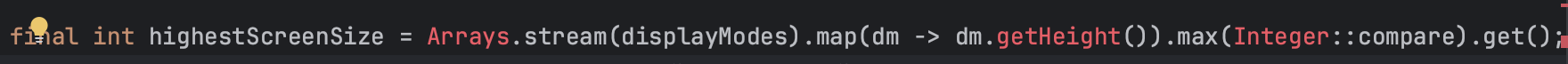
Wilker Martins -58535

**Code Smells:**

**1-Message Chain** (src/net/sf/freecol/common/util/Utils.java)

There are a series of method calls or property accesses chained together, typically separated by dots. To refactor the code and avoid the message chain, you can break it down into separate steps using meaningful variable names. This will make the code more readable and easier to understand.

**2-Large Classe** (src/net/sf/freecol/common/util/CollectionUtils.java)



There are a situation where a class has grown too large and complex, making it difficult to understand, maintain, and modify. By breaking down a large class into smaller, focused, and cohesive classes, we can improve code maintainability, readability, and reusability.

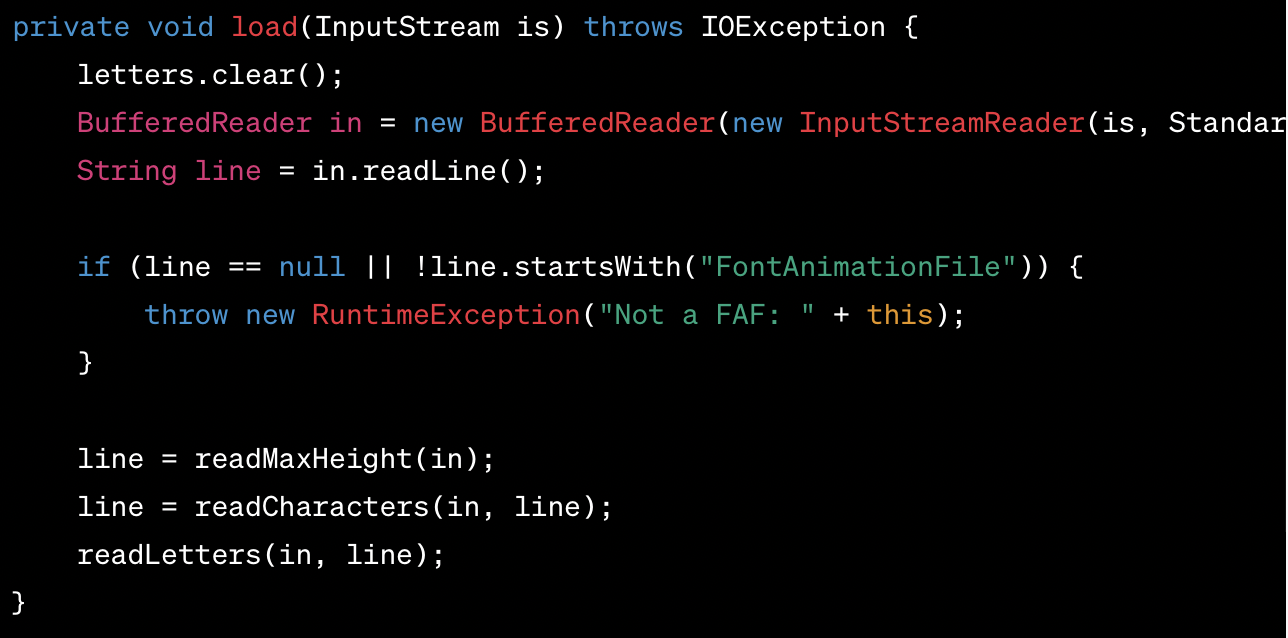
**3-Long Method** (src/net/sf/freecol/common/resources/FAFile.java)

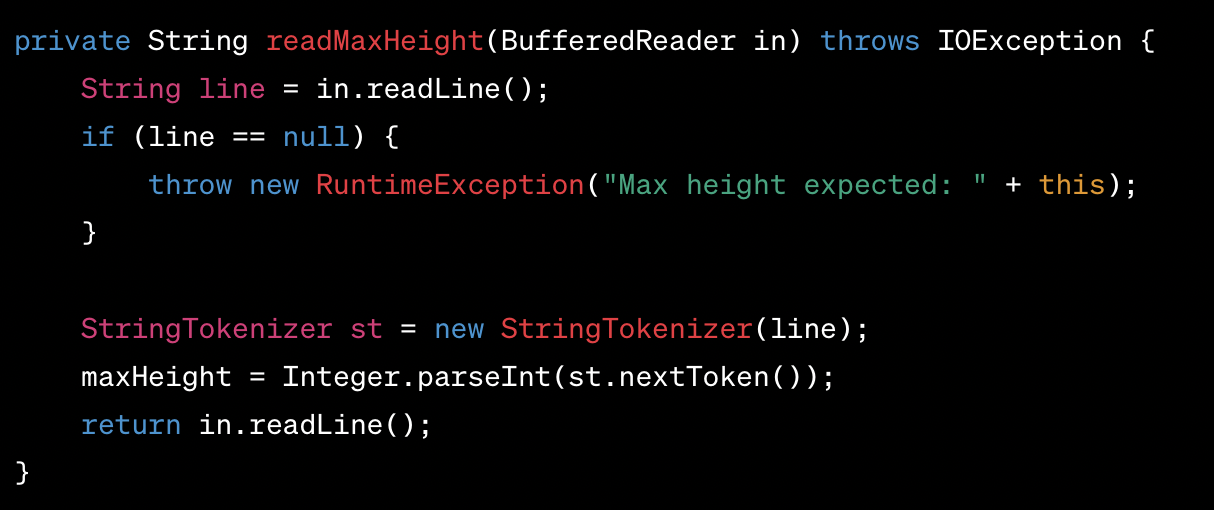


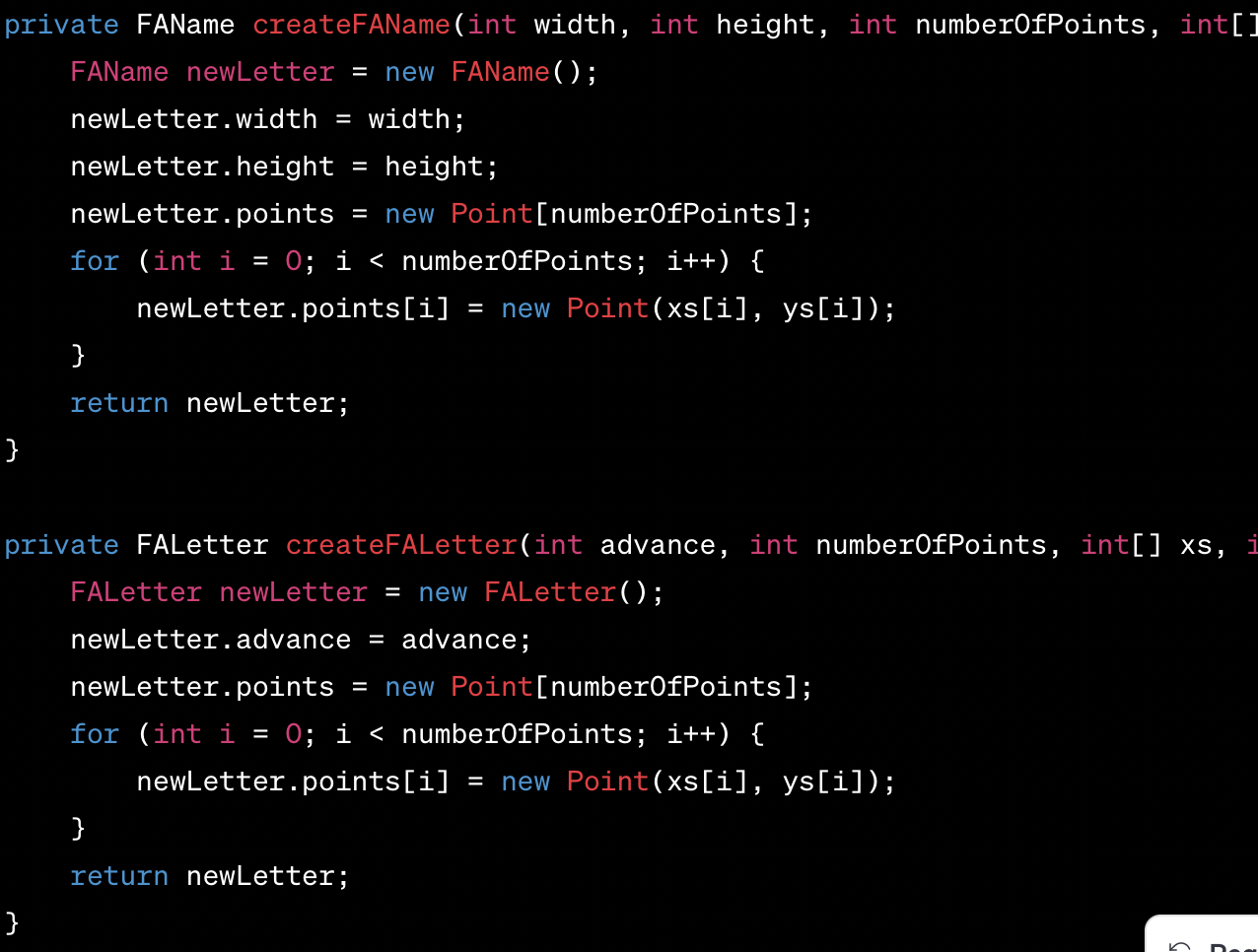
...

The method is overly long and complex. This can make the code harder to read, understand, and maintain. To address this code smell, we can break down the method into smaller, more focused methods, each responsible for a specific part of the overall process.

For exemple:







The **load** method has been split into smaller methods with clear responsibilities. This makes the code more organized, easier to read, and simpler to maintain.