` – округляет указанное число до указанного числа десятичных разрядов после запятой.

`Round(double Число)`, `Round(decimal Число)` – округляет указанное число до ближайшего целого по правилам арифметики.

`Round(double Число, int КоличествоРазрядов, MidpointRounding)`, `Round(decimal Число, int КоличествоРазрядов, MidpointRounding)` – округляет указанное число до указанного числа десятичных разрядов после запятой. Третий аргумент задает правила округления, если значение находится ровно посередине между двумя числами и может принимать значения:

`MidpointRounding.AwayFromZero` – до ближайшего числа в сторону большего по модулю значения;

`MidpointRounding.ToEven` – до ближайшего четного числа.

Методы тригонометрических функций

Для вычисления тригонометрических функций класс `Math` предусматривает ряд методов. Все методы оперируют значениями углов типа `double`, заданными в радианах и возвращают значение типа `double`. Если угол задан в градусах, его можно перевести в радианы

при помощи формулы: `double deg = 30; //угол в градусах double rad=deg * Math.PI/180; //угол в радианах; Sin(угол) – вычисление синуса угла; Cos(угол) – вычисление косинуса угла; Tan(угол) – вычисление тангенса угла; Asin(значение) – вычисление арксинуса значения из диапазона $[-1; 1]$, возвращаемое значение лежит в диапазоне $[-\pi/2; \pi/2]$; Acos(значение) – вычисление арккосинуса значения из диапазона $[-1; 1]$, возвращаемое значение лежит в диапазоне $[0; \pi]$ и др.`

Логарифмические функции

Класс `Math` предусматривает ряд методов для работы с экспонентой и логарифмами:

`Exp(double степень)` – возвращает значение числа e (`Math.E`) в указанной степени.

`Log(double число)` – возвращает натуральный логарифм указанного числа.

`Log10(double число)` – возвращает десятичный логарифм указанного числа.

`Log(double число, double основание)` – возвращает логарифм указанного числа по указанному основанию.

Возведение в степень и извлечение квадратного корня

Для возведения числа в степень предусмотрен метод `Pow(double, double)`, в качестве первого аргумента которого указывается число, возводимое в степень, а в качестве второго аргумента – показатель степени. Для извлечения квадратного корня из числа типа `double` можно также использовать метод `Sqrt(double)`.

УДК 621.762.4

Руйчева А.П.

РАЗВИТИЕ МАШИННОГО ОБУЧЕНИЯ

БНТУ, Минск

Научный руководитель Липень С.Г.

Публицисты всего мира обещают нам неизбежное пришествие искусственного интеллекта. Его можно сравнить с малышом, который постепенно растет и развивается. Удивительно, сколько ресурсов было вложено в развитие этой отрасли за последние