

## Lab 1 UML exercise

### Orientation/reference

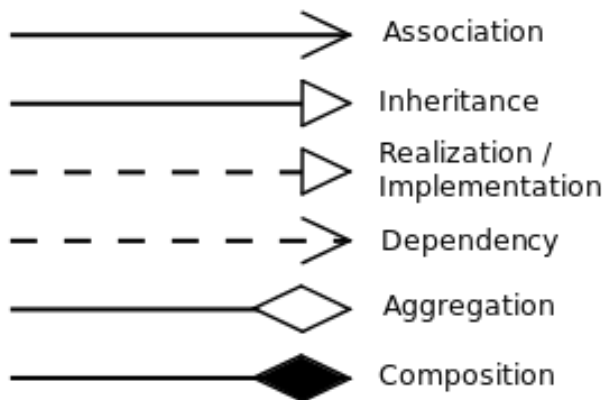
Throughout this quarter, we will be using UML (Unified Modeling Language) to help understand and express software engineering design decisions.

From Wikipedia: “In software engineering, a **class diagram** in the Unified Modeling Language(UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.”

We will only use a small subset of UML to start with and slowly add more features as the quarter proceeds. In general, we will be using the following symbols to denote visibility:

<b>+</b>	<b>public</b>
<b>-</b>	<b>private</b>
<b>#</b>	<b>protected</b>

And we will also eventually use the following symbols to denote relationships (keep this as a reference for later):



**Resources:**

This quarter, we will be using the freely available UML editor, yEd Graph Editor (<https://www.yworks.com/products/yed>). This tool has many features. The manual is available at <http://yed.yworks.com/support/manual/uml.html>. You can install it on your computer, or on a lab computer you can type “yed” to launch it.

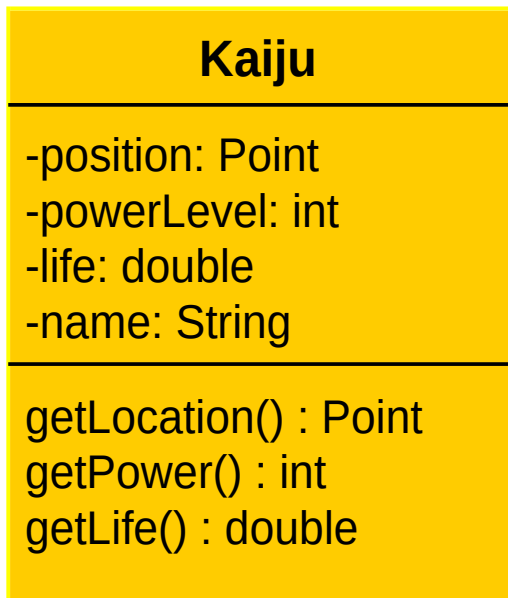
**Task:**

Today’s task is intended to start orienting you to the tool and UML. Create a new UML diagram for your point class (task 2 of main lab 1). There are step by step instructions with visual aids at the end of this lab.

Your diagram will be very simple. For example, for a class to represent a strange beast called a “kaiju,” you might have:

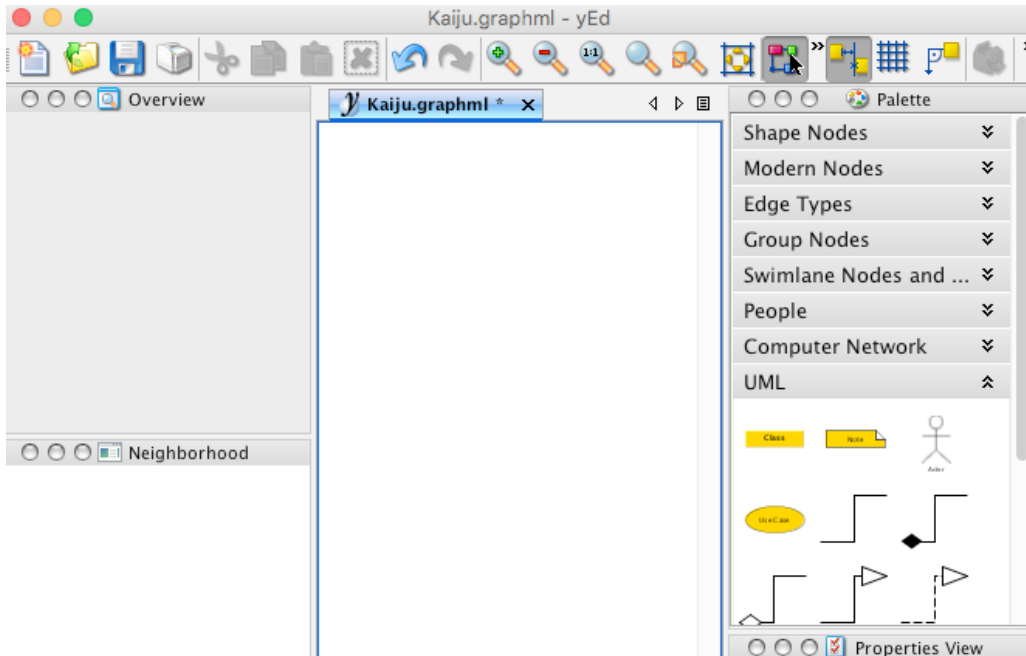
```
class Kaiju {  
  
    private Point position;  
    private int powerLevel;  
    private double life;  
    private String name;  
  
    public Kaiju(Point start, int power, String name, double life) {  
        this.position = start;  
        this.powerLevel = power;  
        this.name = name;  
        this.life = life;  
    }  
  
    public Point getLocation() { return position; }  
    public int getPower() { return powerLevel; }  
    public double getLife() { return life; }  
}
```

Resulting in a diagram like this:

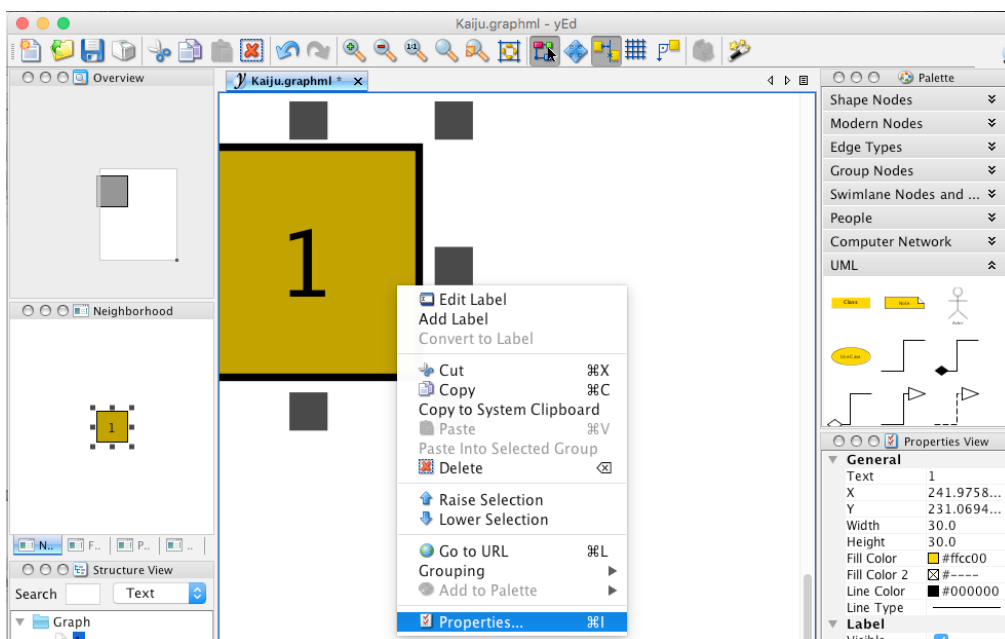


Here is a reference of the necessary steps using the yEd tool (after selecting “new”).

1) Add a new class by using the right hand “palette” menu, and by selecting, the UML tab and dragging a new “class” into the main graph area.



2) Edit the properties of the new class by right clicking on the new class node



3) Select the “UML” tab in the properties window and add the class attributes and methods using the notation listed above.

