**Composition Context**

Contains:

* Time (in seconds – float)
* Variable Map
* Pattern
* ResultSet

**DBCompositionBuilder**

Contains:

* Composer
* DBConnection
* PatternLibrary (Map of musical patterns)
* Composition Context (To be sent to all auxiliary functions)
* Control Pattern Heap
* ResultSet Stack

Can:

* Insert pattern from context into composer
* Save the context pattern into the control pattern heap with given time offset
* Add all control patterns in the heap into the composer
* Get a clone pattern from a musical library and override current context pattern
* Perform SQL Query and add result to stack (Context always contains the peeked resultset)
* Execute a given auxiliary function that changes its context

**Auxiliary Functions (Can also be considered as sub builders)**

* May contain data members that help it execute the command

(For example, PatternTransformer)

* Contains parameter fields
* Parameters are sent via constructor
* Contain just one method:

void execute(CompositionContext context)

Can:

* Transform the pattern in the context
* Change parameters

**Parameters**

* Contain just one method:

T value(CompositionContext context)

Base Implementations:

* Constant value – ignores context
* ResultSet reference – gets value from ResultSet according to column name and attempts to convert it to target value
* Parameter reference – gets value from Parameter map according to parameter name and attempts to convert it to parameter value