

Code Smells

1º - Long Method

```
691      CommandLineParser parser = new DefaultParser();
692      boolean usageError = false;
693      >      try {...} catch (ParseException e) {
939          System.err.println("\n" + e.getMessage() + "\n");
940          usageError = true;
941      }
```

Este bloco de try catch tem 300, pelo que se pode classificar como um Long Method, visto que é bastante extenso.

A sua localização é “src/net/sf/freecol/FreeCol.java”, na linha 693.

Uma possível solução para este problema seria repartir este bloco de código em métodos separados, de forma que a informação não esteja todo concentrada neste bloco.

2º - Large Class

```
1      > /.../
19
20      package net.sf.freecol.common.model;
21
22      > import ...
62
63
64      > /** Represents all pieces that can be moved on the map-board. This includes: ...*/
1 heritor  ± Mike Pope +19
71      public class Unit extends GoodsLocation
72          implements Consumer, Locatable, Movable, Nameable, Ownable {
73
74          5 usages
75          private static final Logger logger = Logger.getLogger(Unit.class.getName());
76
77          1 usage  ± Mike Pope
78          > private static class ClosestSettlementGoalDecider implements GoalDecider {...};
140
141          /** Class index for units. */
142          1 usage
143          private static final int UNIT_CLASS_INDEX = 40;
144
145          public static final String TAG = "unit";
```

Esta classe tem à volta de 5000 linhas, que se torna bastante confuso de analisar, sendo muito fácil perdermo-nos aqui no meio.

A sua localização é “src/net/sf/freecol/common/model/Unit.java”.

Uma possível solução seria tornar esta classe numa classe abstrata, de forma a poder subdividir os vários métodos para tipos de unidades diferentes em diferentes subclasses.

3º - Data Class

```
package net.sf.freecol.client.gui.dialog;

± Paolo Bizzarri +1

public class Parameters {

    public final int distToLandFromHighSeas;

    public final int maxDistanceToEdge;

    1 usage ± Paolo Bizzarri
    Parameters(int distToLandFromHighSeas, int maxDistanceToEdge) {
        this.distToLandFromHighSeas = distToLandFromHighSeas;
        this.maxDistanceToEdge = maxDistanceToEdge;
    }
}
```

Este excerto de código consiste numa classe inteira, que apenas está a definir certos parâmetros, logo é uma “Data Class”, pois esta não faz nada de relevante para além de guardar valores (e poucos).

A sua localização é “src/net/sf/freecol/client/gui/dialog/Parameters.java”.

Uma possível solução seria definir estes parâmetros como variáveis globais da classe onde são usadas, visto que também são usadas em poucas classes (<5).