





1. Features

- 1.1. Creation
 - 1.1.1. Class
 - 1.1.2. Abstract Class
 - 1.1.3. Interface
 - 1.1.4. Text Area
- 1.2. Modification
 - 1.2.1. Add/Remove Attribute
 - 1.2.2. Add/Remove Method
- 1.3. Connection
 - 1.3.1. Inheritance
 - 1.3.2. Association
- 1.4. Utility
 - 1.4.1. Debug Snapshot
 - 1.4.2. Generate Java Code

2. Creation

- 2.1. Class
 - 2.1.1. From the toolbar, select the “Class” button 
 - 2.1.2. Click anywhere in the drawing to place a class figure
 - 2.1.3. The class will have a default name of “Class” and will have no attributes or methods
- 2.2. Abstract Class
 - 2.2.1. From the toolbar, select the “Abstract Class” button 
 - 2.2.2. Click anywhere in the drawing to place an abstract class figure
 - 2.2.3. The abstract class will have a default name of “*Class*” (italicized to show abstractness) and will have no attributes or methods
- 2.3. Interface
 - 2.3.1. From the toolbar, select the “Interface” button 
 - 2.3.2. Click anywhere in the drawing to place an interface figure
 - 2.3.3. The interface will have a default name of “Class” and will have no methods
 - 2.3.4. An interface also not contain attributes
- 2.4. Text Area
 - 2.4.1. From the toolbar, select the “Text Area” button 
 - 2.4.2. Click anywhere in the drawing to place a text area figure
 - 2.4.3. A text area figure may be resized by selecting the figure, and then dragging the blue handles to your desired size
- 2.5. Any of these figures may be moved around the drawing by clicking and dragging the figure

3. Modification

3.1. Attributes

3.1.1. Addition

- 3.1.1.1. To add an attribute to a class or an abstract class, click the “Add Attribute” button on the toolbar **+A** and select the class figure that is to receive the attribute
- 3.1.1.2. Alternatively, you may right-click the class figure and select “Add Attribute” from the context menu
- 3.1.1.3. In the dialog box that appears, type in the attribute signature you wish to add
 - 3.1.1.3.1. Attribute signatures can be in UML specification, e.g. “+ height : double” or in Java format, e.g. “public double height”
 - 3.1.1.3.2. You may also wish to provide an initialization value, e.g. “public double height = 73.5”
 - 3.1.1.3.3. If you specify that an attribute is static, e.g. “public static int manyNodes”, it will be displayed with an underline

3.1.2. Removal

- 3.1.2.1. To remove an attribute, right-click the class that contains the attribute and select “Remove Attribute” from the context menu
- 3.1.2.2. In the dialog box, type either the full signature of the attribute you wish to remove, or just the name of the attribute

3.2. Methods

3.2.1. Addition

- 3.2.1.1. To add a method to a class, click the “Add Method” button on the toolbar **+M** and select the class figure that is to receive the method
- 3.2.1.2. Alternatively, you may right-click on the class figure and select “Add Method” from the context menu
- 3.2.1.3. In the dialog box that appears, type in the method signature you wish to add
 - 3.2.1.3.1. Method signatures can be in UML Specification, e.g. “+ print() : void” or in Java format, e.g. “public void print()”
 - 3.2.1.3.2. You may also provide parameters in UML specification or Java format, e.g. “+ print(s : int) : void” or “public void print(int s)”
 - 3.2.1.3.3. If you specify that a method is static, it will be displayed with an underline
- 3.2.1.4. Methods may be overloaded by adding multiple methods with the same signature but with different parameters

3.2.2. Removal

- 3.2.2.1. To remove a method, right-click the class that contains the method and select “Remove Method” from the context menu
- 3.2.2.2. In the dialog box, type either the full signature of the method you wish to remove or just the name of the method
- 3.2.2.3. If you type just the name of a method, any overloaded methods with that name will be removed

4. Connection

4.1. Inheritance


- 4.1.1. To display inheritance between two classes, you will need to draw an inheritance line from the subclass to its superclass
- 4.1.2. Click the “Inheritance” button on the toolbar and click the subclass
- 4.1.3. While holding down the left mouse button, draw a line to the middle of the superclass
- 4.1.4. Release the mouse to form a new inheritance connection
- 4.1.5. Alternatively, you can click the subclass, and then draw a line from the red handle on top to its superclass without having to click any toolbar buttons
- 4.1.6. An inheritance line will have an unfilled triangular arrow pointing at the superclass
- 4.1.7. Any inheritance line drawn to an interface will be displayed as a dotted line

4.2. Association


- 4.2.1. Associations can be drawn between two classes
- 4.2.2. Click the “Association” button on the toolbar and click the first class
- 4.2.3. While holding down the left mouse button, draw a line to the middle of the second class
- 4.2.4. Release the mouse to form a new association connection
- 4.2.5. Alternatively, you can click the first class, and then draw a line from the red handle in the middle to the second class
- 4.2.6. Line Decorations
 - 4.2.6.1. By default, an association has no decorations on the end, thus it is a bi-directional association
 - 4.2.6.2. You can change the end decorations on the line by right-clicking the association line, and selecting either “Set Start Decoration” or “Set End Decoration” from the context menu
 - 4.2.6.3. In the dialog that appears, select either Composition, Aggregation or Association and hit “Ok” to change the line’s decorations
- 4.2.7. You can also set an association line’s role name, as well as its multiplicity via the right-click context menu

5. Utility

5.1. Debug Snapshot

- 5.1.1. While creating your UML class diagram, you can check to see what the data representation of your classes is quickly and easily by clicking the “Debug Snapshot” button on the toolbar 
- 5.1.2. The Java data representation of the classes will be printed to the console window

5.2. Generate Java Code

- 5.2.1. To turn your UML class diagram in to runnable Java code, click the “Generate Java” button on the toolbar 
- 5.2.2. In the file chooser dialog that appears, select a location to save your java file and click “Generate”

5.3. Templates (Works in Linux and Mac)

5.3.1. Creation

5.3.1.1. To create a reusable template from your drawing, click “File” and select “Save Template”

5.3.1.2. Type a filename and hit “Ok” and your template will be saved in your default templates directory

5.3.2. To open a template for editing, click “File” and select “Templates” and choose the template you wish to open. Your template will be loaded to the window

5.4. Saving a file

5.4.1. To save your UML drawing, click “File” and select “Save” or “Save as”

5.4.2. Enter a name for your file in the file chooser window that appears and click “Save”

5.4.3. The program will output an XML file of your drawing to the location you choose

5.5. Opening a file

5.5.1. To open a previously saved file, click “File” and select “Open”

5.5.2. Select the XML file you wish to open and click “Open”

5.5.3. The editor will be filled with your previously saved drawing