

Commit

extract class - 1, extract subclass - 1

Browse files

ComplexConditional#3 (#1) + GodClass#1 (#1) + MagicNumber#1 (#1) + MagicNumber#2 (#1) + MagicNumber#3 (#1) + MagicNumber#4 (#1)

 **matews-sousa** committed on Jun 21

1 parent [f0eaf15](#) commit [26f9b70](#)



Showing 4 changed files with 68 additions and 59 deletions.

Whitespace

Ignore whitespace



Split

Unified

2  .idea/misc.xml 

Load diff

Some generated files are not rendered by default. Learn more about [how customized files appear on GitHub](#).

67  sgab-core/src/sgab/model/dao/AcervoDAO.java 

1111import java.util.List;

1212import java.util.Map;

1313import sgab.model.dto.Exemplar;

1414+ import sgab.model.dto.util.AcervoHelper;

1415import sgab.model.dto.util.ExemplarStatus;

1516import sgab.model.dto.util.ExemplarTipo;

1617import sgab.model.exception.PersistenciaException;

1718import java.util.ArrayList;

1819import sgab.model.dto.Autor;

1920import sgab.model.dto.Biblioteca;

2021+ import sgab.util.Helpers;

2022

2123public class AcervoDAO implements GenericDAO<Exemplar, Long>{

2224private Map<Long, Exemplar> exemplares = new HashMap<>();

7577List<Exemplar> listExemplar = new ArrayList<>();

7678

7779for (Exemplar exe: exemplares.values()) {

78- if (exemplarDisponivelParaConsulta(exe, biblioteca))

```

80 +         if (AcervoHelper.exemplarDisponivelParaConsulta(exe, biblioteca))
79 81             listExemplar.add(exe);
80 82     }
81 83
82 84     return listExemplar;
83 85 }
84 -
85 -     private boolean exemplarDisponivelParaConsulta(Exemplar exemplar, Biblioteca
biblioteca) {
86 -         boolean tipoConsulta = exemplar.getTipo() == ExemplarTipo.CONSULTA;
87 -         boolean ativo = exemplar.getStatus() != ExemplarStatus.DESATIVADA;
88 -         boolean pertenceBiblioteca = exemplar.getBibliotecaPosse() == biblioteca;
89 -
90 -         return tipoConsulta && ativo && pertenceBiblioteca;
91 -     }
92 86
93 87     public List<Exemplar> listarTransferencia(Biblioteca biblioteca) {
94 88         List<Exemplar> listExemplar = new ArrayList<>();
134 128     public List<Exemplar> listarAcervoParaReserva(){
135 129         List<Exemplar> resultados = new LinkedList<>();
136 130         for(Exemplar exemplar : exemplares.values()){
137 -             if(exemplarDisponivelParaReserva(exemplar)){
131 +             if(AcervoHelper.exemplarDisponivelParaReserva(exemplar)){
138 132                 resultados.add(exemplar);
139 133             }
140 134         }
144 138     public List<Exemplar> listarAcervoParaReserva(String titulo){
145 139         List<Exemplar> resultados = new LinkedList<>();
146 140         for(Exemplar exemplar : exemplares.values()){
147 -             if(checaSemelhanca(exemplar.getObra().getTitulo(), titulo) &&
exemplarDisponivelParaReserva(exemplar)){
141 +             if(Helpers.checaSemelhanca(exemplar.getObra().getTitulo(), titulo)
142 +                 && AcervoHelper.exemplarDisponivelParaReserva(exemplar)){
148 143                 resultados.add(exemplar);
149 144             }
150 145         }
154 149     public List<Exemplar> listarAcervoParaReserva(Autor autor){
155 150         List<Exemplar> resultados = new LinkedList<>();
156 151         for(Exemplar exemplar : exemplares.values()){
157 -             if(exemplar.getObra().getAutor().equals(autor) &&
exemplarDisponivelParaReserva(exemplar)) {
152 +             if(exemplar.getObra().getAutor().equals(autor)
153 +                 && AcervoHelper.exemplarDisponivelParaReserva(exemplar)) {
158 154                 resultados.add(exemplar);
159 155             }
160 156         }
164 160     public List<Exemplar> listarAcervoParaReserva(Biblioteca biblioteca){
165 161         List<Exemplar> resultados = new LinkedList<>();
166 162         for(Exemplar exemplar : exemplares.values()){
167 -             if(exemplar.getBibliotecaPosse().equals(biblioteca) &&
exemplarDisponivelParaReserva(exemplar)){

```

```

163 +         if(exemplar.getBibliotecaPosse().equals(biblioteca) &&
    AcervoHelper.exemplarDisponivelParaReserva(exemplar)){
168 164         resultados.add(exemplar);
169 165     }
170 166 }
171 167 return resultados;
172 168 }
173 -
174 - private boolean exemplarDisponivelParaReserva(Exemplar exemplar) {
175 -     ExemplarStatus status = exemplar.getStatus();
176 -     ExemplarTipo tipo = exemplar.getTipo();
177 -     boolean ativo = status != ExemplarStatus.DESATIVADA;
178 -     boolean reparo = status != ExemplarStatus.REPARO;
179 -     boolean transferencia = status != ExemplarStatus.TRANSFERENCIA;
180 -     boolean normal = tipo == ExemplarTipo.NORMAL;
181 -
182 -     return ativo && reparo && transferencia && normal;
183 - }
184 -
185 - private boolean checaSemelhanca(String nomeObra, String nomeInput){
186 -     String nomeA = nomeObra.toLowerCase();
187 -     String nomeB = nomeInput.toLowerCase();
188 -     float count = 0;
189 -
190 -     //igualar o tamanho das Strings
191 -     if(nomeA.length() > nomeB.length()){
192 -         for(int i = 0; i < (nomeObra.length() - nomeInput.length()); i++){
193 -             nomeB = nomeB.concat(" ");
194 -         }
195 -     }
196 -     if(nomeA.length() < nomeB.length()){
197 -         for(int i = 0; i < (nomeObra.length() - nomeInput.length()); i++){
198 -             nomeA = nomeA.concat(" ");
199 -         }
200 -     }
201 -
202 -     for(int i=0, j=0; i<nomeA.length(); i++, j++) {
203 -         if(nomeA.charAt(i) != nomeB.charAt(j)){
204 -             count++;
205 -             if(i+1 == nomeA.length())
206 -                 continue;
207 -             if(nomeA.charAt(i+1)==nomeB.charAt(j))
208 -                 j--;
209 -         }
210 -     }
211 -     float porcentagem = count / nomeA.length();
212 -
213 -     if(porcentagem < 0.21)
214 -         return true;
215 -
216 -     return false;
217 - }

```

218 169 }

21 sgab-core/src/sgab/model/dto/util/AcervoHelper.java

```
2      2
3      3      import java.util.LinkedList;
4      4      import java.util.List;
5      5      +
6      6      + import sgab.model.dto.Biblioteca;
7      7      import sgab.model.dto.Exemplar;
8      8
9      9      public class AcervoHelper {
21     23
22     24          return erros;
23     25      }
24     26      +
25     27      + public static boolean exemplarDisponivelParaReserva(Exemplar exemplar) {
26     28      +     ExemplarStatus status = exemplar.getStatus();
27     29      +     ExemplarTipo tipo = exemplar.getTipo();
28     30      +     boolean ativo = status != ExemplarStatus.DESATIVADA;
29     31      +     boolean reparo = status != ExemplarStatus.REPARO;
30     32      +     boolean transferencia = status != ExemplarStatus.TRANSFERENCIA;
31     33      +     boolean normal = tipo == ExemplarTipo.NORMAL;
32     34      +
33     35      +     return ativo && reparo && transferencia && normal;
34     36      +     }
35     37      +
36     38      + public static boolean exemplarDisponivelParaConsulta(Exemplar exemplar,
37     39      +     Biblioteca biblioteca) {
38     40      +     boolean tipoConsulta = exemplar.getTipo() == ExemplarTipo.CONSULTA;
39     41      +     boolean ativo = exemplar.getStatus() != ExemplarStatus.DESATIVADA;
40     42      +     boolean pertenceBiblioteca = exemplar.getBibliotecaPosse() == biblioteca;
41     43      +
42     44      +     return tipoConsulta && ativo && pertenceBiblioteca;
43     45      +     }
24     46      }
```

37 sgab-core/src/sgab/util/Helpers.java

```
...     ...      @@ -0,0 +1,37 @@
1      1      + package sgab.util;
2      2      +
3      3      + public class Helpers {
4      4      +     public static boolean checaSemelhanca(String nomeObra, String nomeInput){
5      5      +         String nomeA = nomeObra.toLowerCase();
6      6      +         String nomeB = nomeInput.toLowerCase();
7      7      +         float count = 0;
8      8      +
9      9      +         //igualar o tamanho das Strings
10     10      +         if(nomeA.length() > nomeB.length()){
11     11      +             for(int i =0; i < (nomeObra.length() - nomeInput.length()); i++){
```

```
12 +         nomeB = nomeB.concat(" ");
13 +     }
14 + }
15 + if(nomeA.length() < nomeB.length()){
16 +     for(int i =0; i < (nomeObra.length() - nomeInput.length()); i++){
17 +         nomeA = nomeA.concat(" ");
18 +     }
19 + }
20 +
21 + for(int i=0, j=0; i<nomeA.length(); i++, j++) {
22 +     if(nomeA.charAt(i) != nomeB.charAt(j)){
23 +         count++;
24 +         if(i+1 == nomeA.length())
25 +             continue;
26 +         if(nomeA.charAt(i+1)==nomeB.charAt(j))
27 +             j--;
28 +     }
29 + }
30 + float porcentagem = count / nomeA.length();
31 +
32 + if(porcentagem < 0.21)
33 +     return true;
34 +
35 + return false;
36 + }
37 + }
```

0 comments on commit 26f9b70

Please [sign in](#) to comment.