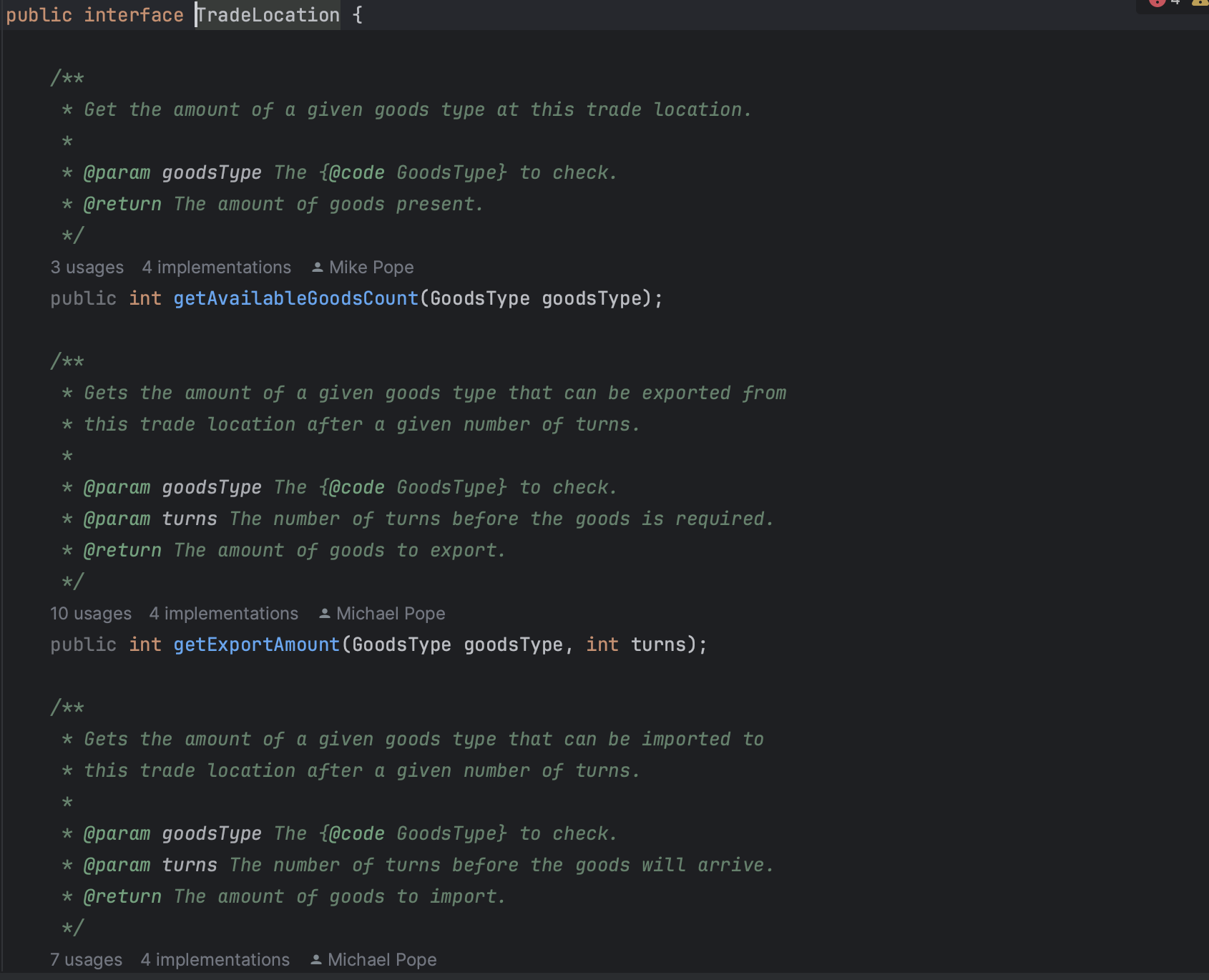
Wilker Martins -58535

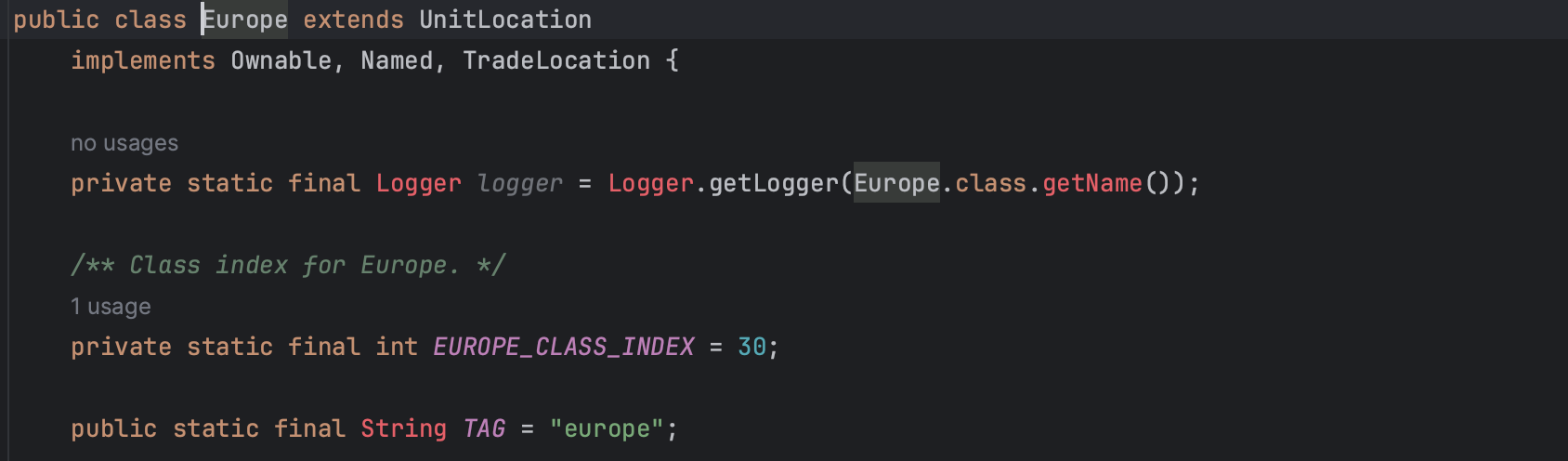
**Design Patterns**

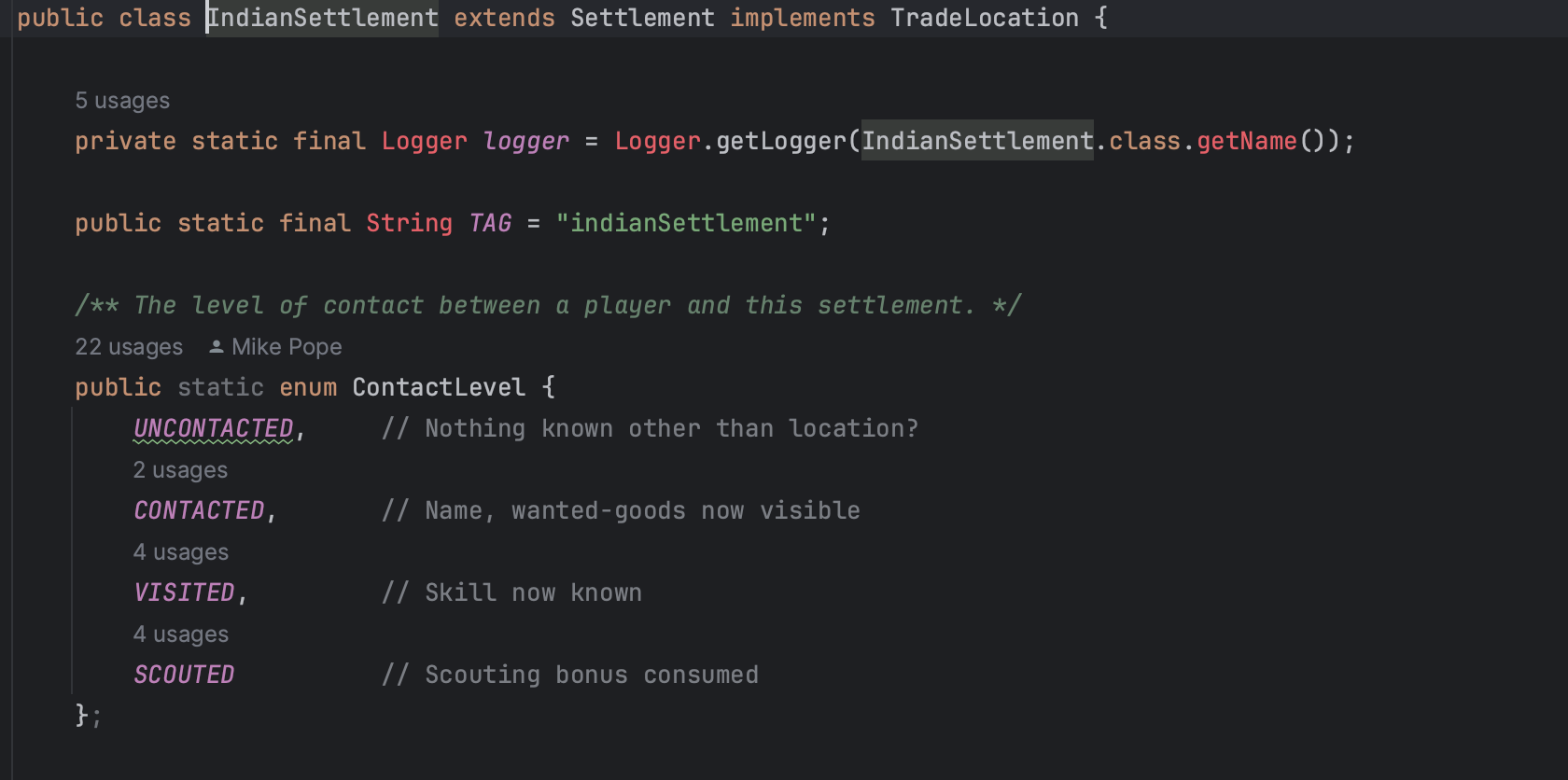
**1-Factory Method** (src/net/sf/freecol/common/model/TradeLocation.java)



Classes that implements the interface

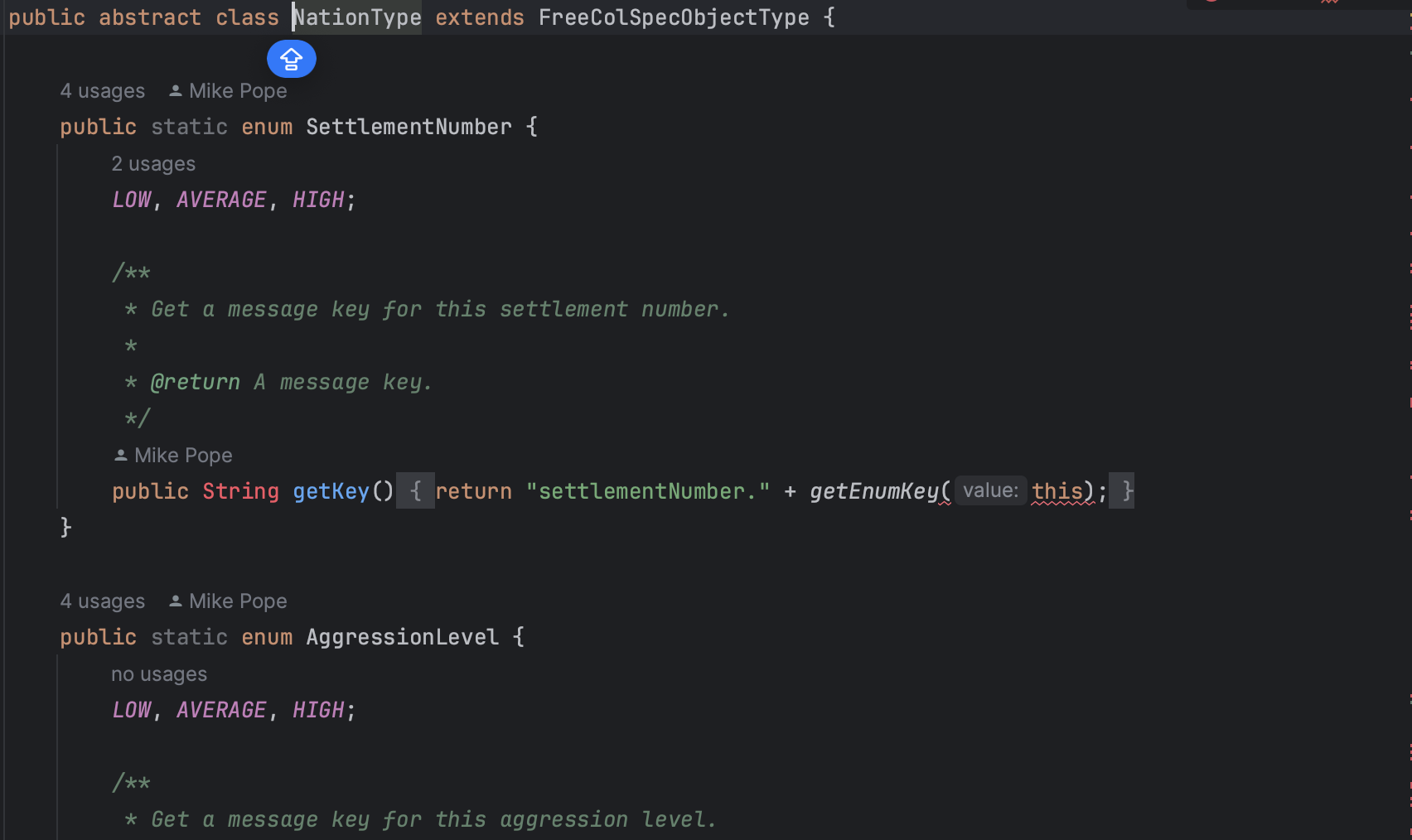
(src/net/sf/freecol/common/model/Europe.java)



(src/net/sf/freecol/common/model/IndianSettlement.java)

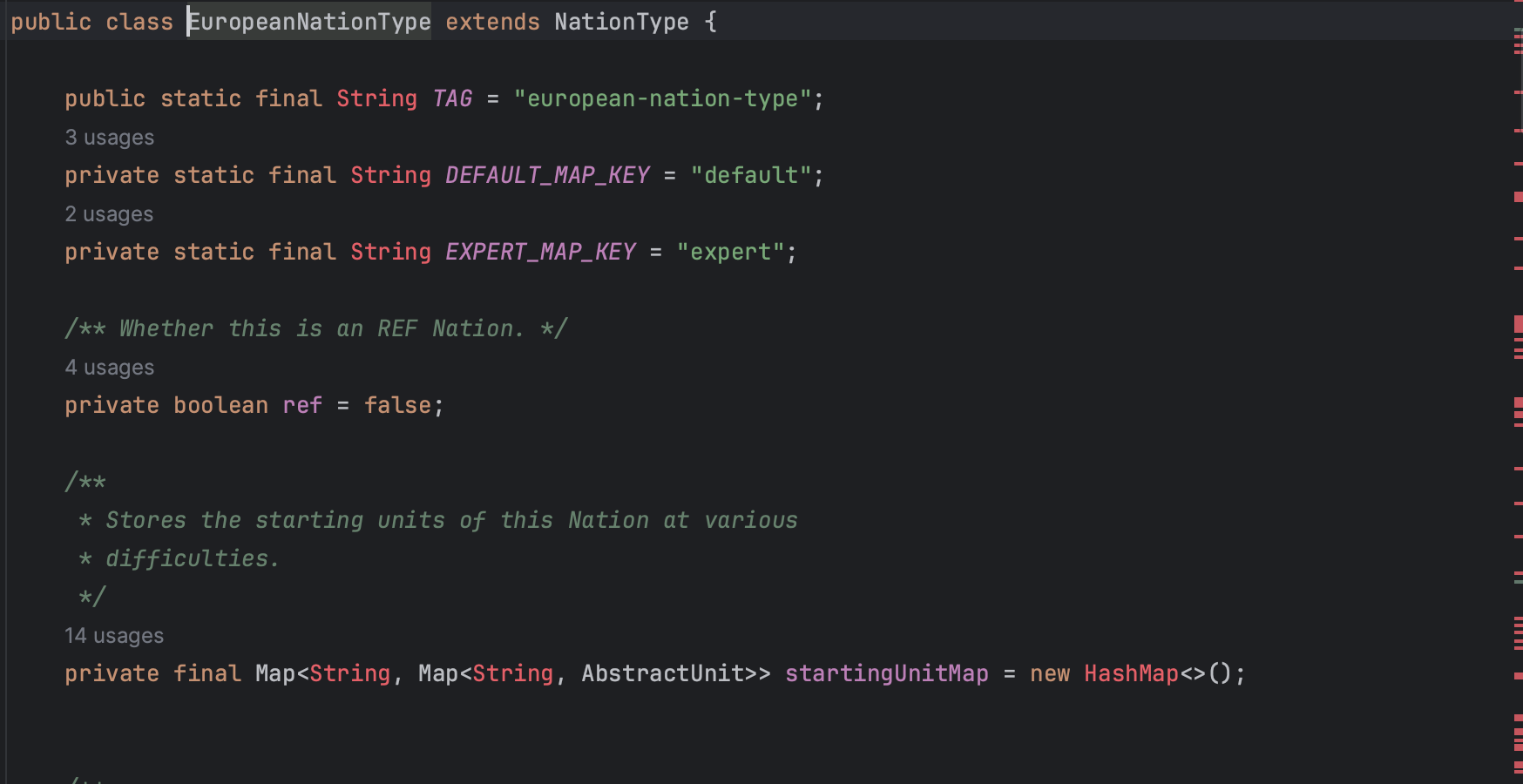
In the **TradeLocation** interface, you're abstracting the process of creating **TradeLocation** objects. This abstraction allows you to decouple the client code from the concrete classes (**European** and **IndianSettlement**) that implement the interface.

**2-Template Method Pattern** (src/net/sf/freecol/common/model/NationType.java)

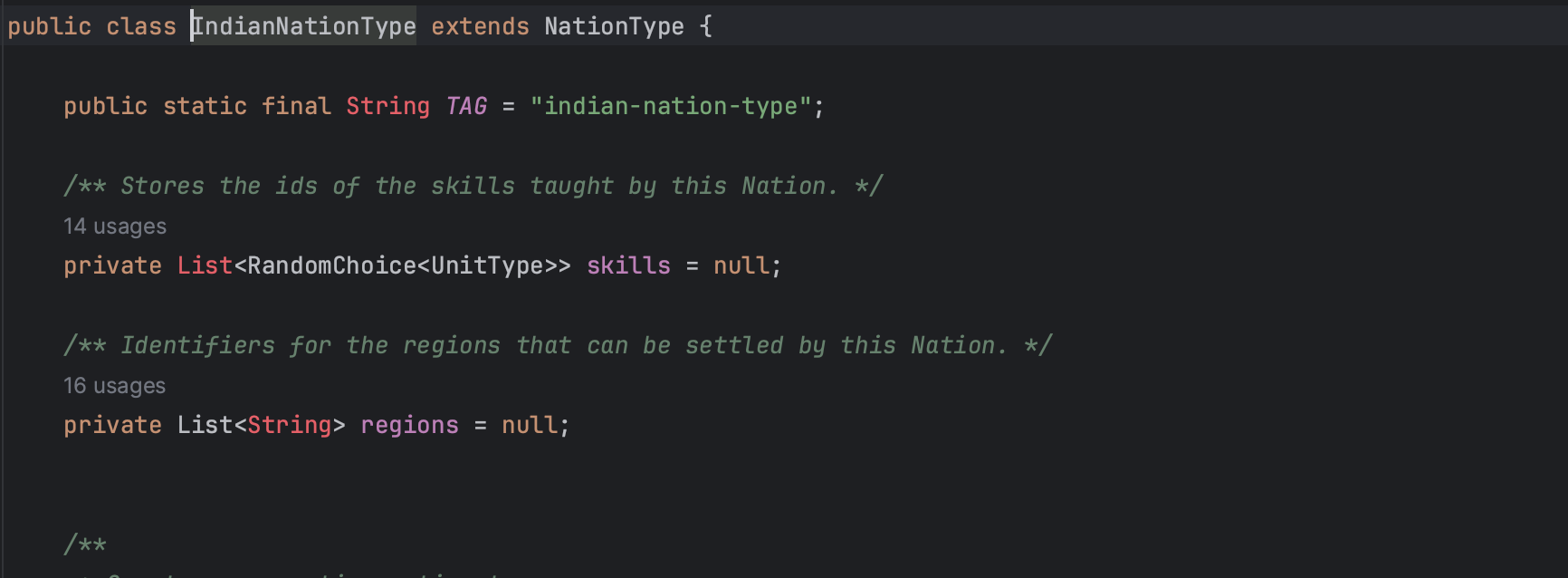


Classes that Extend the Super Class NationType

(src/net/sf/freecol/common/model/EuropeanNationType.java)



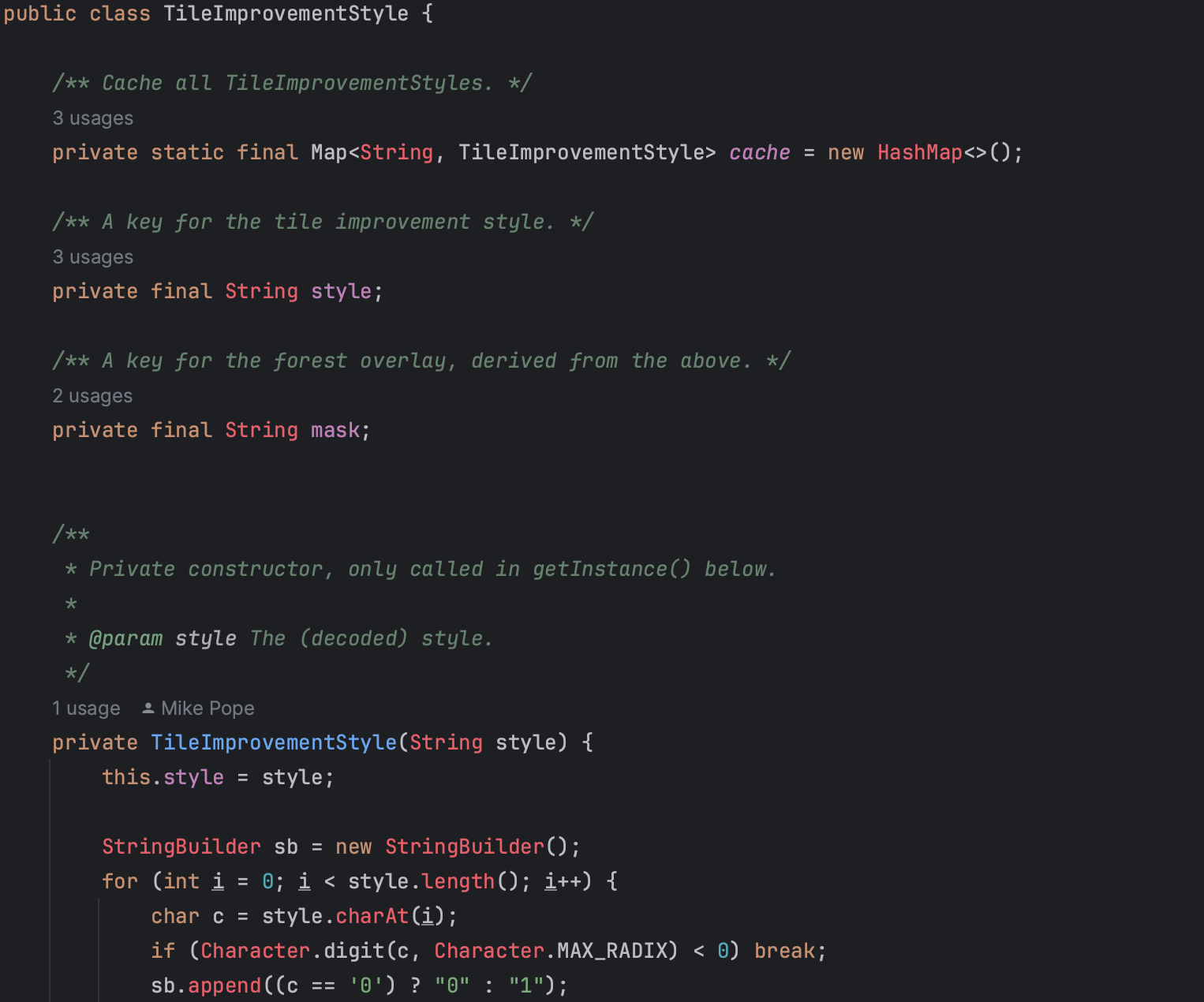
(src/net/sf/freecol/common/model/IndianNationType.java)



By defining a template method in the abstract class **NationType**, you provide a common structure for creating different nation types. This common structure contains the algorithm's skeleton, including the sequence of steps required to create a nation type. This structure is reused by all subclasses, eliminating the need to duplicate code.

**3-Singleton**

(src/net/sf/freecol/common/model/TileImprovementStyle.java)



The **TileImprovementStyle** class has a private constructor, which means it cannot be instantiated directly from outside the class. The class maintains a cache (represented as the **Map** named **cache**) to store already created instances of the **TileImprovementStyle** class.