Position, Matrix, Container Data Model

Overview

This document describes the functional specification of the Position, Matrix, and Container data model. The data model is used to store and access data in memory following the described business rules.

Requirements

Fast data-structure for creating, storing and accessing in-memory data

Can handle large data sets (e.g. 1000 containers of 50 matrices, with 500 x,y position data points)

Easy for client applications to create and access data (using standard C# methods).

Data

Data Overview

The PMC data model is comprised of the following elements:

Point - the simplest element is a data point. This is a generic type describing the simplest shape in 1D, 2D or 3D. It may be of any supported numerical C# type (e.g. integer, double, decimal). This should be implemented as immutable.

Position – an indexed collection of points.

Matrix – an indexed collection of positions.

Container – an indexed collection of matrices.

Containers – an indexed collection of Containers.

Position Data Types

The data stored at a position is a list of points. Each list item represent a point in 1D, 2D or 3D.

For 1D data point each list item a single numerical value (x).

For 2D data point each list item is a numerical pair (x,y)

For 3D data point each list item is of a numerical triple (x,y,z)

Rules across collections

- All numerical data across all data in a Containers collection is the same C# numerical type
- All positions in a specific matrix will be of the same position type.
- All matrices in all containers have the same number of positions.
- Each container in a containers collection contains the same number of matrices and the type of each indexed matrix will be the same across containers.
- The number of data points at each position for XY may vary between positions and matrices
- The number of data points at each XYZ position will be the same for all positions on a matrix and across equivalent matrix indexes across containers.
- Different XYZ matrices may have a different number of data positions
- An XY position may contain 0 or more data points
- An XYZ position may contain 0 or more data points
- An X position may contain no points.

Examples

Example 1: A container collection contains 3 containers. All data points are **decimal**. Each matrix contains 100 positions. Each container contains 2 matrices with the first matrix in each container being XY data and the second matrix in each container being X data. Position 1 of the XY data contains 50 points. Position 2 of the XY data contains 200 points. The other XY positions are empty. Position 1 and 2 of the X data matrix contain a numerical value, the others do not

Example 2: A container collection contains 10 containers. All data points are **double**. Each container contains 10 matrices with the first 5 matrices being XY data and the remaining 5 being X data. All positions in all matrices contain values.

Tasks

Build a class-library that exposes a data model for creation of Containers

Stored data should be immutable and should support object/collection initializers.

Data templating of the class (i.e. generic type) should be supported, so that the point element may be of **double**, **int** or **decimal** type.

Provide factory methods to create a collection of containers.

XML-document all public types and members of the class-library in code.

Provide accessors for accessing data by index and via IEnumerable.

Verify the data added to a collection follows the rules (throw exceptions of a special type if it does not).

Create unit tests with high code coverage and testing specified upper data set sizes.

Build example console client application that demonstrates use of the class in code and outputs contents of container to console.