

UNIVERSITY OF ABERYSTWYTH

FIRST YEAR GROUP PROJECT

UML2 Java

Authors:
Daniel Malý,
Samuel Sherar,
Lee Smith

UML2Java Graphical Designer

UML2Java is a Graphic User Interface program that allows a user to create UML2 class diagrams. The program will then output the class files from the created class diagram including all relevant fields, methods and relationships to other classes. The UML designer was created as a requirement for our first year group project.

Contents

1	File Menu	3
1.1	Create new Class Diagram	3
1.2	Open an existing diagram	3
1.3	Saving a diagram	3
1.4	Printing a Diagram	4
1.5	Exiting the Designer	4
2	Edit Menu	4
2.1	Undoing an Action	4
2.2	Redoing an Action	5
3	Editing Canvas Settings	5
3.1	Resize the Canvas	5
3.2	Fitting to Diagram	5
4	Exporting	6
4.1	Export to Code	6
4.2	Export Image of Diagram	6
5	Adding Elements to the Class Diagram	7
5.1	Adding a new class	7
5.1.1	Adding Data fields to the new Class	8
5.1.2	Adding Methods to the new Class	8
5.2	Adding a new Relationship between classes	9
6	Editing	10
6.1	Changing the Class Name	10
6.2	Moving a Class	10
6.3	Editing a Data Field	11
6.4	Editing a Method in a Class	11
6.5	Editing a Relationship	12
6.5.1	Editing the Relationship Type	12
6.5.2	Moving Relationship Points	12
6.5.3	Adding a Label to a Relationship	13
6.5.4	Adding Cardinalities to the Line	13

1 File Menu

1.1 Create new Class Diagram

To create a new Class Diagram, go to the top menu bar and press **File** → **New**

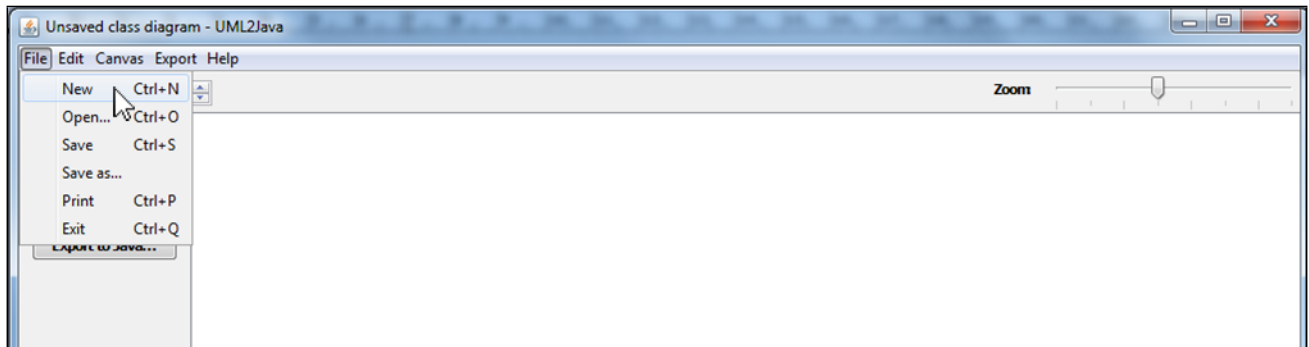
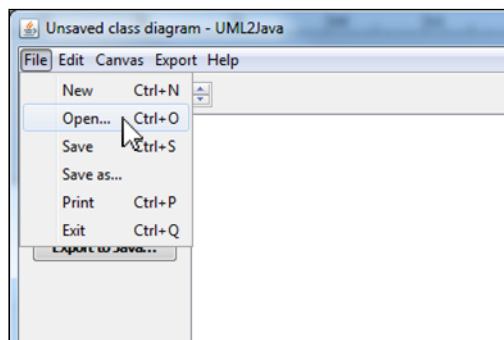


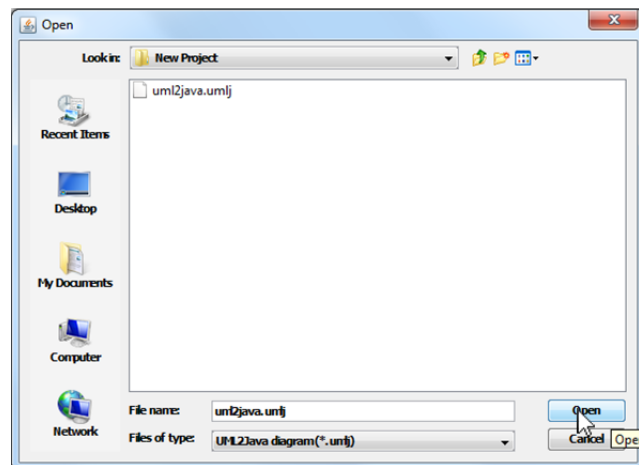
Figure 1: Creating a new Diagram

1.2 Open an existing diagram

To open a previously saved diagram, go to the top menu bar and press **File** → **Open...** or simply press **CTRL+O**. This will open up a new window asking for the location of the file you wish to open.



(a) File → Open

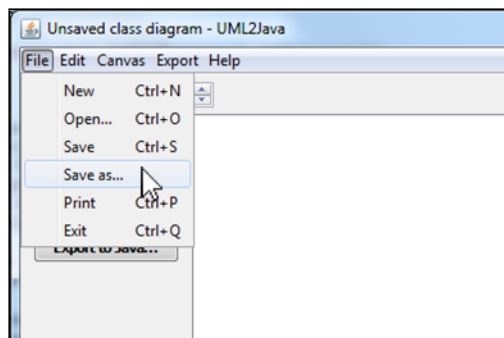


(b) Select File to open

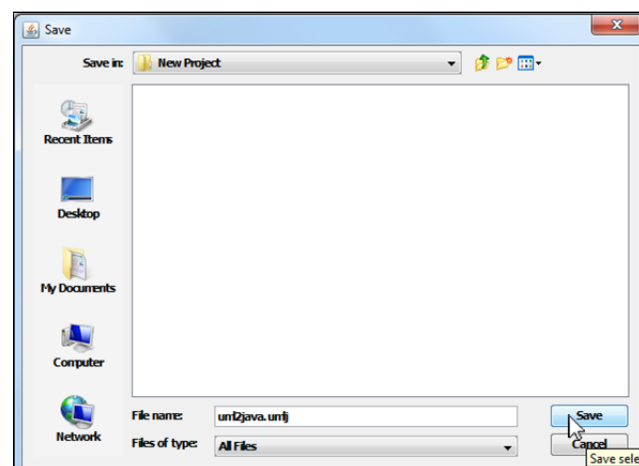
Figure 2: Opening an existing diagram

1.3 Saving a diagram

To save a diagram, go to the top menu bar and press **File** → **Save**/**Save as...** or simply press **CTRL+S**.



(a) File → Save



(b) Select Save Location

Figure 3: Saving a diagram

1.4 Printing a Diagram

To print a created diagram, go to the top bar and press **File** → **Print** or simply press **CTRL+P** and select the desired printer from the open window

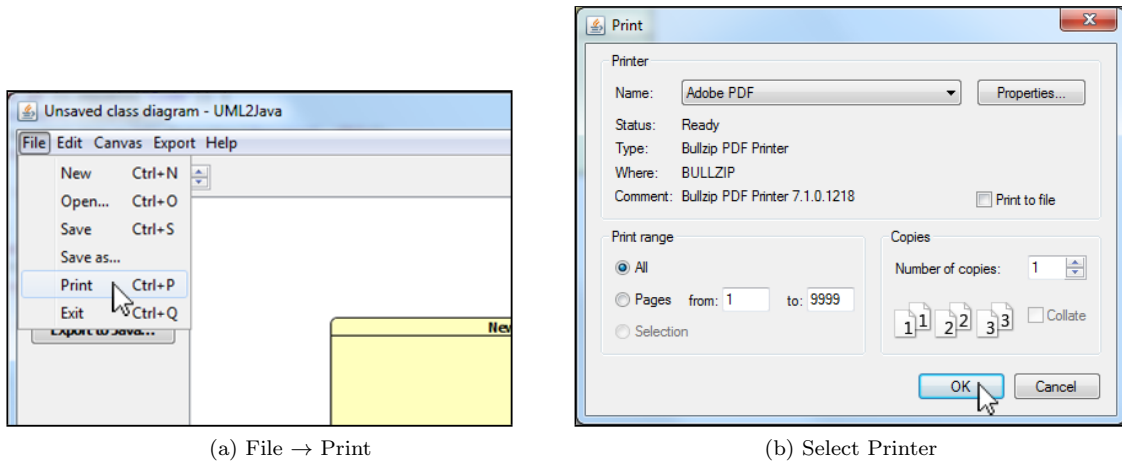


Figure 4: Printing a diagram

1.5 Exiting the Designer

To end the current session in UML2Java

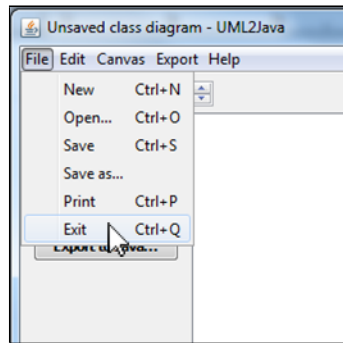


Figure 5: Exiting the Designer

2 Edit Menu

2.1 Undoing an Action

If you require to undo the previous action, go to the top bar and press **Edit** → **Undo** or press **CTRL+Z** from the keyboard

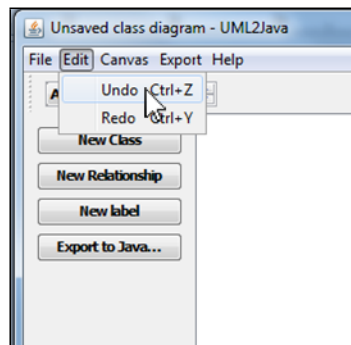


Figure 6: Undo last Action

2.2 Redoing an Action

if you need to redo a previously undone action, simply go to the top bar and press **Edit** → **Redo** or press **CTRL+Y** from your keyboard

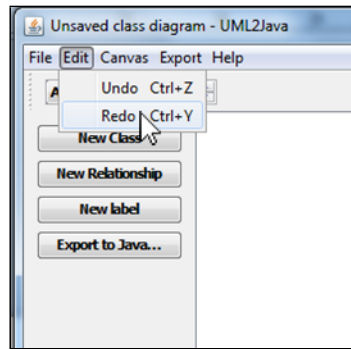
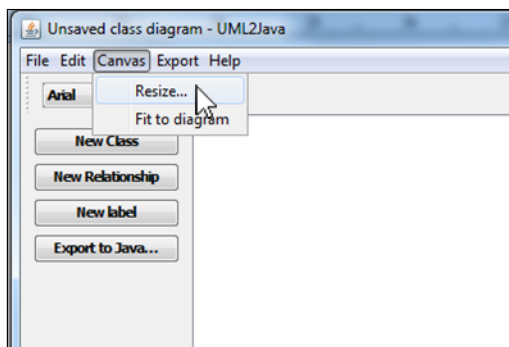


Figure 7: Redoing an Action

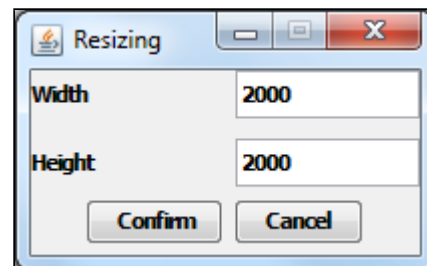
3 Editing Canvas Settings

3.1 Resize the Canvas

If you require a different canvas size , you can do this from the top menu, pressing **Canvas** → **Resize**



(a) Canvas → Resize...



(b) Select resize options

Figure 8: Printing a diagram

3.2 Fitting to Diagram

The canvas menu option also allows you to resize the canvas to a state where all of the current contents of the diagram will be able to be seen with little white space around the resulting design, to achieve this press **Canvas** → **Fit to Diagram**

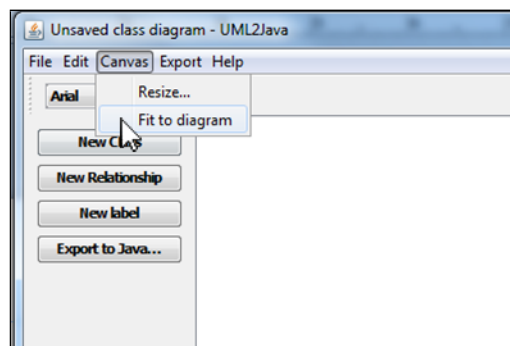
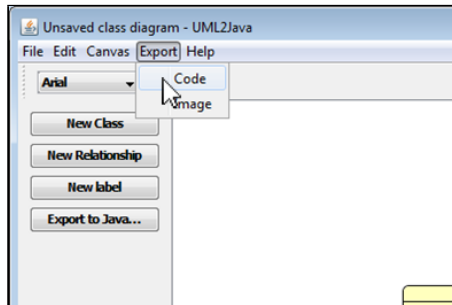


Figure 9: Fit diagram to visible canvas

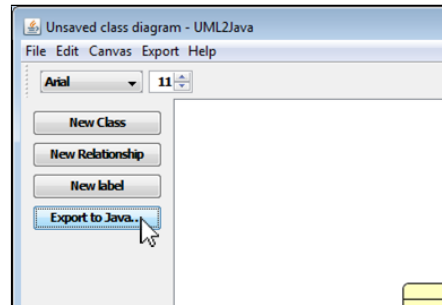
4 Exporting

4.1 Export to Code

To create the code files of your created diagram, go to the top bar and select **Export** → **Code** this will generate the .java files of your created diagram, it will take into account any relationships you have created between classes, any variable declarations you have made, and methods you have created. Or you can go to the left menu bar and select **Export to Java** from the list there

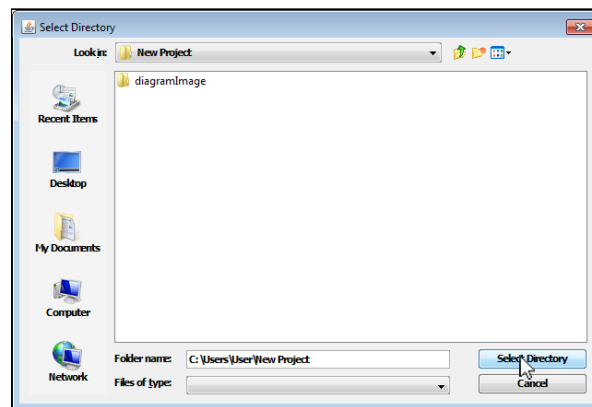


(a) Export → Code



(b) Export to Java...

You will then be asked to select a directory to save the code files to, you can select an existing directory or create a new one

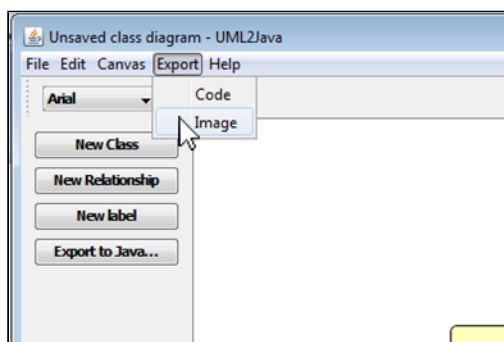


(c) Choose Directory to create files

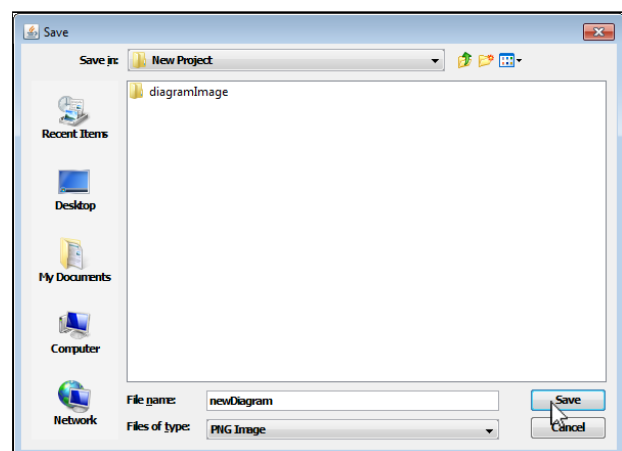
Figure 10: Exporting to Java code

4.2 Export Image of Diagram

To create an image file of the class diagram you have created, go to **Export** → **Image** and a save box will appear, where you can select the location of where you wish to save the file and also the file type, either .png or .jpg



(a) Export → Image



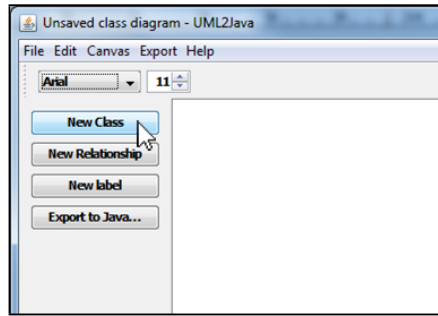
(b) Choose Directory to save image

Figure 11: Export Diagram as Image

5 Adding Elements to the Class Diagram

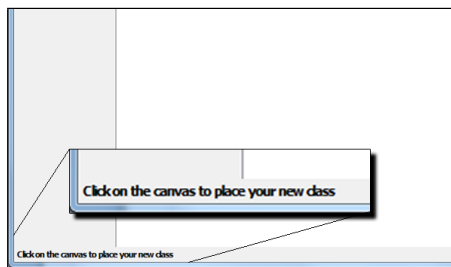
5.1 Adding a new class

To create a new class on the diagram, go to the left-hand menu and select **New Class**



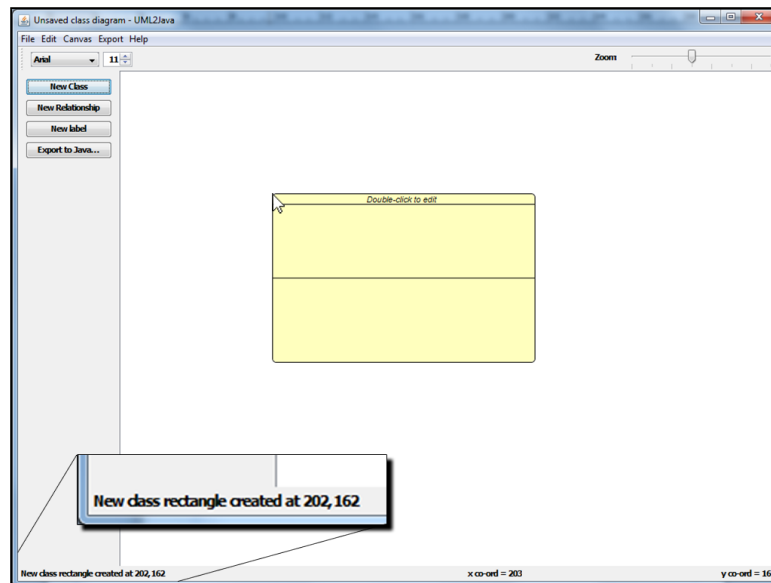
(a) Add new class

To make sure that you have selected to add a new class, once pressed, a message will appear in the status bar at the bottom of the window



(b) Click to create new class

To add the class to the diagram, left-click on the white canvas and your new class will appear



(c) Added Class

You will receive confirmation via the status bar at the bottom of the screen, giving the x and y co-ordinates of the generated class box

Figure 12: Adding a new class to diagram

5.1.1 Adding Data fields to the new Class

To add a new Data field to the Created Class, **Right Click** on the class, and select **Add data field**, the new data field will then be added to the diagram.

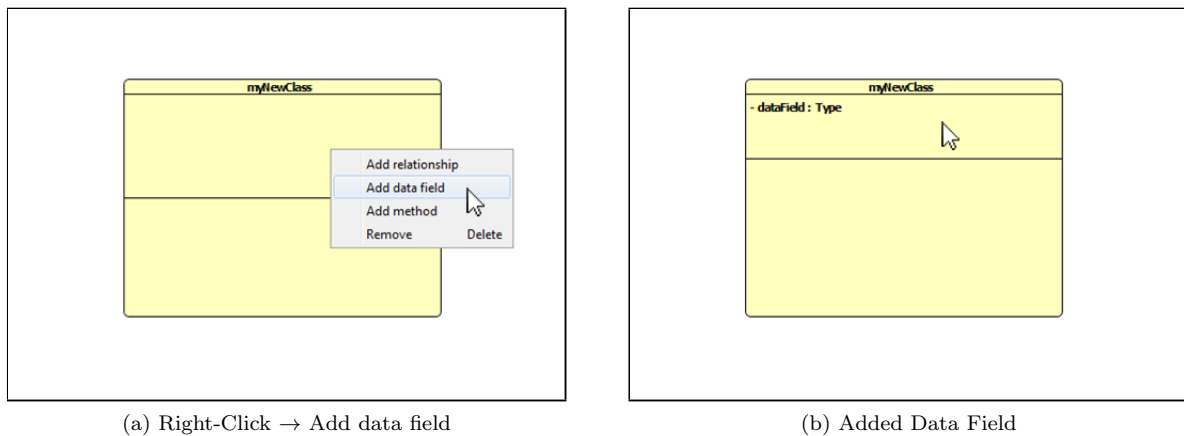


Figure 13: Adding a new Data Field to the Class

5.1.2 Adding Methods to the new Class

to add a new method to the class, **Right Click** on the class, and select **Add method**, the new method will then be added to the diagram

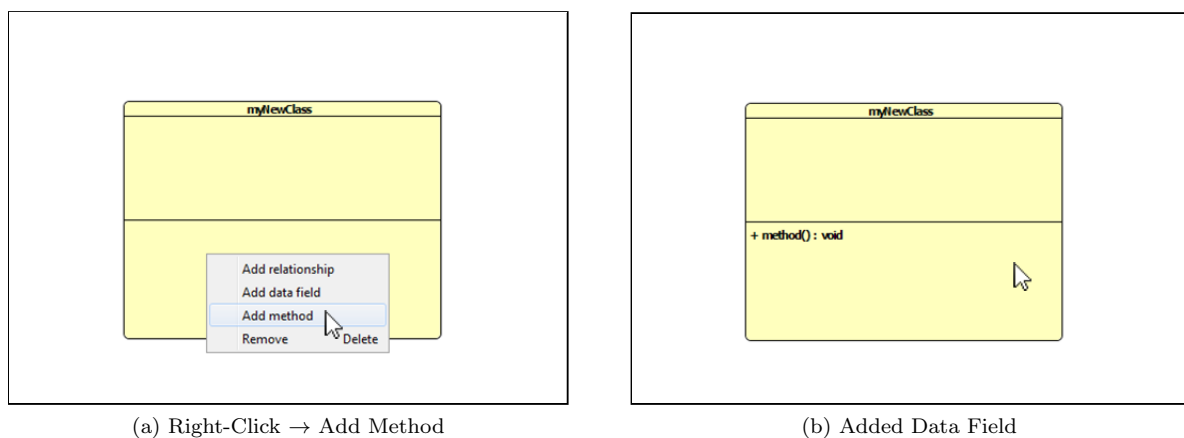
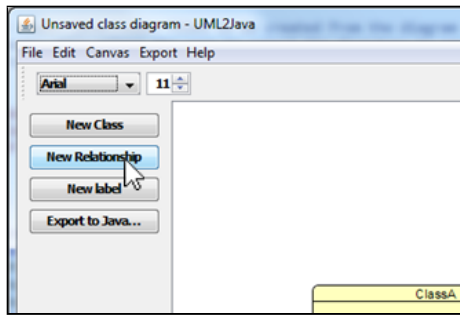


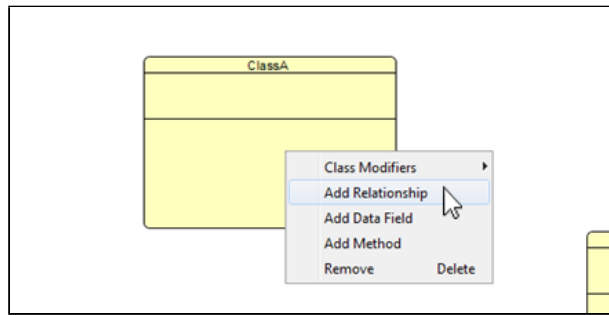
Figure 14: Adding a new Method to the Class

5.2 Adding a new Relationship between classes

To add a relationship between two classes, press **New Relationship** from the left-hand menu, or right click on a class to bring up a popup menu

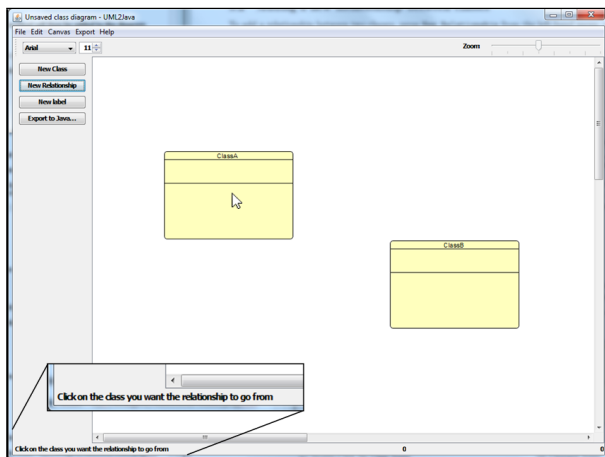


(a) Press New Relationship

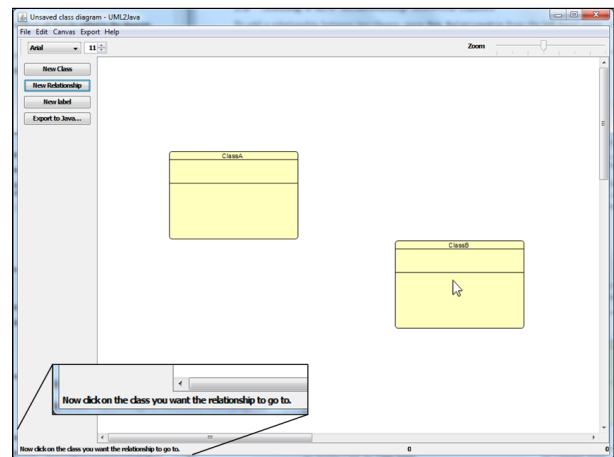


(b) Right-Click → Add Relationship

If you selected option a) then you will be required to press the initial class and the linking class. Option b) will automatically assume the class you used to **Add Relationship** is the initial class. The current state of what needs to be selected will be displayed in the status bar at the bottom of the window.

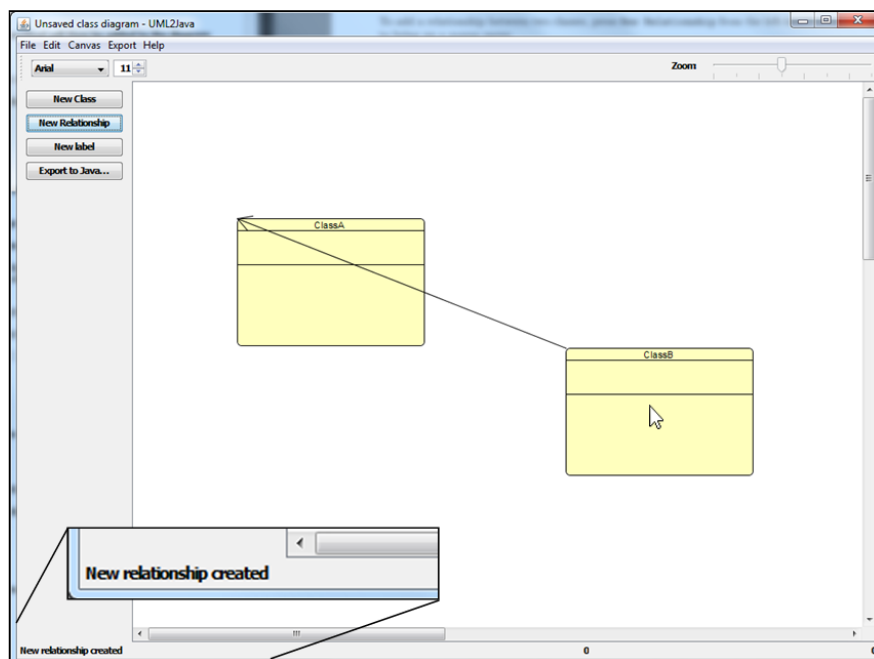


(c) 1st Class



(d) 2nd Class

An arrow will then be drawn linking the two classes



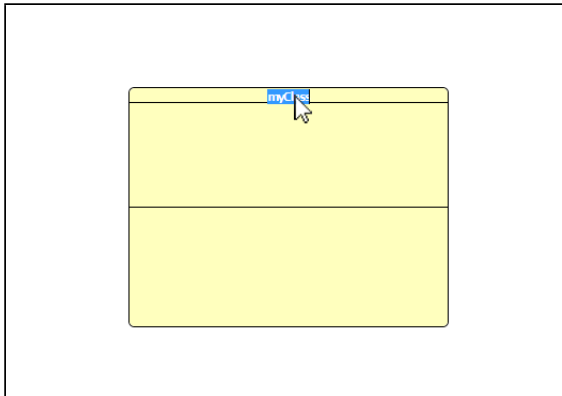
(e) Relationship Drawn

Figure 15: Adding a relationship between classes

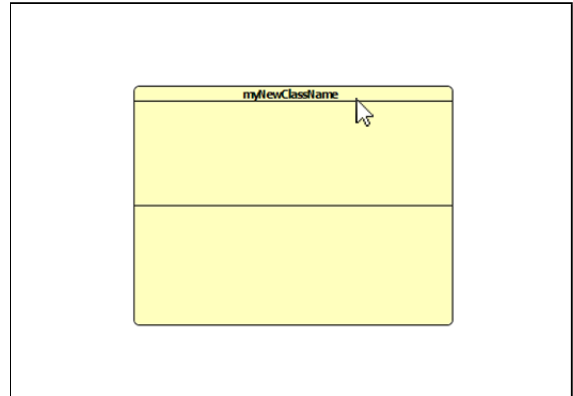
6 Editing

6.1 Changing the Class Name

To change the class name of a class added to the diagram (see *Adding new Class*) and Double-Click the class name, the text should highlight and you can then enter the new name, press return to confirm your new Class Name



(a) Double-Click the Class Name

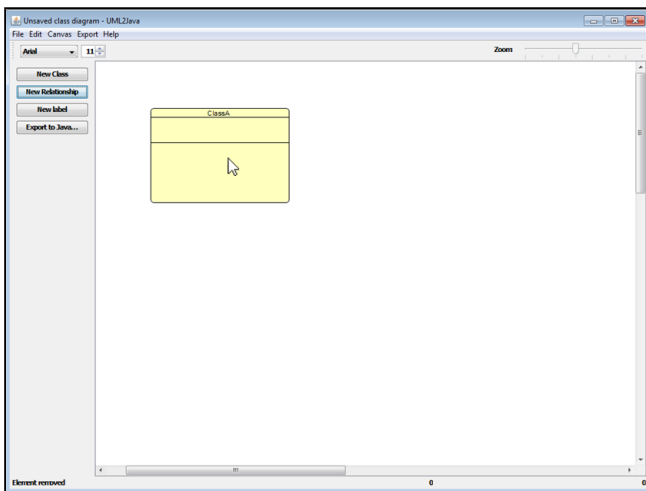


(b) Changed Name

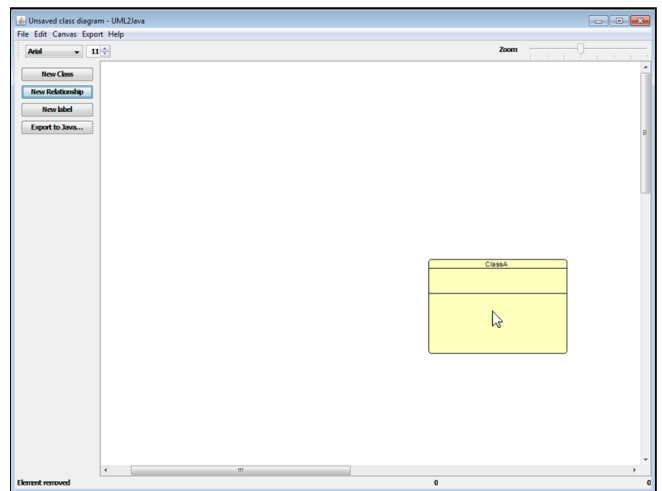
Figure 16: Editing the class name of a Class

6.2 Moving a Class

To move the classes simply click and hold onto a class and move it to the desired new location on the screen



(a) Click and Hold a class



(b) Release at desired location

Figure 17: Editing a data field in a class

6.3 Editing a Data Field

To change a data field in a class added to the diagram (*see Adding new Class & Adding data fields*) Double-Click the data field you wish to change, the text should highlight and you can then edit the data field, your edit **MUST** be in the format

(+,-,#) dataFieldName : DataType

or if initialised

(+,-,#) dataFieldName : DataType = initialisation

Press Return ↵ to confirm your new Class Name

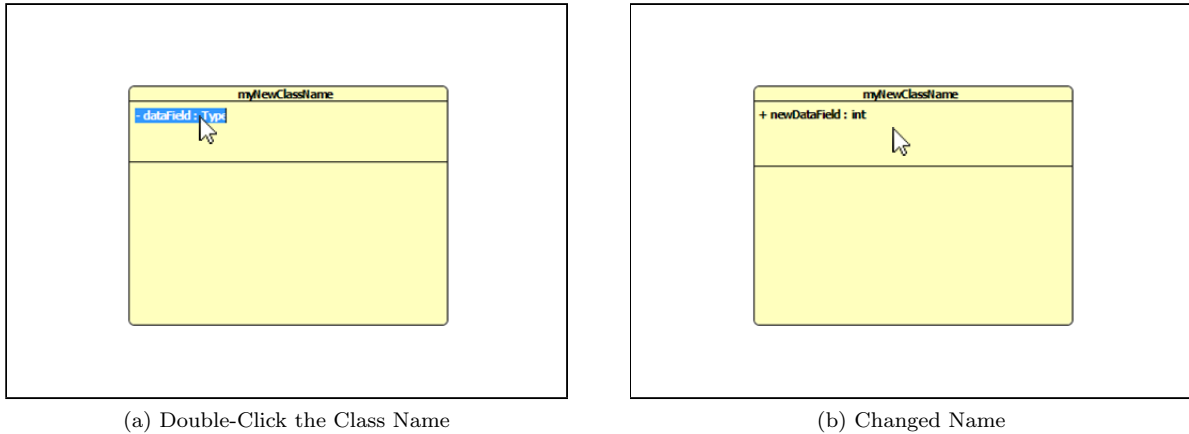


Figure 18: Editing a data field in a class

6.4 Editing a Method in a Class

To change a method in a class added to the diagram (*see Adding new Class & Adding Methods*) Double-Click the method you wish to change, the text should highlight and you can then edit the method definition, you edit **MUST** be in the format

(+,-,#) methodName() : Return Type

or if arguments are to be passed into the method

(+,-,#) methodName(arg1 : Arg1Type, arg2 : Arg2Type...) : ReturnType

Press Return ↵ to confirm your new Class Name

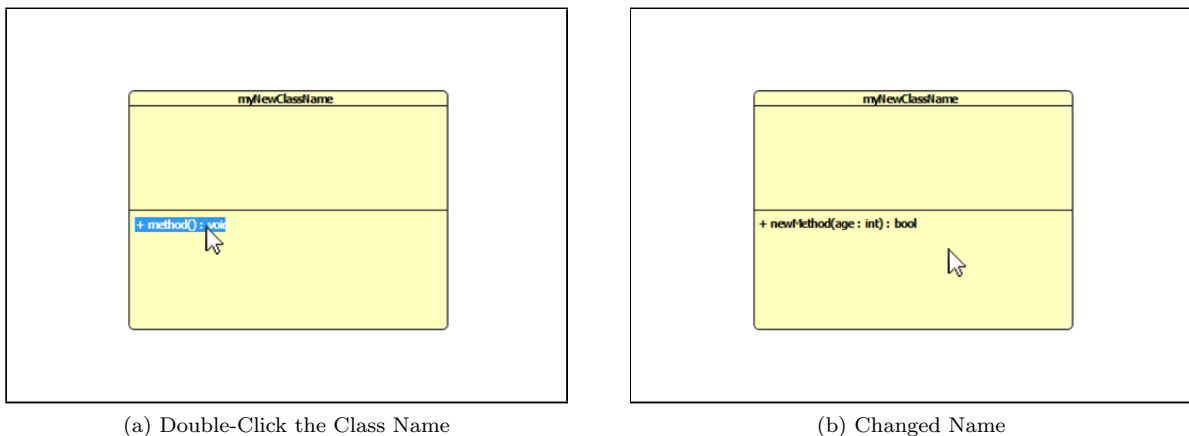


Figure 19: Editing a method in a class

6.5 Editing a Relationship

6.5.1 Editing the Relationship Type

To change the type of relationship between 2 classes, Right-Click on the relationship arrow and hover over **Change Relationship** and a sub-menu should appear. From this menu you can change the type of relationship, these include: Aggregation, Composition, Inheritance, Uses, and Implements

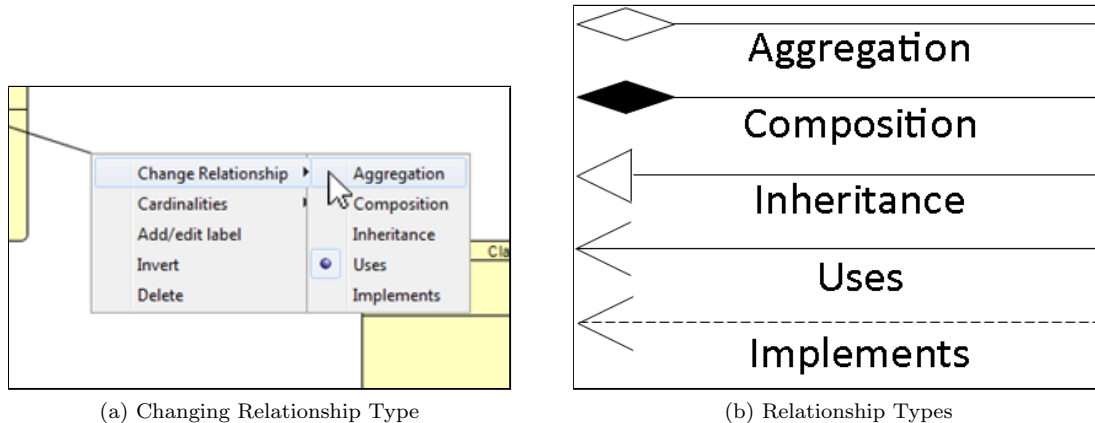
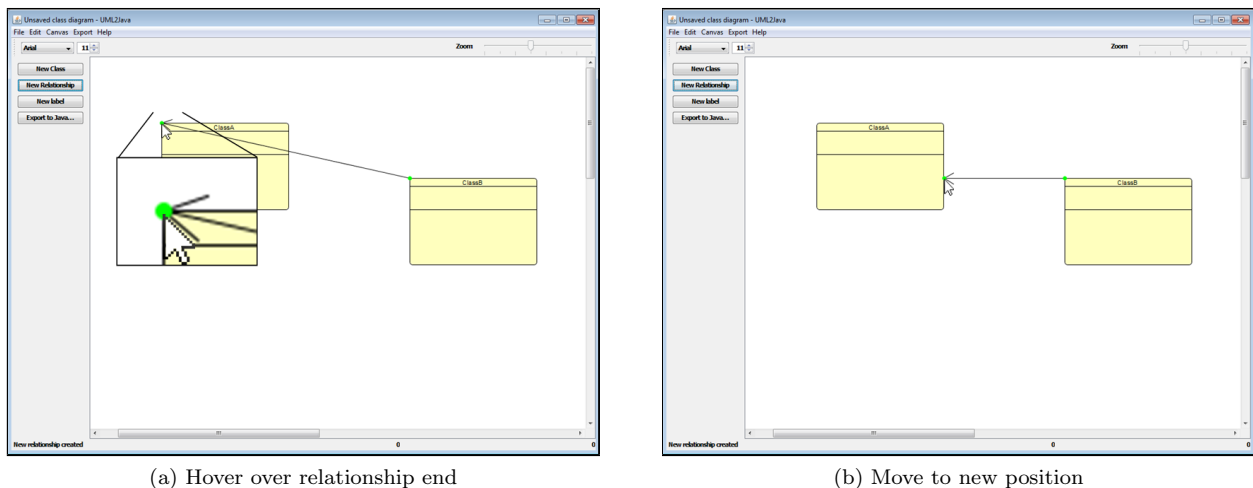


Figure 20: Changing Relationship Types

6.5.2 Moving Relationship Points

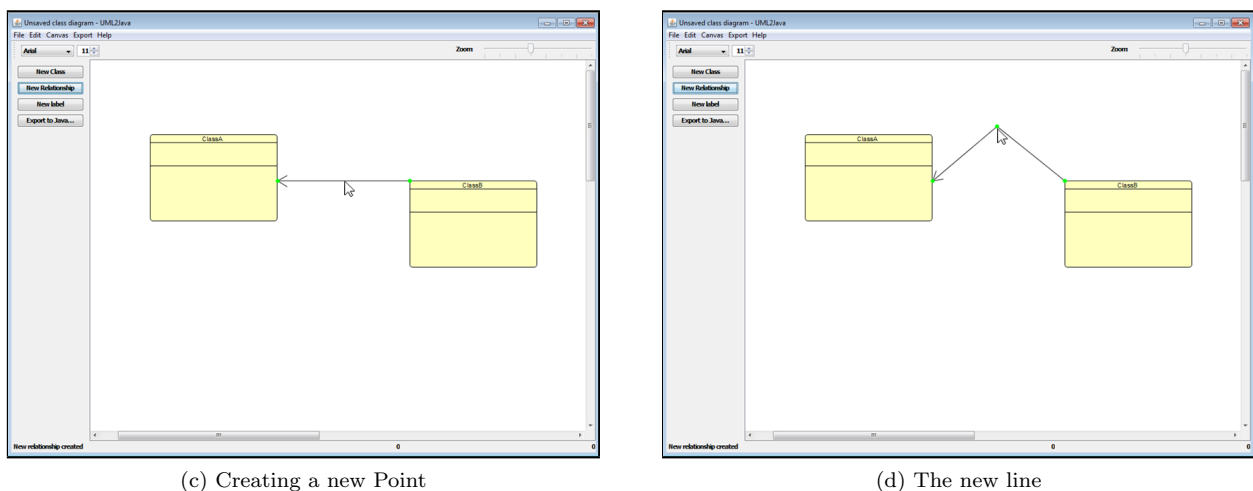
You can edit how the relationships look, as by default they snap to the top-left of the class. Simply hover over the end point of a relationship and green circles will appear over the ends, these will allow you to move the relationship.



(a) Hover over relationship end

(b) Move to new position

You can also add more points to a relationship, allowing them to change direction to move around other classes where necessary. To do this click and hold at a point on the line and drag to a new location. This will create a new point that will have a green circle and allow you to move points on the line.



(c) Creating a new Point

(d) The new line

Figure 21: Moving Relationships and Editing points on the line

6.5.3 Adding a Label to a Relationship

To add a label to a relationship, **Right-Click** and select **Add/Edit Label**. This will allow you to add or edit the current label on the relationship. These are important, as when creating the code files these are used. You can not create labels on Inheritance and Implements relationships

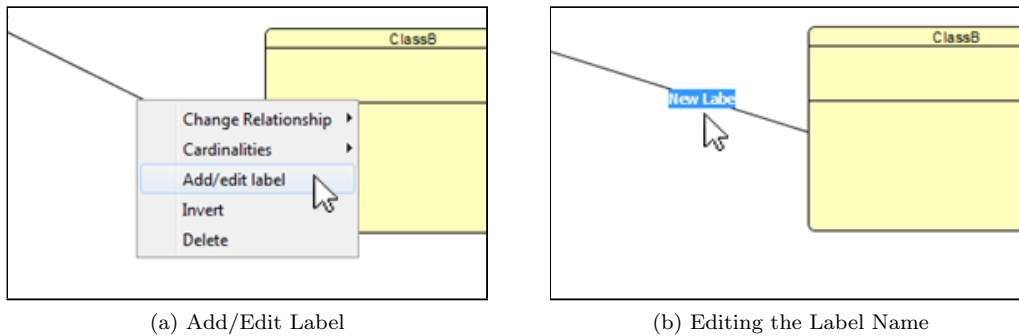
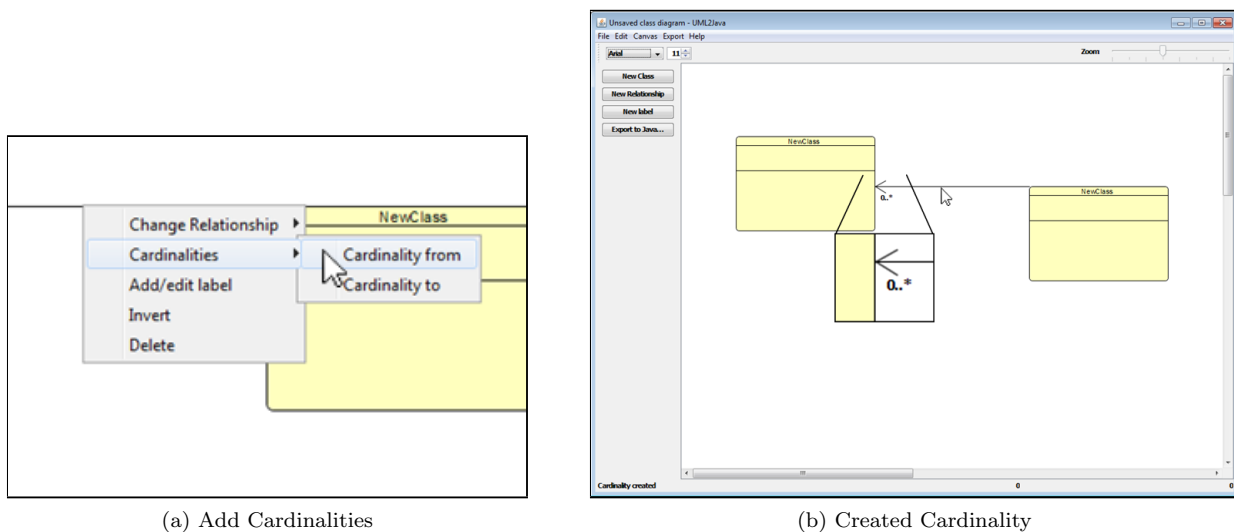


Figure 22: Adding labels to relationships

6.5.4 Adding Cardinalities to the Line

You can add cardinalities to relationships. These include, Zero-to-Many, Many-to-Many and fixed cardinalities. To do this, **Right-Click** on the relationship and hover over **Cardinalities**, a sub-menu will appear with 2 options, **Cardinality From** and **Cardinality To**. Cardinality From will create the cardinality at the Arrow head side, and Cardinality To will create the open end cardinality.



To edit the text, simply double click and this will allow you to edit the text. You will not be able to export the code if cardinalities contain erroneous data, such as words. must either be Whole Numbers or in the form (#..#, or #..*)

Figure 23: Adding Cardinalities to relationships

List of Figures

1	Creating a new Diagram	3
2	Opening an existing diagram	3
3	Saving a diagram	3
4	Printing a diagram	4
5	Exiting the Designer	4
6	Undo last Action	4
7	Redoing an Action	5
8	Printing a diagram	5
9	Fit diagram to visible canvas	5
10	Exporting to Java code	6
11	Export Diagram as Image	6
12	Adding a new class to diagram	7
13	Adding a new Data Field to the Class	8
14	Adding a new Method to the Class	8
15	Adding a relationship between classes	9
16	Editing the class name of a Class	10
17	Editing a data field in a class	10
18	Editing a data field in a class	11
19	Editing a method in a class	11
20	Changing Relationship Types	12
21	Moving Relationships and Editing points on the line	12
22	Adding labels to relationships	13
23	Adding Cardinalities to relationships	13

Index

Adding to diagram

- Classes, 7
- Data Fields/Variables, 8
- Methods, 8
- Relationships, 9

Canvas Settings

- Fit to Diagram, 5
- Resize, 5

Edit

- Adding Cardinalities, 13
- Redo, 5
- Undo, 4

Editing

- Class Name, 10
- Data Fields/Variables, 11
- Methods, 11
- Moving the Classes, 10
- Relationships, 12
 - Adding Label, 13
 - Moving Relationship Points, 12
 - Relationship Type, 12

Exporting

- Code, 6
- Image, 6

File

- Create New diagram, 3
- Exiting UML2Java, 4
- Open Saved diagram, 3
- Printing, 4
- Saving, 3