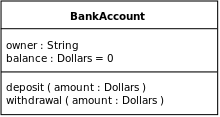
Class Diagrams



Class diagrams are broken up into 3 different sections. The top section contains the class name and is centered in the box. The second or middle section contains the attributes of the class, or the variable names and types. The third or bottom section contains the methods that the class will use. When diagraming the middle and bottom section of the class diagram the format is that both will be lower case and aligned to the left side.

Symbols can be placed before the attributes and methods to specify their visibility. A ‘+’ sign means public, a ‘-‘ sign means private, a ‘#’ means protected, a ‘/’ means it is derived or can be combined with others, and a ‘ ~ ’ means it is a package. Another piece of information that can be added to a variable or method is to underline the name. By underlining the name you are identifying the scope of the member (variable or method). If the name is underlined it means that the value of the member is the same outside of the class while if it is not underlined then the member only has the value inside of the class.

When connecting the classes in the diagram how they are connected will represent their relationship. A bi-directional and uni-directional is shown by using just a straight line to connect two classes.

An aggregation relationship is represented by a hollowed out diamond next to a class and then a line that connects the diamond to another class. What this represents is that the class with the hollow diamond can exist without the other class.

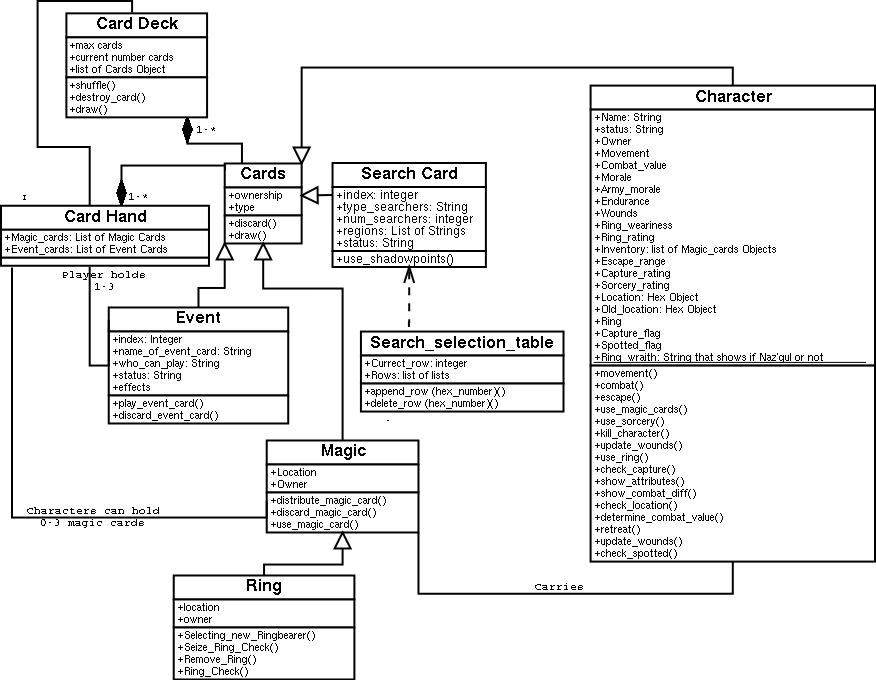
The composition relationship is basically the opposite of aggregation. Composition is a filled in diamond next to the class and has a line connecting the diamond to another class. This represents that the class with the diamond cannot work or exist without the other class. A great example of this would be how a computer cannot work without a motherboard.

A line with a hollow arrow at the end will indicate that the class that is pointing an arrow at the other is a subclass. The class that has the arrow pointed at it is considered a super class as well. The subclass will inherit traits from the super class.

The last kind of relationship is known as dependency. This relationship is drawn out using a dotted line with an open arrow next to the needed class. The relationship is not as strong as composition because if one class is removed the other class will still remain but will not be able to function correctly.

Multiplicity is added next to the classes where they lines connect them. Multiplicity is used to show how many objects participate in the association. This is represented by numbers. 0..1 is no or one instance, 1 is exactly one instance, 0..\* is zero or more instances, and 1..\* is one or more instances.

Sandbox from the internet



Sandbox I created

