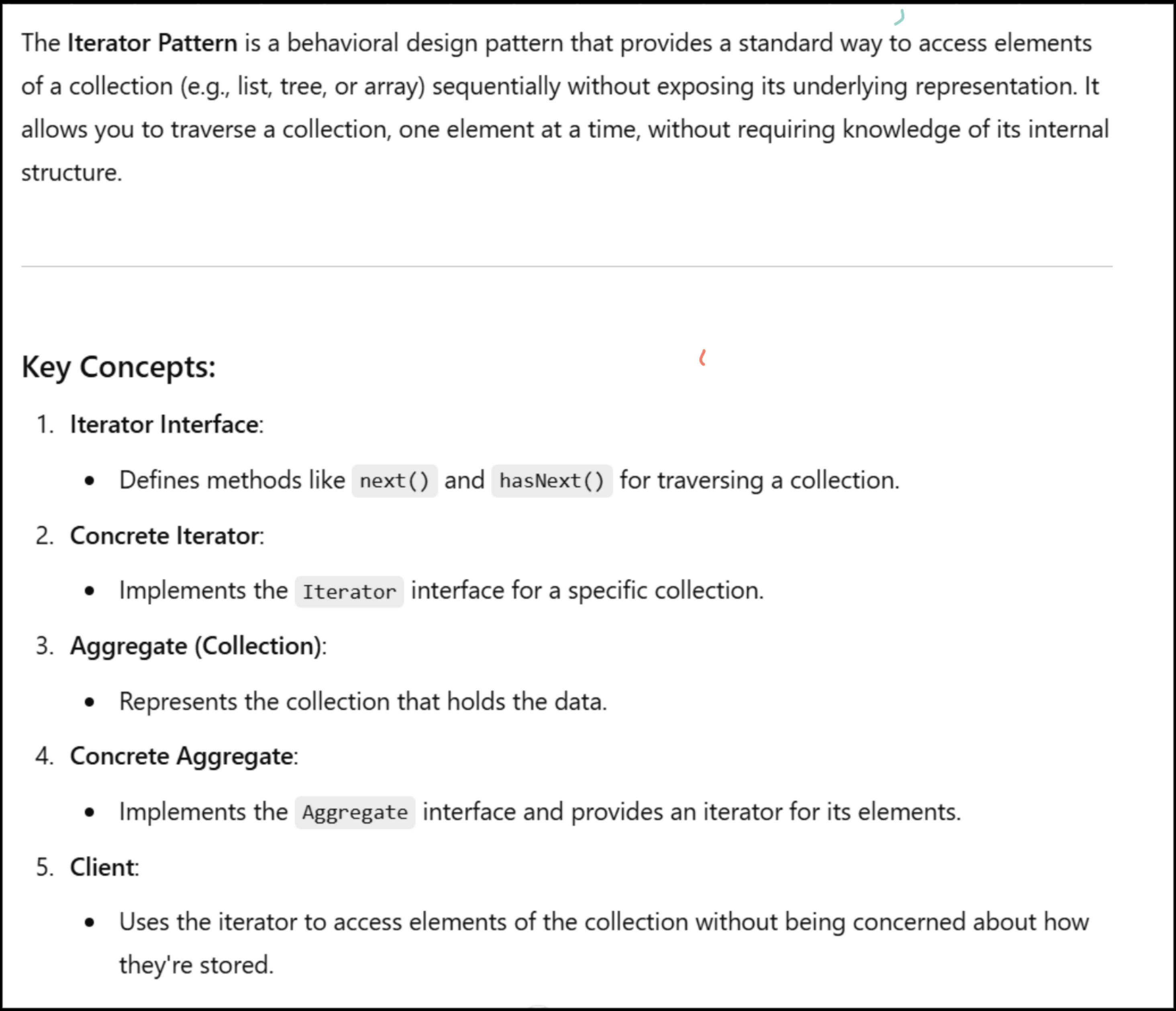
[Iterator Design Pattern]

* In JS, iterators are based on it



The **Iterator Pattern** is a behavioral design pattern that provides a standard way to access elements of a collection (e.g., list, tree, or array) sequentially without exposing its underlying representation. It allows you to traverse a collection, one element at a time, without requiring knowledge of its internal structure.

Key Concepts:

1. **Iterator Interface**:
   - Defines methods like `next()` and `hasNext()` for traversing a collection.
2. **Concrete Iterator**:
   - Implements the `Iterator` interface for a specific collection.
3. **Aggregate (Collection)**:
   - Represents the collection that holds the data.
4. **Concrete Aggregate**:
   - Implements the `Aggregate` interface and provides an iterator for its elements.
5. **Client**:
   - Uses the iterator to access elements of the collection without being concerned about how they're stored.

* STL iterator class