Basic Java Model

The *Basic Java Model* pattern creates elements and a diagram that is used to model Java programming constructs. The model provides a visualization of the programming code structure and allows code to be generated automatically from the model. An Aggregation relationship is used to model a Class that is comprised of another Class and Generalization relationships define three Classes that all specialize another Class.

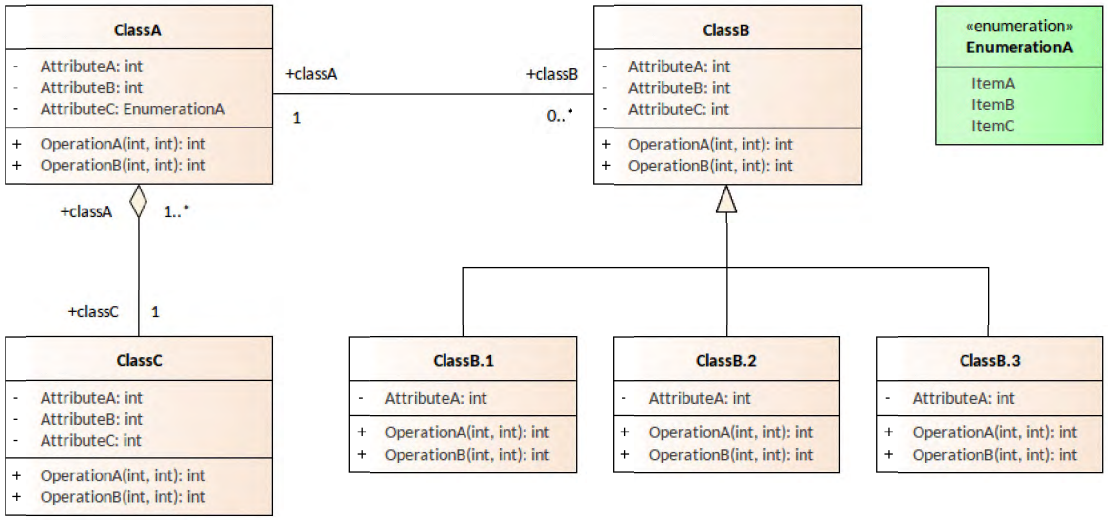

Figure 1. Shows a Class diagram used to define the important Classes and their relationships allowing the structure of programming code to be visualized.

Discussion

The purpose of the pattern is to provide a starting point for modeling programming code. It provides a powerful way of visualizing the structure of programs and allows elements in the diagram to be traced to other elements in the model including Use Cases, User Stories, Requirements, Business Rules, and Components. It can be used to generate the programming code files which can then be opened and completed using the built-in code editor.

It is typically created in the design and implementation phase of an initiative although it is also commonly viewed during the testing or support phases.

The following is a list of some things you may want to do when working with this pattern.

- Change the name of the Classes to suit the initiative.
- Change the names of the Attributes and Operations and associated Parameters.
- Change the name of the Enumeration to suit the initiative.
- Create additional Classes including Attributes and Operations, Enumerations and define relationships between the Classes.

The following is a list of some of the next steps available when applying the pattern.

- Generate programming code using the built-in Code Generation engine.
- Manage the Code inside Enterprise Architect using the built-in Code Editors.
- Manage the format of the generated code using the Code Template Framework.
- Document the model using the automatic Document Generator.
- Enter into discussions with colleagues and customers using the Discussion facility.
- Participate in a review using the built-in Review window

Reference

The following help topics will assist you learn about how to work with this pattern.

Generate Source Code

Local Paths

C# Options

Code Template Framework

Code Engineering Options

Tools you may find useful

The following are some of the tools that will be helpful when working with this pattern.

Source Code Editor

The Source Code Editor is a fully featured programming source code editor. It has a structure tree for easy navigation of attributes, properties and methods. Line numbers can be displayed and syntax highlight options can be configured. Many of the features that software engineers are familiar with in their favorite IDE, such as Intelli-sense and code completion are included in the editor. Viewing the source code juxtaposed with the Models from which it is generated brings a great clarity to the design effort and its implementation. For more details see the Editing Source Code help topic.

Visual Execution Analyzer

The Visual Execution Analyzer is made up of an advanced and powerful suite of tools that allow you to build, debug, record, profile, simulate and otherwise construct and verify your software development while keeping the code tightly integrated with your model. Enterprise Architect has rich support for a wide range of popular compilers and platforms, in particular the Java, .Net and Microsoft Windows C++ environments. Software development becomes a highly streamlined visual experience, quite unlike working in traditional environments. For more details see the Visual Execution Analyzer help topic.

Model Transformation

The Model Transformation facility allows a modeler to transform a Conceptual data model to a Logical data model and in turn a Logical Data Model to a Physical Data Model. The transformations are driven by user-defined or built-in templates. This facility will save time and effort and reduce the possibility of errors. For more details see the Model Transformation help topic.

Hand Drawn and Whiteboard Diagrams

The Hand Drawn and Whiteboard Mode are display options available for any diagram that changes a system-drawn diagram to appear as though it was drawn by hand and, optionally, hand drawn on a whiteboard. It is a powerful device to engage an audience by presenting the diagram in a rough and more immediate style giving the impression that it is just a sketch that can be changed. For more details see the Hand Drawn and Whiteboard Mode help topic.

Alternate and Images for Diagram Elements

Most standard elements allow an alternate image to be defined for an element that will be used in place of the graphical notation for the element either on a selected diagram

or as a default on all diagrams. For more details see the Using the Image Manager help topic.

Pan and Zoom

The Pan and Zoom facility is one of the tools that can be used to navigate around a large diagram. Often the resolution of a diagram must be reduced to ensure it is wholly visible but by using the Pan and Zoom window you can leave the diagram at a readable resolution and pan around to areas of interest zooming in when necessary. For more details see the Pan and Zoom help topic.

Document Generator

The Document Generator is a powerful facility in Enterprise Architect that allows a Database Engineer or other stakeholder to create high quality corporate or technical documentation directly from the model, suitable for internal or external audiences. For more details see the Documentation help topic or the more general topic on Model Publishing.