Software Construction
CRC Cards for Control Flow Graph Generator
06/22/2016
Team 7

Figure 1: Control Flow Graph CRC Card

Class Name: Control Flow Graph	
Superclass:	
Subclasses:	
Description: It is in charge of building a control flow graph by the use of the parsed meaning of each statement from a given file. Building a CFG is a composition of the following components: node, edge, points, and parsed statement instruction.	
Responsibilities: Reads parsed statements. Build graph. Create nodes (blocks). Create edges (directed connections). Set points for each node.	**Collaborations:** Node Edge
Comments: Class that is the intermediary between parsed statements from the file and the XML file with its tags. This class will manage the parsed data (conditions, iteration, declaration in the code). It is the instantiated class that holds proper human words to determine the nodes and edges	

Figure 2: GUI CRC Card

Class Name: GUI	
Superclass:	
Subclasses:	
Description: Class that displays CFG with graphical elements.	
Responsibilities: Displays GUI content for CFG.	**Collaborations:**
Comments: Class in charge of visualizing the created CFG.	

Figure 3: Edge CRC Card

Class Name: Edge	
Superclass:	
Subclasses:	
Description: A directed path from one node to another.	
Responsibilities: Identify statement flow for nodes. Label statement flow for nodes.	**Collaborations:**
Comments: Entry point, and exit point by each node.	

Figure 4: Node CRC Card

Class Name: Node	
Superclass:	
Subclasses:	
Description: It is the representation of a basic block.	
Responsibilities: Know statement. Create nodes based on statement type: iterative, conditional, sequential.	**Collaborations:** Statement
Comments: Class that contains corresponding lines reflecting the sentence, node identifier	

Figure 5: Statement CRC Card

Class Name: Statement
Superclass:
Subclasses: Sequential, Conditional, Iterative

Description: Syntax that defines the statements.	
Responsibilities: Get statement. Identify terminals. Identify non-terminals.	**Collaborations:**
Comments: Parent class that determines type of statements that is being applied. A statement is composed by terminal, non terminal. Determine statement type is based on the terminal,non-terminal	

Figure 6: Conditional CRC Card

Class Name: Conditional	
Superclass: Statement	
Subclasses:	
Description: Syntax rules for conditional statements.	
Responsibilities: Identifies conditional statements. Send conditional block.	**Collaborations:** Grammar
Comments:	

Figure 7: Sequential CRC Card

Class Name: Sequential	
Superclass: Statement	
Subclasses:	
Description: Syntax rules for sequential statements.	
Responsibilities: Identifies sequential statements Send sequential block	**Collaborations:** Grammar

Comments:

Figure 8: Iterative CRC Card

Class Name: Iterative	
Superclass: Statement	
Subclasses:	
Description: Syntax rules for iterative statements.	
Responsibilities: Identify iterative statements. Send iterative block.	**Collaborations:** Grammar
Comments:	

Figure 9: XML CRC Card

Class Name: XML	
Superclass:	
Subclasses:	
Description: This class is in charge of providing an XML service with instructions represented with tags.	
Responsibilities: Generate XML file. Add XML tags that reflects the CFG structure. Transfer XML file.	**Collaborations:**
Comments: Part of the project specification. XML file with tags will be used to send instructions to the visualization library. The visualization library might be a web service as D3 Java Swing alike.	

Figure 10: Parser CRC Card

Class Name: Parser	
Superclass:	
Subclasses:	
Description: Uses a set of rules provided by a grammar to parse statements.	
Responsibilities: Retrieve statements from file. Call rules for each statements. Parse statements. Store parsed statements. Send parsed statements to the CFG.	**Collaborations:** Grammar
Comments:	

Figure 11: Grammar CRC Card

Class Name: Grammar	
Superclass:	
Subclasses:	
Description: A set of syntax rules for a given language.	
Responsibilities: Knows terminals. Knows nonterminals. Retrieve statements from file. Knows rules for each statements.	**Collaborations:**
Comments:	