**Synopsis**

The learning objectives of this seminar are same as that of the lecture for this week. Please if you have not done, read the lecture slides before you attempt seminar or do both in parallel. This laboratory session is intended to enable you to understand and design UML diagrams.

1. Download and Install Violet UML Editor

2. Create class diagram

3. Create use case diagram

# **Download and Install Violet UML Editor**

What is Violet? Violet is a UML editor with these benefits:

1. It is very easy to learn and use
2. It draws nice-looking diagrams of the most commonly used types (class, sequence, state, etc.)
3. It is completely free (includes source, distributed under the GNU General Public License)
4. It is cross-platform

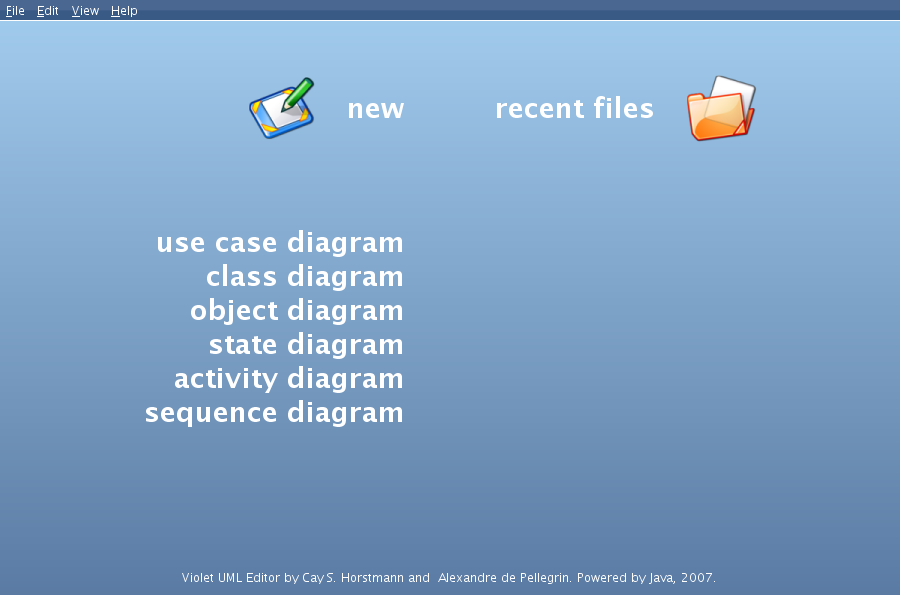
Violet is intended for students, teachers, and authors who need to produce simple UML diagrams quickly. It is not intended as an industrial strength tool. Programs such as Rational Rose, StarUML, ArgoUML, and so on, are fine choices for serious users of UML. However students and casual users may not like them because they are too hard to learn or too expensive. If you just want to draw simple UML diagrams without too much fuss, chances are you'll like Violet. If you have more serious needs, check out one of the other programs.

Downloading and Running Violet

1. Violet comes as an executable Java archive file which you can download here: <https://www.d.umn.edu/~gshute/cs2511/javalabs/uml_diagrams/violet.jar>
2. You must have Java installed on your machine.
3. On some operating systems you can launch Violet by ***double-clicking its icon (i.e downloaded .jar file, it works on my computer)***. If that doesn't work for you, open a command window, change to the folder where your violet.jar file is located, and run the command:

***java -jar violet.jar***

You should see a screen that looks like:

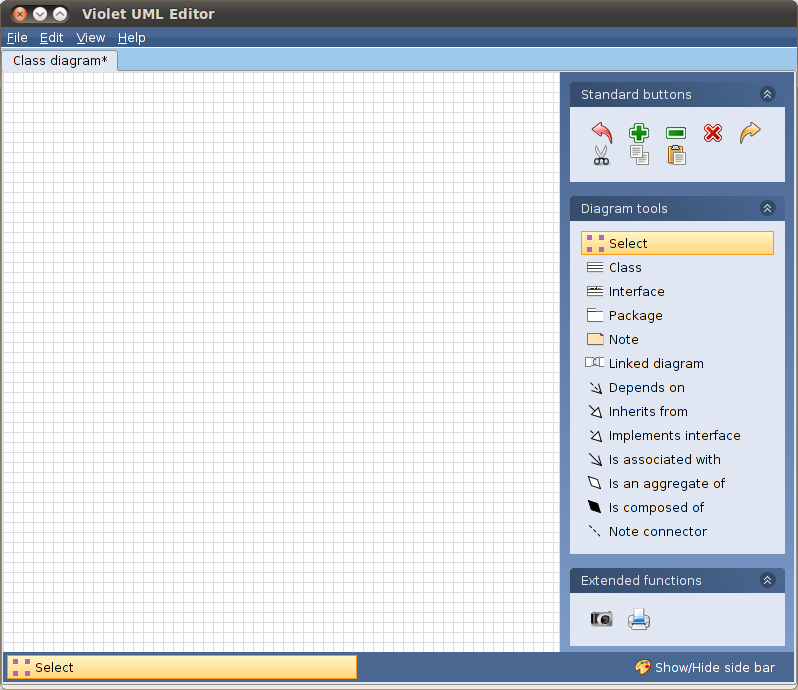


# **Class Diagrams**

This section describes basic operations for class diagrams in Violet, such as creating a new diagram and manipulating its parts. This short overview is not a complete description of Violet; it is just intended to get you started.

**Creating a New Class Diagram**

Select File > New > Class Diagram to create a new drawing area laid out in a grid. Note the diagram tools arrayed to the right of the drawing area window:

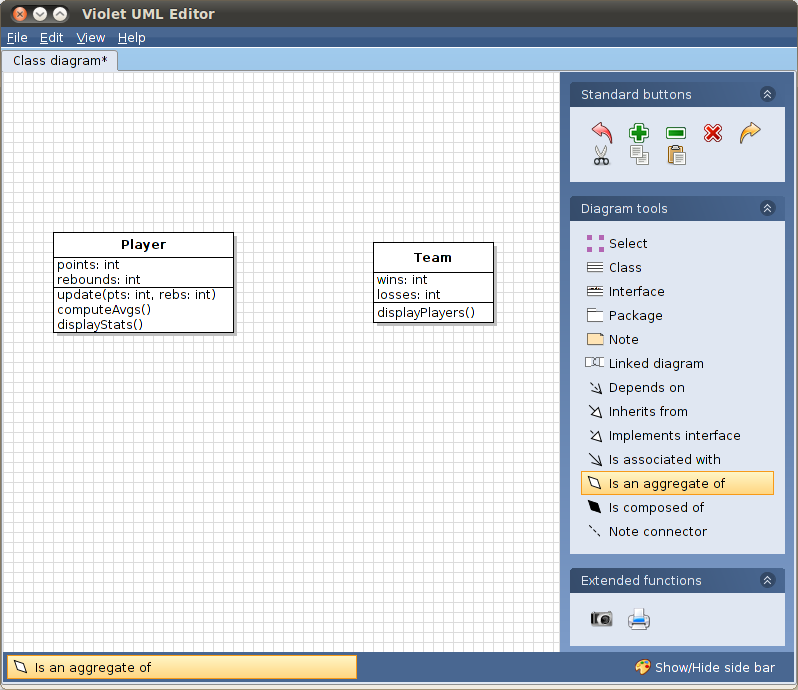


**Creating Nodes**

Click to select the "Class" tool. Any time a tool is selected, that is the type of diagram component (whether it is a class, relationship arrow, etc.) that will be created when you click again.

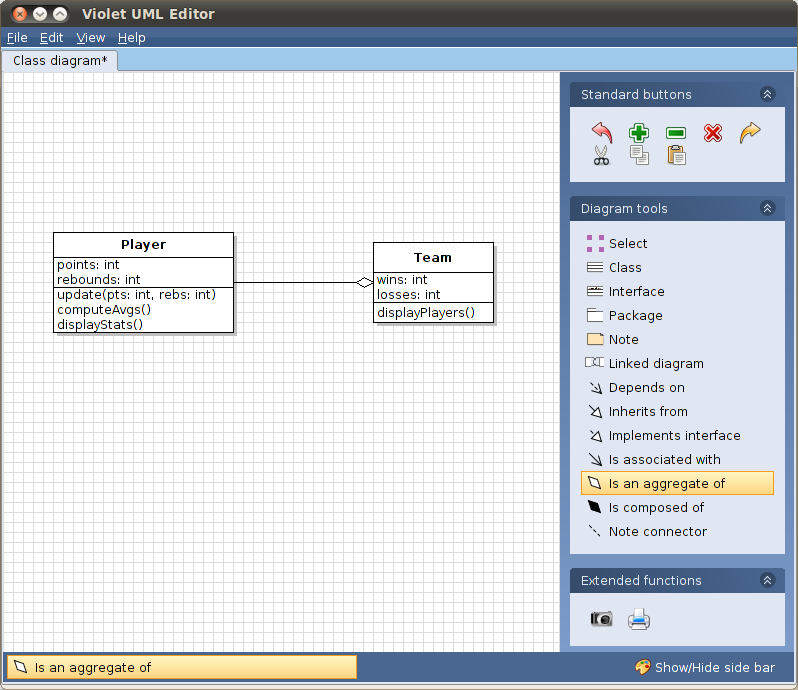
Locate a place in the window where you want to put a node (class icon), and click there. An empty rectangle will appear. You can move the node around by dragging it with the mouse. If you right-click in the rectangle you get a Properties dialog that will allow you to enter a class name, attributes, and methods.

Experiment until you get two nodes that look like this:



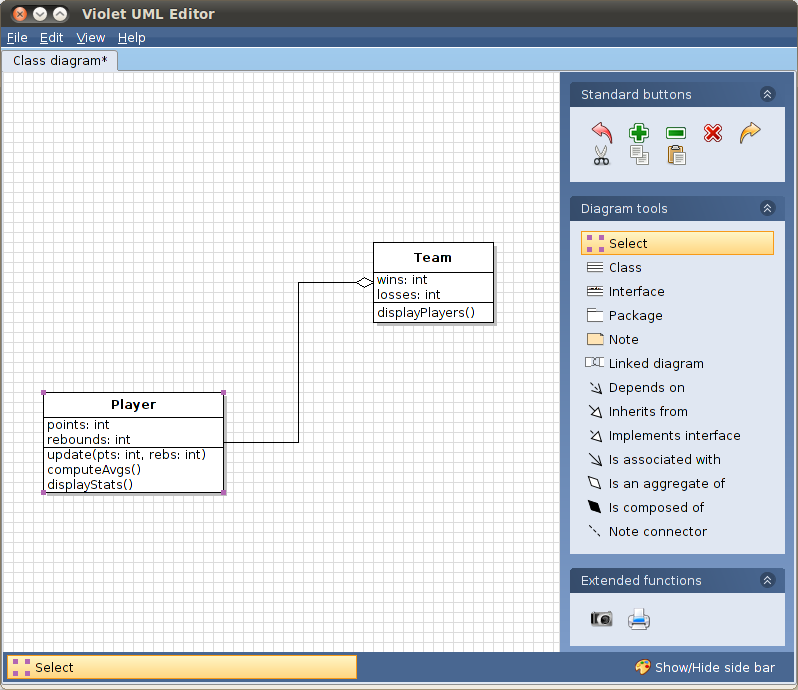
**Connecting Nodes with Edges**

Since a team is an aggregate of players, select the "Is an aggregate of" tool, click anywhere in the Team class, and drag to anywhere in the Player class. If the classes occupy the same horizontal strip within the diagram, you will get an edge (connecting line) like:



**Moving Nodes**

Click the Select tool and drag the Player class icon down an inch or two:

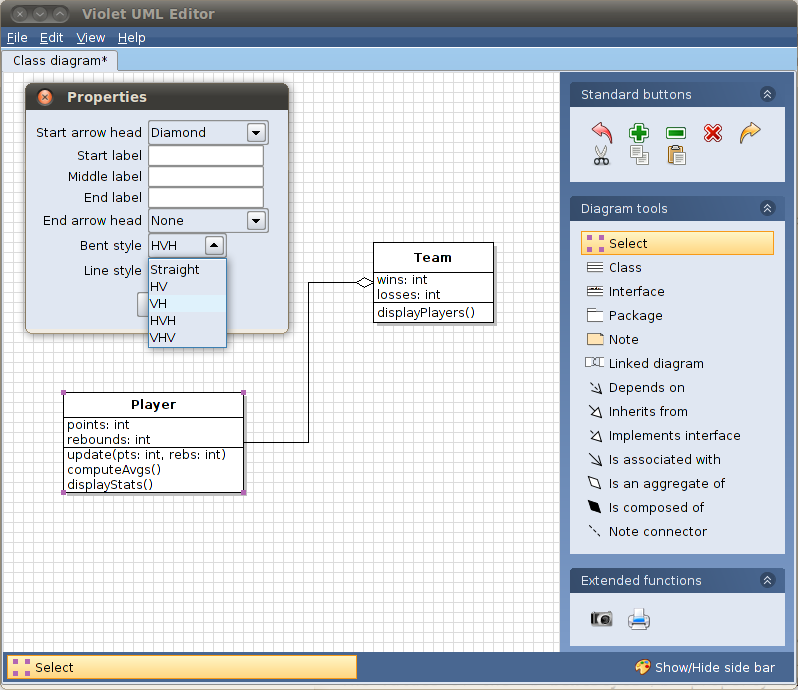


**Edge Properties**

All edges have a bentStyle property that defines the style of the edge.

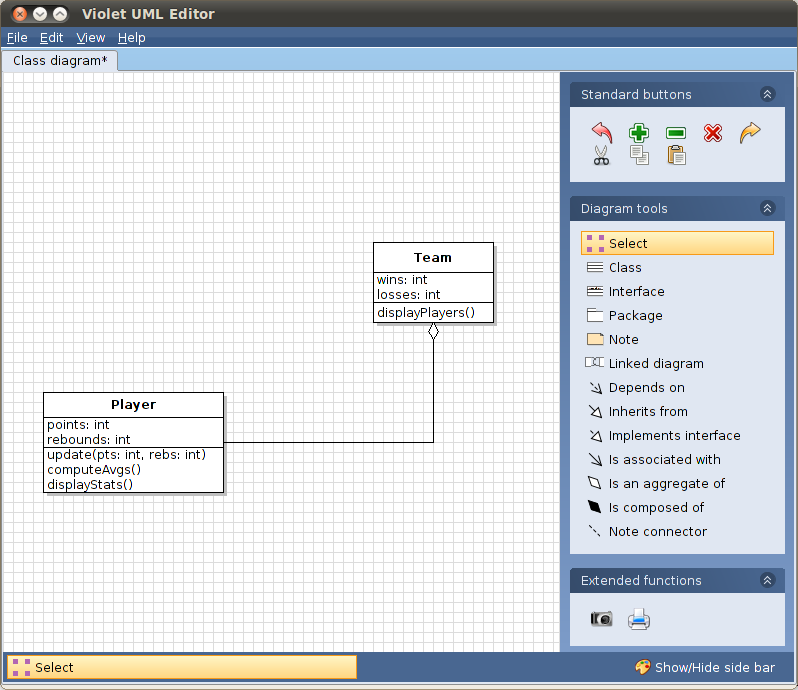
The bentStyle of our edge is currently HVH (horizontal-vertical-horizontal).

You can control the bentStyle of an edge by right-clicking on it to get the edge's property editor:



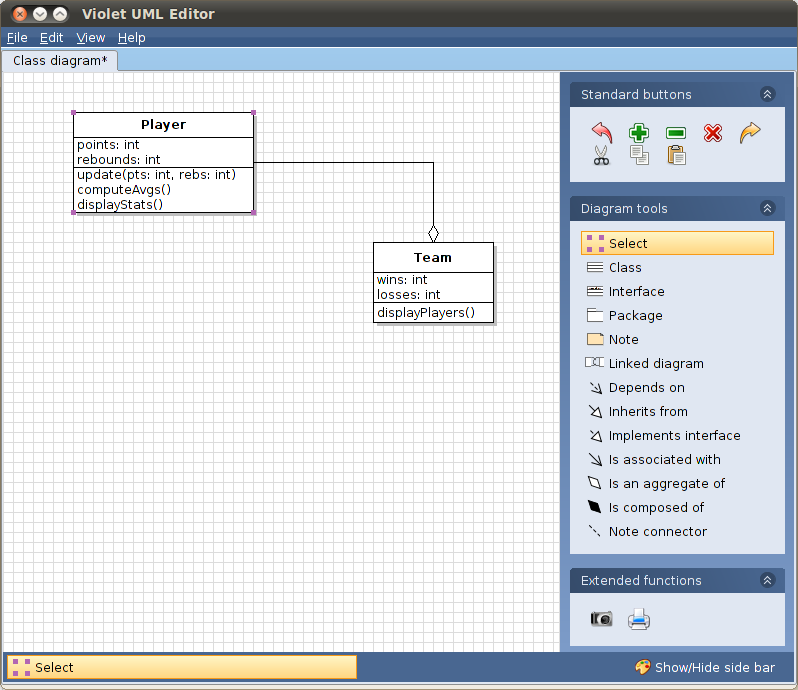
**Changing Edge Properties**

Here the edge's bentStyle property has been changed to VH (vertical-horizontal; the vertical emanates from the origin class icon, which is Team):



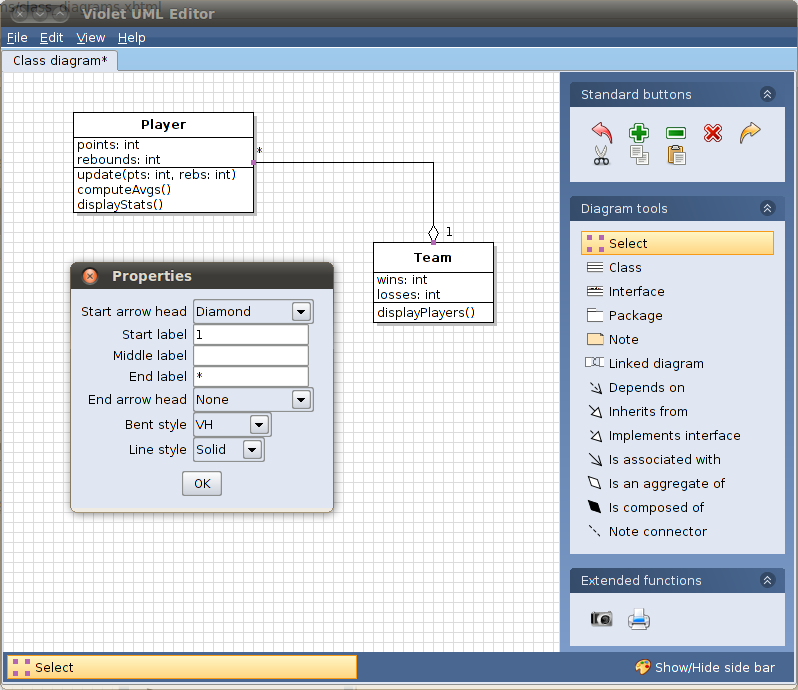
**Maintaining Edge Properties**

Violet will attempt to maintain the bentStyle properties of an edge as you move associated nodes around:



**Class Multiplicities**

You can use an edge's property editor to indicate the class multiplicities for the nodes that it connects:

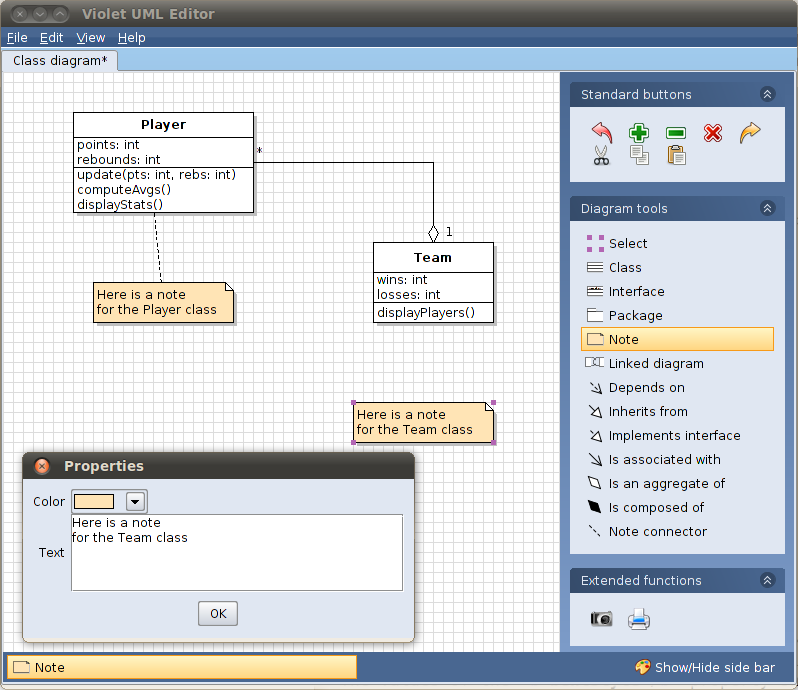


**Diagram Notes**

You can add notes to a diagram by selecting the Note tool and clicking anywhere in the diagram.

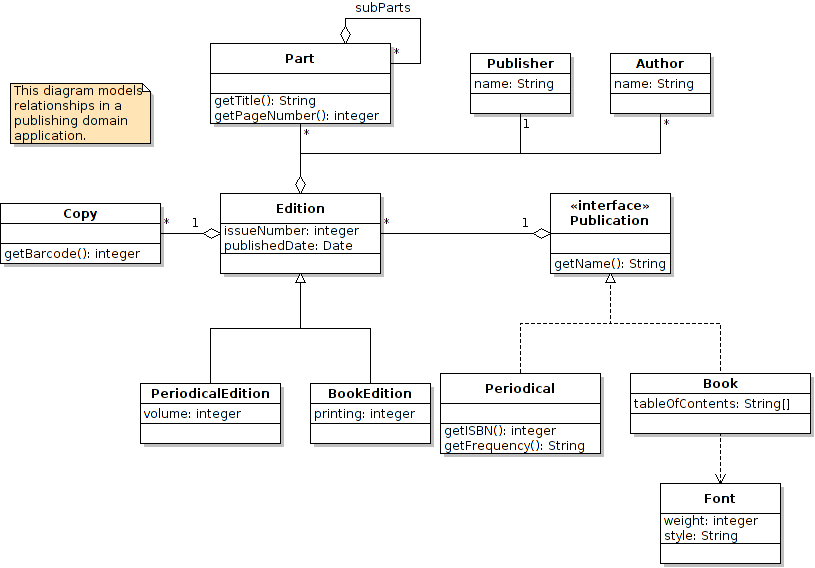
Then, right-click on the note icon to get the note's property editor.

Finally, select the Note connector tool to connect the note to some aspect of the diagram:



**Class diagram exercise**

Design a class diagram as shown below:



# **Use case diagrams**

The creation is a use case diagram is done in a similar fashion to a class diagram by choosing File > New > Use case Diagram and then using different elements (e.g. actors, ovals, and relationship symbols) from the right pane.

**Use case diagram exercise**

Draw a use case diagram for ATM Machine. Please refer the lecture slices. One good example is also given on this web page: <https://www.uml-diagrams.org/bank-atm-uml-use-case-diagram-example.html>