

Group 2

Achievement F1 - Design Patterns

Advanced Software Design

Uppsala University

Mikael Hagafors Sara Back Daniel Owen-Berghmark Henrik Bihlar

December 6, 2016

Null Object

The *Null Object pattern* is used to represent the absence of an object and implement the desired function in the Null Object, instead of explicitly checking for the absence of the object.

At the moment our system would return null if there is no match made, i.e. all possible responders answers negatively to the request. Instead we can use a Null Object to represent the absence of a match.

Iterator

The *Iterator pattern* can be used to give access to the data without exposing the structure of the object.

In the system, a good use for the iterator pattern would be to change all `List<Match>` to iterators. This change would allow us to represent the matches as any kind of collection and change the representation at any time without breaking the system.

Observer Pattern

The *Observer pattern* solves the problem of updating the state of all dependent objects when an object changes state.

Introducing Observer patterns would allow our system to update requesters and responders about new matches and feedback as they are created.