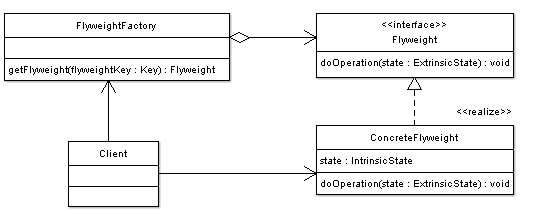
# Flyweight Pattern

**Definition**: The pattern for saving memory (basically) by sharing properties of objects. Imagine a huge number of similar objects which all have most of their properties the same. It is natural to move these properties out of these objects to some external data structure and provide each object with the link to that data structure.

Flyweights are stored in a Factory's repository. The client restrains herself from creating Flyweights directly, and requests them from the Factory. Each Flyweight cannot stand on its own. Any attributes that would make sharing impossible must be supplied by the client whenever a request is made of the Flyweight.

**Problem**: Designing objects down to the lowest levels of system “granularity” provides optimal flexibility, but can be unacceptably expensive in terms of performance and memory usage.

**Solution**:



* **Flyweight** — Declares an interface through which flyweights can receive and act on the extrinsic state.
* **ConcreteFlyweight** — Implements the Flyweight interface and stores **intrinsic** state (sharable among objects). A ConcreteFlyweight object must be sharable. The Concrete flyweight object must maintain state that it is intrinsic to it, and must be able to manipulate state that is extrinsic. In a war game example, a graphical representation is an intrinsic state, where location and health states are extrinsic. Soldier moves, the motion behavior manipulates the external state (location) to create a new location.
* **FlyweightFactory** — The factory **creates and manages flyweight objects**. In addition, the factory ensures the sharing of the flyweight objects. The factory maintains a pool of different flyweight objects and returns an object from the pool if it is already created, adds one to the pool and returns it in case it is new.
* **Client** — A client maintains references to flyweights in addition to computing and maintaining extrinsic state.

A key concept is the distinction between “intrinsic” and “extrinsic” state. Intrinsic state consists of information that is independent of the flyweight’s context — information that is sharable (i.e. each Icon’s name, width, and height).

* Icon (Flywight): store sharable states
* FlyweightFactory: creates and manages Icon