

Commit

 This commit does not belong to any branch on this repository, and may belong to a fork outside of the repository.

problemas:

Browse files

- * Data class 2
- * magic number 4
- * long parameters list 6
- * god class 1



cmatheusIA committed on Jun 21

1 parent [f27d410](#) commit [d1971e6](#)

Showing 5 changed files with 125 additions and 222 deletions.

Whitespace

Ignore whitespace

Split

Unified

38 src/main/java/br/sistemalojaroupas/model/entities/Brand.java

```
...      ...      @@ -1,8 +1,3 @@
1          - /*
2          - * To change this license header, choose License Headers in Project Properties.
3          - * To change this template file, choose Tools | Templates
4          - * and open the template in the editor.
5          - */
6      1      package br.sistemalojaroupas.model.entities;
7      2
8      3      import java.util.Objects;
12     7      import org.dizitart.no2.objects.Index;
13     8      import org.dizitart.no2.objects.Indices;
14     9
15          - /**
16          - *
17          - * @author silas
18          - */
19     10     @Indices(
20          -     @Index(value = "name", type = IndexType.Unique)
21     11     +     @Index(value = "name", type = IndexType.Unique)
21     12     )
```

```
22 13 public class Brand {
23 14 -
14 15 +
24 15 @Id
25 16 private NitriteId id;
26 17 private String name;
27 18 -
18 19 +
28 19 public Brand() {
29 20 -
20 21 +
30 21 }
31 22 -
22 23 +
32 23 public Brand(String name) {
33 24     this.name = name;
34 25 }
47 38
48 39 @Override
49 40 public int hashCode() {
50 41     int hash = 7;
51 42     hash = 29 * hash + Objects.hashCode(this.id);
52 43     return hash;
41 44 +
42 45     return Objects.hash(id);
53 46 }
54 47
55 48 @Override
56 49 public boolean equals(Object obj) {
57 50     if (this == obj) {
58 51         return true;
59 52     }
60 53     if (obj == null) {
61 54         return false;
62 55     }
63 56     if (getClass() != obj.getClass()) {
64 57         return false;
65 58     }
66 59     final Brand other = (Brand) obj;
67 60     if (!Objects.equals(this.id, other.id)) {
49 61 +
50 62     if (obj == null || getClass() != obj.getClass()) {
68 63         return false;
69 64     }
70 65     return true;
52 66 +
53 67     Brand other = (Brand) obj;
54 68 +
55 69     return Objects.equals(id, other.id);
71 70 }
72 71
73 72 -
74 73 -
75 74 @Override
76 75 public String toString() {
77 76     return name;
```

```

78     59     }
79     -
80     60     }

```

60 src/main/java/br/sistemalojaroupas/model/entities/Employee.java

```

...     @@ -1,8 +1,3 @@
1         - /*
2         -  * To change this license header, choose License Headers in Project Properties.
3         -  * To change this template file, choose Tools | Templates
4         -  * and open the template in the editor.
5         -  */
6     1     package br.sistemalojaroupas.model.entities;
7     2
8     3     import java.io.Serializable;
15    10    import org.dizitart.no2.objects.Index;
16    11    import org.dizitart.no2.objects.Indices;
17    12
18         - /**
19         -  *
20         -  * @author silas
21         -  */
22    13    @Indices(
23         -     @Index(value = "cpf", type = IndexType.Unique)
14         +     @Index(value = "cpf", type = IndexType.Unique)
24    15    )
25    16    public class Employee implements Serializable, TableContract {
26         -
27         -     @Id
17         +
18         +     @Id
28    19    private NitriteId id;
29    20    private String cpf;
30    21    private String name;
33    24    private Date birthDate;
34    25    private Date admissionDate;
35    26    private Double salary;
36         -
27         +
37    28    private Address address;
38    29    private Office office;
39         -
30         +
40    31    private final static SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
41         -
32         +
42    33    public Employee() {
43    34    }
44    35
56    47    public NitriteId getId() {
57    48    return id;

```

```

58      49      }
59      -
      50      +
60      51      public String getCpf() {
61      52          return cpf;
62      53      }
92      83      public Date getBirthDate() {
93      84          return birthDate;
94      85      }
95      -
      86      +
96      87      public String getFormattedBirthDate() {
97      88          return sdf.format(birthDate);
98      89      }
104     95      public Date getAdmissionDate() {
105     96          return admissionDate;
106     97      }
107     -
      98     +
108     99      public String getFormattedAdmissionDate() {
109    100          return sdf.format(admissionDate);
110    101      }
139    130
140    131      @Override
141    132      public int hashCode() {
142        -          int hash = 5;
143        -          hash = 83 * hash + Objects.hashCode(this.id);
144        -          return hash;
      133      +          return Objects.hash(id);
145    134      }
146    135
147    136      @Override
148    137      public boolean equals(Object obj) {
149    138          if (this == obj) {
150    139              return true;
151    140          }
152        -          if (obj == null) {
153        -              return false;
154        -          }
155        -          if (getClass() != obj.getClass()) {
      141      +          if (obj == null || getClass() != obj.getClass()) {
156    142              return false;
157    143          }
158        -          final Employee other = (Employee) obj;
159        -          if (!Objects.equals(this.id, other.id)) {
160        -              return false;
161        -          }
162        -          return true;
      144      +          Employee other = (Employee) obj;
      145      +          return Objects.equals(id, other.id);
163    146      }
164    147

```

```

165 148     @Override
166 149     public String toString() {
167     -         return "Employee{" + "cpf=" + cpf + ", name=" + name + ", email=" + email +
        ", phone=" + phone + ", birthDate=" + birthDate + ", admissionDate=" +
        admissionDate + ", salary=" + salary + ", address=" + address + '}';
        150 +         return "Employee{" +
        151 +             "cpf='" + cpf + '\'' +
        152 +             ", name='" + name + '\'' +
        153 +             ", email='" + email + '\'' +
        154 +             ", phone='" + phone + '\'' +
        155 +             ", birthDate=" + birthDate +
        156 +             ", admissionDate=" + admissionDate +
        157 +             ", salary=" + salary +
        158 +             ", address=" + address +
        159 +             '}'';
168 160     }
169 161
170 162     @Override
171 163     public Object[] tableRowModel() {
172 164         return new Object[] {
173     -             getCpf(),
174     -             getName()
        165 +             cpf,
        166 +             name
175 167         };
176 168     }
177     -
        169 +
178 170 }

```

103 src/main/java/br/sistemalojaroupas/model/entities/Product.java

```

...     @@ -1,8 +1,3 @@
1     - /*
2     -  * To change this license header, choose License Headers in Project Properties.
3     -  * To change this template file, choose Tools | Templates
4     -  * and open the template in the editor.
5     -  */
6     1     package br.sistemalojaroupas.model.entities;
7     2
8     3     import java.io.Serializable;
11    6     import java.util.Objects;
12    7     import org.dizitart.no2.objects.Id;
13    8
14    - /**
15    -  *
16    -  * @author silas
17    -  */
18    9     public class Product implements Serializable, TableContract {
19    10         private final static long serialVersionUID = 1L;
20    -

```

```

11 +
21 12 private static SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
22 -
13 +
23 14 @Id
24 15 private Long id;
25 16 private String description;
28 19 private Double salePrice;
29 20 private Date date = new Date();
30 21 private int quantity;
31 -
22 +
32 23 private Color color;
33 24 private Category category;
34 25 private Brand brand;
35 -
26 +
36 27 public Product() {}
37 28
38 29 public Product(String description, Category category, String size, Color color,
Double costPrice, Double salePrice, int quantity) {
104 95 public Date getDate() {
105 96 return date;
106 97 }
107 -
98 +
108 99 public String getFormattedDate() {
109 - String formattedDate = sdf.format(this.date);
110 - return formattedDate;
100 + return sdf.format(this.date);
111 101 }
112 102
113 103 public void setDate(Date date) {
129 119 public void setBrand(Brand brand) {
130 120 this.brand = brand;
131 121 }
132 -
122 +
133 123 @Override
134 124 public int hashCode() {
135 - int hash = 7;
136 - hash = 67 * hash + Objects.hashCode(this.id);
137 - return hash;
125 + return Objects.hash(id);
138 126 }
139 127
140 128 @Override
141 129 public boolean equals(Object obj) {
142 130 if (this == obj) {
143 131 return true;
144 132 }
145 - if (obj == null) {

```

```

146     -         return false;
147     -     }
148     -     if (getClass() != obj.getClass()) {
149     -         return false;
150     -     }
151     -     final Product other = (Product) obj;
152     -     if (!Objects.equals(this.id, other.id)) {
153         133 +         if (obj == null || getClass() != obj.getClass()) {
154         134 +             return false;
155         135 +         }
156     -         return true;
157         136 +         Product other = (Product) obj;
158         137 +         return Objects.equals(id, other.id);
159     -     }
160     -
161     -
162     -
163     -
164     -
165     -
166     -
167     -
168     -
169     -
170         142 +         return "Product{"
171         143 +             + "id="
172         144 +             + id
173         145 +             + ", description="
174         146 +             + description
175         147 +             + ", category="
176         148 +             + category
177         149 +             + ", size="
178         150 +             + size
179         151 +             + ", brand="
180         152 +             + brand
181         153 +             + ", color="
182         154 +             + color
183         155 +             + ", costPrice="
184         156 +             + costPrice

```

```

157 +         + ", salePrice="
158 +         + salePrice
159 +         + ", date="
160 +         + date
161 +         + ", quantity="
162 +         + quantity
181 163         + '}';
182 164     }
183 165
184 166     @Override
185 167     public Object[] tableRowModel() {
186 168         return new Object[] {
187 -         getId(),
188 -         getDescription(),
189 -         getColor(),
190 -         getSize(),
191 -         getCategory(),
192 -         getBrand(),
193 -         String.format("%.2f", getCostPrice()),
194 -         String.format("%.2f", getSalePrice()),
195 -         getQuantity(),
196 -         getFormattedDate()
169 +         id,
170 +         description,
171 +         color,
172 +         size,
173 +         category,
174 +         brand,
175 +         String.format("%.2f", costPrice),
176 +         String.format("%.2f", salePrice),
177 +         quantity,
178 +         getFormattedDate()
197 179     };
198 180 }
199 -
200 - }
    - }
181 + }

```

138 src/main/java/br/sistemalojaroupas/model/entities/Sale.java

```

...    ...    @@ -1,8 +1,3 @@
1      - /*
2      -  * To change this license header, choose License Headers in Project Properties.
3      -  * To change this template file, choose Tools | Templates
4      -  * and open the template in the editor.
5      -  */
6      1      package br.sistemalojaroupas.model.entities;
7      2
8      3      import java.io.Serializable;
13     8      import java.util.Set;
14     9      import org.dizitart.no2.objects.Id;

```



```

15      10
16      - /**
17      -  *
18      -  * @author silas
19      -  */
20      11      public class Sale implements Serializable, TableContract {
21      12      -      private final static long serialVersionUID = 1L;
22      13      -
23      14      -      private static SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy
24      15      HH:mm:ss");
25      16      +      private static final long serialVersionUID = 1L;
26      17      +      private static final SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy
27      18      HH:mm:ss");
28      19      +
29      20      @Id
30      21      private Long id;
31      22      -
32      23      +
33      24      private Set<SaleItem> items = new HashSet<>();
34      25      -
35      26      private Employee employee;
36      27      private Customer customer;
37      28      private Integer payment;
38      29      private Integer installments;
39      30      -
40      31      public static final int MONEY = 1;
41      32      public static final int CREDIT = 2;
42      33      public static final int DEBIT = 3;
43      34      public static final int MONEY_CREDIT = 4;
44      35      -
45      36      private Date moment;
46      37      -
47      38      +
48      39      public static class PaymentType {
49      40      +      public static final int MONEY = 1;
50      41      +      public static final int CREDIT = 2;
51      42      +      public static final int DEBIT = 3;
52      43      +      public static final int MONEY_CREDIT = 4;
53      44      +      }
54      45      +
55      46      public Sale() {
56      47      -
57      48      +
58      49      }
59      50
60      51      public Long getId() {
61      52      public Set<SaleItem> getItems() {
62      53      return items;
63      54      }
64      55      -
65      56      +

```

```
58 48      public Date getMoment() {
59 49          return moment;
60 50      }
61 -
62 51 +
63 52      public void setMoment(Date moment) {
64 53          this.moment = moment;
65 54      }
66 -
67 55 +
68 56      public void addItem(SaleItem item) {
69 57          items.add(item);
70 58      }
71 -
72 59 +
73 60      public void removeItem(SaleItem item) {
74 61          items.remove(item);
75 62      }
76 63
77 64      public Employee getEmployee() {
78 65          return employee;
79 66      }
80 -
81 67 +
82 68      public String getEmployeeName() {
83 69          if (employee == null) {
84 70              return "N/A";
85 71          }
86 72          else {
87 73              return employee.getName();
88 74          }
89 75          return (employee == null) ? "N/A" : employee.getName();
90 76      }
91 77
92 78      public void setEmployee(Employee employee) {
93 79      public Customer getCustomer() {
94 80          return customer;
95 81      }
96 -
97 82 +
98 83      public String getCustomerName() {
99 84          if (customer == null) {
100 85              return "N/A";
101 86          }
102 87          else {
103 88              return customer.getName();
104 89          }
105 90          return (customer == null) ? "N/A" : customer.getName();
106 91      }
107 -
108 92 +
109 93      public String getCustomerCpf() {
```

```

105     -         if (customer == null) {
106     -             return "N/A";
107     -         }
108     -         else {
109     -             return customer.getCpf();
110     -         }
111     85 +         return (customer == null) ? "N/A" : customer.getCpf();
112     86     }
113     87
114     88     public void setCustomer(Customer customer) {
115     89         this.customer = customer;
116     90     }
117     91
118     92     public String getPaymentName() {
119     93         if (payment == null) {
120     94             throw new IllegalStateException("O método de pagamento não pode ser
nulo.");
121     95         }
122     96 -         if (payment == MONEY) {
123     96 +         if (payment == PaymentType.MONEY) {
124     97             return "DINHEIRO";
125     98         }
126     99 -         if (payment == CREDIT) {
127     99 +         if (payment == PaymentType.CREDIT) {
128     100             return "CRÉDITO";
129     101         }
130     102 -         if (payment == DEBIT) {
131     102 +         if (payment == PaymentType.DEBIT) {
132     103             return "DÉBITO";
133     104         }
134     105 -         if (payment == MONEY_CREDIT) {
135     105 +         if (payment == PaymentType.MONEY_CREDIT) {
136     106             return "DINHEIRO + CRÉDITO";
137     107         }
138     108         return "N/A";
139     112         return payment;
140     113     }
141     114
142     115
143     116
144     117
145     118
146     119
147     120
148     121
149     122
150     123
151     124
152     125
153     126
154     127
155     128
156     129
157     130
158     131
159     132
160     133
161     134
162     135
163     136
164     137
165     138
166     139
167     140
168     141
169     142
170     143
171     144
172     145
173     146
174     147
175     148
176     149
177     150
178     151
179     152
180     153
181     154
182     155
183     156
184     157
185     158
186     159
187     160
188     161
189     162
190     163
191     164
192     165
193     166
194     167
195     168
196     169
197     170
198     171
199     172
200     173
201     174
202     175
203     176
204     177
205     178
206     179
207     180
208     181
209     182
210     183
211     184
212     185
213     186
214     187
215     188
216     189
217     190
218     191
219     192
220     193
221     194
222     195
223     196
224     197
225     198
226     199
227     200
228     201
229     202
230     203
231     204
232     205
233     206
234     207
235     208
236     209
237     210
238     211
239     212
240     213
241     214
242     215
243     216
244     217
245     218
246     219
247     220
248     221
249     222
250     223
251     224
252     225
253     226
254     227
255     228
256     229
257     230
258     231
259     232
260     233
261     234
262     235
263     236
264     237
265     238
266     239
267     240
268     241
269     242
270     243
271     244
272     245
273     246
274     247
275     248
276     249
277     250
278     251
279     252
280     253
281     254
282     255
283     256
284     257
285     258
286     259
287     260
288     261
289     262
290     263
291     264
292     265
293     266
294     267
295     268
296     269
297     270
298     271
299     272
300     273
301     274
302     275
303     276
304     277
305     278
306     279
307     280
308     281
309     282
310     283
311     284
312     285
313     286
314     287
315     288
316     289
317     290
318     291
319     292
320     293
321     294
322     295
323     296
324     297
325     298
326     299
327     300
328     301
329     302
330     303
331     304
332     305
333     306
334     307
335     308
336     309
337     310
338     311
339     312
340     313
341     314
342     315
343     316
344     317
345     318
346     319
347     320
348     321
349     322
350     323
351     324
352     325
353     326
354     327
355     328
356     329
357     330
358     331
359     332
360     333
361     334
362     335
363     336
364     337
365     338
366     339
367     340
368     341
369     342
370     343
371     344
372     345
373     346
374     347
375     348
376     349
377     350
378     351
379     352
380     353
381     354
382     355
383     356
384     357
385     358
386     359
387     360
388     361
389     362
390     363
391     364
392     365
393     366
394     367
395     368
396     369
397     370
398     371
399     372
400     373
401     374
402     375
403     376
404     377
405     378
406     379
407     380
408     381
409     382
410     383
411     384
412     385
413     386
414     387
415     388
416     389
417     390
418     391
419     392
420     393
421     394
422     395
423     396
424     397
425     398
426     399
427     400
428     401
429     402
430     403
431     404
432     405
433     406
434     407
435     408
436     409
437     410
438     411
439     412
440     413
441     414
442     415
443     416
444     417
445     418
446     419
447     420
448     421
449     422
450     423
451     424
452     425
453     426
454     427
455     428
456     429
457     430
458     431
459     432
460     433
461     434
462     435
463     436
464     437
465     438
466     439
467     440
468     441
469     442
470     443
471     444
472     445
473     446
474     447
475     448
476     449
477     450
478     451
479     452
480     453
481     454
482     455
483     456
484     457
485     458
486     459
487     460
488     461
489     462
490     463
491     464
492     465
493     466
494     467
495     468
496     469
497     470
498     471
499     472
500     473
501     474
502     475
503     476
504     477
505     478
506     479
507     480
508     481
509     482
510     483
511     484
512     485
513     486
514     487
515     488
516     489
517     490
518     491
519     492
520     493
521     494
522     495
523     496
524     497
525     498
526     499
527     500
528     501
529     502
530     503
531     504
532     505
533     506
534     507
535     508
536     509
537     510
538     511
539     512
540     513
541     514
542     515
543     516
544     517
545     518
546     519
547     520
548     521
549     522
550     523
551     524
552     525
553     526
554     527
555     528
556     529
557     530
558     531
559     532
560     533
561     534
562     535
563     536
564     537
565     538
566     539
567     540
568     541
569     542
570     543
571     544
572     545
573     546
574     547
575     548
576     549
577     550
578     551
579     552
580     553
581     554
582     555
583     556
584     557
585     558
586     559
587     560
588     561
589     562
590     563
591     564
592     565
593     566
594     567
595     568
596     569
597     570
598     571
599     572
600     573
601     574
602     575
603     576
604     577
605     578
606     579
607     580
608     581
609     582
610     583
611     584
612     585
613     586
614     587
615     588
616     589
617     590
618     591
619     592
620     593
621     594
622     595
623     596
624     597
625     598
626     599
627     600
628     601
629     602
630     603
631     604
632     605
633     606
634     607
635     608
636     609
637     610
638     611
639     612
640     613
641     614
642     615
643     616
644     617
645     618
646     619
647     620
648     621
649     622
650     623
651     624
652     625
653     626
654     627
655     628
656     629
657     630
658     631
659     632
660     633
661     634
662     635
663     636
664     637
665     638
666     639
667     640
668     641
669     642
670     643
671     644
672     645
673     646
674     647
675     648
676     649
677     650
678     651
679     652
680     653
681     654
682     655
683     656
684     657
685     658
686     659
687     660
688     661
689     662
690     663
691     664
692     665
693     666
694     667
695     668
696     669
697     670
698     671
699     672
700     673
701     674
702     675
703     676
704     677
705     678
706     679
707     680
708     681
709     682
710     683
711     684
712     685
713     686
714     687
715     688
716     689
717     690
718     691
719     692
720     693
721     694
722     695
723     696
724     697
725     698
726     699
727     700
728     701
729     702
730     703
731     704
732     705
733     706
734     707
735     708
736     709
737     710
738     711
739     712
740     713
741     714
742     715
743     716
744     717
745     718
746     719
747     720
748     721
749     722
750     723
751     724
752     725
753     726
754     727
755     728
756     729
757     730
758     731
759     732
760     733
761     734
762     735
763     736
764     737
765     738
766     739
767     740
768     741
769     742
770     743
771     744
772     745
773     746
774     747
775     748
776     749
777     750
778     751
779     752
780     753
781     754
782     755
783     756
784     757
785     758
786     759
787     760
788     761
789     762
790     763
791     764
792     765
793     766
794     767
795     768
796     769
797     770
798     771
799     772
800     773
801     774
802     775
803     776
804     777
805     778
806     779
807     780
808     781
809     782
810     783
811     784
812     785
813     786
814     787
815     788
816     789
817     790
818     791
819     792
820     793
821     794
822     795
823     796
824     797
825     798
826     799
827     800
828     801
829     802
830     803
831     804
832     805
833     806
834     807
835     808
836     809
837     810
838     811
839     812
840     813
841     814
842     815
843     816
844     817
845     818
846     819
847     820
848     821
849     822
850     823
851     824
852     825
853     826
854     827
855     828
856     829
857     830
858     831
859     832
860     833
861     834
862     835
863     836
864     837
865     838
866     839
867     840
868     841
869     842
870     843
871     844
872     845
873     846
874     847
875     848
876     849
877     850
878     851
879     852
880     853
881     854
882     855
883     856
884     857
885     858
886     859
887     860
888     861
889     862
890     863
891     864
892     865
893     866
894     867
895     868
896     869
897     870
898     871
899     872
900     873
901     874
902     875
903     876
904     877
905     878
906     879
907     880
908     881
909     882
910     883
911     884
912     885
913     886
914     887
915     888
916     889
917     890
918     891
919     892
920     893
921     894
922     895
923     896
924     897
925     898
926     899
927     900
928     901
929     902
930     903
931     904
932     905
933     906
934     907
935     908
936     909
937     910
938     911
939     912
940     913
941     914
942     915
943     916
944     917
945     918
946     919
947     920
948     921
949     922
950     923
951     924
952     925
953     926
954     927
955     928
956     929
957     930
958     931
959     932
960     933
961     934
962     935
963     936
964     937
965     938
966     939
967     940
968     941
969     942
970     943
971     944
972     945
973     946
974     947
975     948
976     949
977     950
978     951
979     952
980     953
981     954
982     955
983     956
984     957
985     958
986     959
987     960
988     961
989     962
990     963
991     964
992     965
993     966
994     967
995     968
996     969
997     970
998     971
999     972
1000    973
1001    974
1002    975
1003    976
1004    977
1005    978
1006    979
1007    980
1008    981
1009    982
1010    983
1011    984
1012    985
1013    986
1014    987
1015    988
1016    989
1017    990
1018    991
1019    992
1020    993
1021    994
1022    995
1023    996
1024    997
1025    998
1026    999
1027    1000
1028    1001
1029    1002
1030    1003
1031    1004
1032    1005
1033    1006
1034    1007
1035    1008
1036    1009
1037    1010
1038    1011
1039    1012
1040    1013
1041    1014
1042    1015
1043    1016
1044    1017
1045    1018
1046    1019
1047    1020
1048    1021
1049    1022
1050    1023
1051    1024
1052    1025
1053    1026
1054    1027
1055    1028
1056    1029
1057    1030
1058    1031
1059    1032
1060    1033
1061    1034
1062    1035
1063    1036
1064    1037
1065    1038
1066    1039
1067    1040
1068    1041
1069    1042
1070    1043
1071    1044
1072    1045
1073    1046
1074    1047
1075    1048
1076    1049
1077    1050
1078    1051
1079    1052
1080    1053
1081    1054
1082    1055
1083    1056
1084    1057
1085    1058
1086    1059
1087    1060
1088    1061
1089    1062
1090    1063
1091    1064
1092    1065
1093    1066
1094    1067
1095    1068
1096    1069
1097    1070
1098    1071
1099    1072
1100    1073
1101    1074
1102    1075
1103    1076
1104    1077
1105    1078
1106    1079
1107    1080
1108    1081
1109    1082
1110    1083
1111    1084
1112    1085
1113    1086
1114    1087
1115    1088
1116    1089
1117    1090
1118    1091
1119    1092
1120    1093
1121    1094
1122    1095
1123    1096
1124    1097
1125    1098
1126    1099
1127    1100
1128    1101
1129    1102
1130    1103
1131    1104
1132    1105
1133    1106
1134    1107
1135    1108
1136    1109
1137    1110
1138    1111
1139    1112
1140    1113
1141    1114
1142    1115
1143    1116
1144    1117
1145    1118
1146    1119
1147    1120
1148    1121
1149    1122
1150    1123
1151    1124
1152    1125
1153    1126
1154    1127
1155    1128
1156    1129
1157    1130
1158    1131
1159    1132
1160    1133
1161    1134
1162    1135
1163    1136
1164    1137
1165    1138
1166    1139
1167    1140
1168    1141
1169    1142
1170    1143
1171    1144
1172    1145
1173    1146
1174    1147
1175    1148
1176    1149
1177    1150
1178    1151
1179    1152
1180    1153
1181    1154
1182    1155
1183    1156
1184    1157
1185    1158
1186    1159
1187    1160
1188    1161
1189    1162
1190    1163
1191    1164
1192    1165
1193    1166
1194    1167
1195    1168
1196    1169
1197    1170
1198    1171
1199    1172
1200    1173
1201    1174
1202    1175
1203    1176
1204    1177
1205    1178
1206    1179
1207    1180
1208    1181
1209    1182
1210    1183
1211    1184
1212    1185
1213    1186
1214    1187
1215    1188
1216    1189
1217    1190
1218    1191
1219    1192
1220    1193
1221    1194
1222    1195
1223    1196
1224    1197
1225    1198
1226    1199
1227    1200
1228    1201
1229    1202
1230    1203
1231    1204
1232    1205
1233    1206
1234    1207
1235    1208
1236    1209
1237    1210
1238    1211
1239    1212
1240    1213
1241    1214
1242    1215
1243    1216
1244    1217
1245    1218
1246    1219
1247    1220
1248    1221
1249    1222
1250    1223
1251    1224
1252    1225
1253    1226
1254    1227
1255    1228
1256    1229
1257    1230
1258    1231
1259    1232
1260    1233
1261    1234
1262    1235
1263    1236
1264    1237
1265    1238
1266    1239
1267    1240
1268    1241
1269    1242
1270    1243
1271    1244
1272    1245
1273    1246
1274    1247
1275    1248
1276    1249
1277    1250
1278    1251
1279    1252
1280    1253
1281    1254
1282    1255
1283    1256
1284    1257
1285    1258
1286    1259
1287    1260
1288    1261
1289    1262
1290    1263
1291    1264
1292    1265
1293    1266
1294    1267
1295    1268
1296    1269
1297    1270
1298    1271
1299    1272
1300    1273
1301    1274
1302    1275
1303    1276
1304    1277
1305    1278
1306    1279
1307    1280
1308    1281
1309    1282
1310    1283
1311    1284
1312    1285
1313    1286
1314    1287
1315    1288
1316    1289
1317    1290
1318    1291
1319    1292
1320    1293
1321    1294
1322    1295
1323    1296
1324    1297
1325    1298
1326    1299
1327    1300
1328    1301
1329    1302
1330    1303
1331    1304
1332    1305
1333    1306
1334    1307
1335    1308
1336    1309
1337    1310
1338    1311
1339    1312
1340    1313
1341    1314
1342    1315
1343    1316
1344    1317
1345    1318
1346    1319
1347    1320
1348    1321
1349    1322
1350    1323
1351    1324
1352    1325
1353    1326
1354    1327
1355    1328
1356    1329
1357    1330
1358    1331
1359    1332
1360    1333
1361    1334
1362    1335
1363    1336
1364    1337

```

```

153      117      throw new IllegalArgumentException("Opção inválida");
154      118      }
155      -
156      119      this.payment = payment;
157      -
158      120      }
159      121
160      -      public Object getInstallments() {
161      -          if (installments == null) {
162      -              return "N/A";
163      -          }
164      -          return installments;
165      122      +      public Object getFormattedInstallments() {
166      123      +          return (installments == null) ? "N/A" : installments;
167      124      }
168      125
169      126      public void setInstallments(Integer installments) {
170      127          if (payment == null) {
171      128              throw new IllegalStateException("O método de pagamento ainda não foi
172      129              definido.");
173      130          }
174      131          if (payment == Sale.CREDIT || payment == Sale.MONEY_CREDIT){
175      132          +      if (payment == PaymentType.CREDIT || payment == PaymentType.MONEY_CREDIT) {
176      133              this.installments = installments;
177      134          }
178      135      }
179      136      }
180      137      }
181      138      }
182      139      }
183      140      }
184      141      }
185      142      }
186      143      }
187      144      }
188      145      }
189      146      }
190      147      }
191      148      }
192      149      }
193      150      }
194      151      }
195      152      }
196      153      }
197      154      }
198      155      }

```

```

199      156      return true;
200      157      }
201      -      if (obj == null) {
202      -          return false;
203      -      }
204      -      if (getClass() != obj.getClass()) {
205      -          return false;
206      -      }
207      -      final Sale other = (Sale) obj;
208      -      if (!Objects.equals(this.id, other.id)) {
209      158      +      if (obj == null || getClass() != obj.getClass()) {
210      159      +          return false;
211      160      +      }
212      -      return true;
213      161      +      Sale other = (Sale) obj;
214      162      +      return Objects.equals(id, other.id);
215      163      }
216      164
217      165      @Override
218      166      public String toString() {
219      -      return "Sale{" + "id=" + id + ", items=" + items + ", moment=" + moment +
220      -      "total=" + getTotal() + '}';
221      167      +      return "Sale{" + "id=" + id + ", items=" + items + ", moment=" + moment +
222      -      ", total=" + getTotal() + '}';
223      168      }
224      169
225      170      @Override
226      171      public Object[] tableRowModel() {
227      172      return new Object[]{
228      -      getId(),
229      -      String.format("%.2f", getTotal()),
230      -      getFormattedDate(),
231      -      (customer == null) ? "N/A" : getCustomer().getName(),
232      -      (customer == null) ? "N/A" : getCustomer().getCpf()
233      173      +      getId(),
234      174      +      String.format("%.2f", getTotal()),
235      175      +      getFormattedDate(),
236      176      +      (customer == null) ? "N/A" : getCustomer().getName(),
237      177      +      (customer == null) ? "N/A" : getCustomer().getCpf()
238      178      };
239      179      }
240      -
241      180      }

```

8 src/main/java/br/sistemalojaroupas/view/sale/Payment.java

```

144      144      }// </editor-fold>//GEN-END:initComponents
145      145
146      146      private void btMoneyActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
147      FIRST:event_btMoneyActionPerformed
148      -      saleWindow.getSale().setPayment(Sale.MONEY);
149      147      +      saleWindow.getSale().setPayment(Sale.PaymentType.MONEY);

```

```
148 148
149 149     this.dispose();
150 150     new Payment_Money(this, true).setVisible(true);
155 155 }//GEN-LAST:event_jLabel3MouseClicked
156 156
157 157     private void btDebitActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_btDebitActionPerformed
158 158 -     saleWindow.getSale().setPayment(Sale.DEBIT);
158 158 +     saleWindow.getSale().setPayment(Sale.PaymentType.DEBIT);
159 159
160 160     this.dispose();
161 161     new Payment_Debit(this, true).setVisible(true);
162 162 }//GEN-LAST:event_btDebitActionPerformed
163 163
164 164     private void btCreditActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_btCreditActionPerformed
165 165 -     saleWindow.getSale().setPayment(Sale.CREDIT);
165 165 +     saleWindow.getSale().setPayment(Sale.PaymentType.CREDIT);
166 166
167 167     this.dispose();
168 168     new Payment_Credit(this, true).setVisible(true);
169 169 }//GEN-LAST:event_btCreditActionPerformed
170 170
171 171     private void btCreditAndMoneyActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_btCreditAndMoneyActionPerformed
172 172 -     saleWindow.getSale().setPayment(Sale.MONEY_CREDIT);
172 172 +     saleWindow.getSale().setPayment(Sale.PaymentType.MONEY_CREDIT);
173 173
174 174     this.dispose();
175 175     new Payment_MoneyAndCredit(this, true).setVisible(true);
```

0 comments on commit d1971e6

Please [sign in](#) to comment.