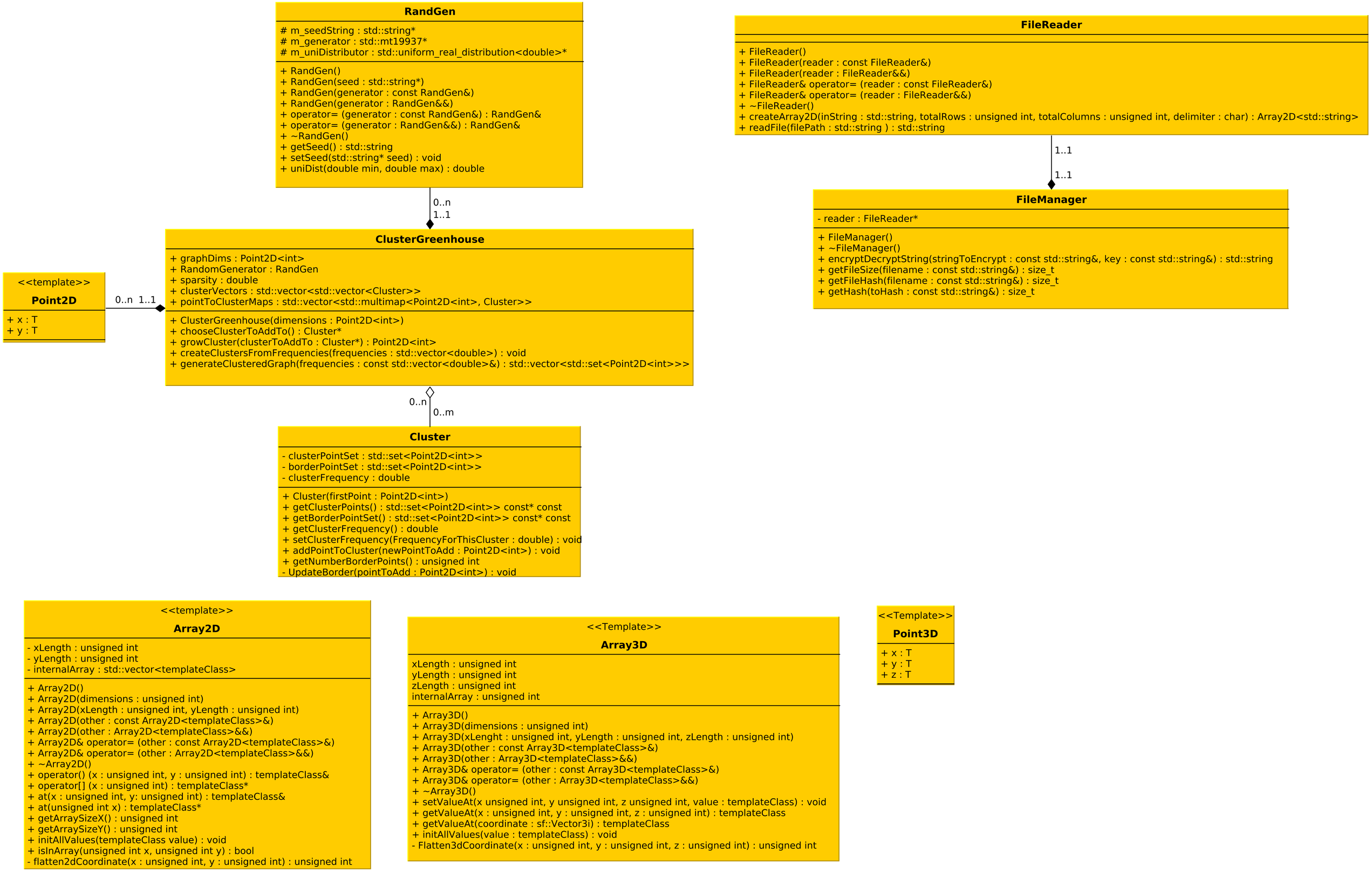
RandGen
m_seedString : std::string*
m_generator : std::mt19937*
m_uniDistributor : std::uniform_real_distribution<double>*
+ RandGen()
+ RandGen(seed : std::string*)
+ RandGen(generator : const RandGen&)
+ RandGen(generator : RandGen&&)
+ operator= (generator : const RandGen&) : RandGen&
+ operator= (generator : RandGen&&) : RandGen&
+ ~RandGen()
+ getSeed() : std::string
+ setSeed(std::string* seed) : void
+ uniDist(double min, double max) : double
0..n
1..1
ClusterGreenhouse
+ graphDims : Point2D<int>
+ RandomGenerator : RandGen
+ sparsity : double
+ clusterVectors : std::vector<std::vector<Cluster>>
+ pointToClusterMaps : std::vector<std::multimap<Point2D<int>, Cluster>>
+ ClusterGreenhouse(dimensions : Point2D<int>)
+ chooseClusterToAddTo() : Cluster*
+ growCluster(clusterToAddTo : Cluster*) : Point2D<int>
+ createClustersFromFrequencies(frequencies : std::vector<double>) : void
+ generateClusteredGraph(frequencies : const std::vector<double>&) : std::vector<std::set<Point2D<int>>>
<<template>>
Point2D
+ x : T
+ y : T
0..n 1..1
0..n
0..m
Cluster
- clusterPointSet : std::set<Point2D<int>>
- borderPointSet : std::set<Point2D<int>>
- clusterFrequency : double
+ Cluster(firstPoint : Point2D<int>)
+ getClusterPoints() : std::set<Point2D<int>> const* const
+ getBorderPointSet() : std::set<Point2D<int>> const* const
+ getClusterFrequency() : double
+ setClusterFrequency(FrequencyForThisCluster : double) : void
+ addPointToCluster(newPointToAdd : Point2D<int>) : void
+ getNumberBorderPoints() : unsigned int
- UpdateBorder(pointToAdd : Point2D<int>) : void
FileReader
+ FileReader()
+ FileReader(reader : const FileReader&)
+ FileReader(reader : FileReader&&)
+ FileReader& operator= (reader : const FileReader&)
+ FileReader& operator= (reader : FileReader&&)
+ ~FileReader()
+ createArray2D(inString : std::string, totalRows : unsigned int, totalColumns : unsigned int, delimiter : char) : Array2D<std::string>
+ readFile(filePath : std::string) : std::string
1..1
1..1
FileManager
- reader : FileReader*
+ FileManager()
+ ~FileManager()
+ encryptDecryptString(stringToEncrypt : const std::string&, key : const std::string&) : std::string
+ getFileSize(filename : const std::string&) : size_t
+ getFileHash(filename : const std::string&) : size_t
+ getHash(toHash : const std::string&) : size_t
<<template>>
Array2D
- xLength : unsigned int
- yLength : unsigned int
- internalArray : std::vector<templateClass>
+ Array2D()
+ Array2D(dimensions : unsigned int)
+ Array2D(xLength : unsigned int, yLength : unsigned int)
+ Array2D(other : const Array2D<templateClass>&)
+ Array2D(other : Array2D<templateClass>&&)
+ Array2D& operator= (other : const Array2D<templateClass>&)
+ Array2D& operator= (other : Array2D<templateClass>&&)
+ ~Array2D()
+ operator() (x : unsigned int, y : unsigned int) : templateClass&
+ operator[] (x : unsigned int) : templateClass*
+ at(x : unsigned int, y: unsigned int) : templateClass&
+ at(unsigned int x) : templateClass*
+ getArraySizeX() : unsigned int
+ getArraySizeY() : unsigned int
+ initAllValues(templateClass value) : void
+ isInArray(unsigned int x, unsigned int y) : bool
- flatten2dCoordinate(x : unsigned int, y : unsigned int) : unsigned int
<<Template>>
Array3D
xLength : unsigned int
yLength : unsigned int
zLength : unsigned int
internalArray : unsigned int
+ Array3D()
+ Array3D(dimensions : unsigned int)
+ Array3D(xLenght : unsigned int, yLength : unsigned int, zLength : unsigned int)
+ Array3D(other : const Array3D<templateClass>&)
+ Array3D(other : Array3D<templateClass>&&)
+ Array3D& operator= (other : const Array3D<templateClass>&)
+ Array3D& operator= (other : Array3D<templateClass>&&)
+ ~Array3D()
+ setValueAt(x unsigned int, y unsigned int, z unsigned int, value : templateClass) : void
+ getValueAt(x : unsigned int, y : unsigned int, z : unsigned int) : templateClass
+ getValueAt(coordinate : sf::Vector3i) : templateClass
+ initAllValues(value : templateClass) : void
- Flatten3dCoordinate(x : unsigned int, y : unsigned int, z : unsigned int) : unsigned int
<<Template>>
Point3D
+ x : T
+ y : T
+ z : T