[TOPCODER]
SOFTWARE

Stub Class Generator 1.0 Requirements Specification

1. Scope

1.1 Overview

The Stub Class Generator component is able to generate class stubs for Java and C#, from the classes represented in UML Model.

1.2 Logic Requirements

1.2.1 CodeGenerator

This class acts as a facade of the component. It provides methods to generate stub code for a classifier / a list of classifiers / a package / a list of packages in a given language and to a specified location.

1.2.2 Language support

Java and C# should be supported.

The C# elements that do not have a representation in the UML Model will be represented as the closest element type, with a proper stereotype. The name of the stereotypes should be configurable.

For instance, the C# property will be an Operation with the <<property> stereotype.

1.3 Required Algorithms

The templates used or the stub generation algorithm.

1.4 Example of the Software Usage

The component will be used in the TopCoder UML Tool to generate class stubs, according to the project's language.

1.5 Future Component Direction

None.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None.

2.1.2 External Interfaces

The design must follow the interface found in the class diagram with the component interfaces. The designer is encouraged to add to the existing interface, but not to remove anything.

2.1.3 Environment Requirements

- Development language: Java 1.5
- Compile target: Java 1.5

2.1.4 Package Structure

com.topcoder.uml.stubclassgenerator

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

- The names of the non-standard element's stereotypes.

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

None

3.2.2 TopCoder Software Component Dependencies:

- UML Model components
- Configuration Manager 2.1.5 - recommended

**Please review the TopCoder Software component catalog for existing components that can be used in the design.

3.2.3 Third Party Component, Library, or Product Dependencies:

None

3.2.4 QA Environment:

- Solaris 7
- RedHat Linux 7.1
- Windows 2000
- Windows 2003

3.3 Design Constraints

The component design and development solutions must adhere to the guidelines as outlined in the TopCoder Software Component Guidelines. Modifications to these guidelines for this component should be detailed below.

3.4 Required Documentation

3.4.1 Design Documentation

- Use-Case Diagram
- Class Diagram
- Sequence Diagram
- Component Specification

3.4.2 Help / User Documentation

- Design documents must clearly define intended component usage in the 'Documentation' tab of Poseidon.

