

Code Smells

Switch statement (VirtualFlow.java) - Miguel Agostinho 60677

```
switch(event.getTextDeltaXUnits()) {  
    case CHARACTERS:  
        // can we get character size here?  
        // for now, fall through to pixel values  
    case NONE:  
        double dx = event.getDeltaX();  
        double dy = event.getDeltaY();  
  
        virtualDelta = (Math.abs(dx) > Math.abs(dy) ? dx : dy);  
    }  
}
```

O switch apresentado abaixo para além de no case “CHARACTERS” não se fazer nada podendo ser um “Dead Code”, poderia ser feito apenas com um if/else evitando assim o uso desnecessário de um switch statement.

Reviews

Daniel Eugénio 59797

Review do code smell nº1 (Switch Statement), é bem identificado, o case Characters não tem qualquer bloco de código, resultando num switch com apenas um caso que, como foi bem evidenciado pelo colega.

Duplicated Code (VirtualFlow.java) - Miguel Agostinho 60677

```
ScrollBar nonVirtualBar = isVertical() ? hbar : vbar;
if (needBreadthBar) {
    double nonVirtualDelta = isVertical() ? event.getDeltaX() : event.getDeltaY();
    if (nonVirtualDelta != 0.0) {
        double newValue = nonVirtualBar.getValue() - nonVirtualDelta;
        if (newValue < nonVirtualBar.getMin()) {
            nonVirtualBar.setValue(nonVirtualBar.getMin());
        } else if (newValue > nonVirtualBar.getMax()) {
            nonVirtualBar.setValue(nonVirtualBar.getMax());
        } else {
            nonVirtualBar.setValue(newValue);
        }
        event.consume();
    }
}
};

});
```

Neste caso podemos observar que chamamos duas vezes o método nonVirtualBar.getMin() e nonVirtualBar.getMax() o que deveríamos fazer era criar duas variáveis locais, uma que guardava o valor retornado por nonVirtualBar.getMin() e a outra que guarda o valor de nonVirtualBar.getMax() e assim evitamos chamar o mesmo método várias vezes.

Reviews

Guilherme Abrantes 60971

Review do code smell Duplicated Code , aqui temos code smell muito bem encontrado, esta duplicação de código poderia passar facilmente despercebida, porém são estas diferenças que tornam tudo mais legível, a solução de criar as variáveis locais acho que seria o ideal.

Comments (ChartUIConfiguration.java) - Miguel Agostinho 60677

```
└─ dbarashev +2
  public class ChartUIConfiguration {  
  
  └─ dbarashev +1
    public ChartUIConfiguration(UIConfiguration projectionConfig) {
      mySpanningRowTextFont = Fonts.TOP_UNIT_FONT;
      mySpanningHeaderBackgroundColor = new Color(0.95f, 0.93f, 0.95f);
      myHeaderBorderColor = new Color(0.482f, 0.482f, 0.482f);
      myWorkingTimeBackgroundColor = Color.WHITE;
      myNoDayTimeBackgroundColor = new Color(0.9f, 0.9f, 0.9f);
      myPublicHolidayTimeBackgroundColor = new Color(240, 220, 240);
      // myHeaderBorderColor = new Color(0, 1f, 0f);
      myBottomUnitGridColor = new Color(0.482f, 0.482f, 0.482f);
      myProjectConfig = projectionConfig;
      myChartStylesOption = new ChartPropertiesOption();
    }  
  
  └─ dbarashev
    ListOption<Map.Entry<String, String>> getChartStylesOption() { return myChartStylesOption; }
  └─ dbarashev
    Font getSpanningHeaderFont() { return mySpanningRowTextFont; }  
  
  └─ usage └─ dbarashev
    public int getHeaderHeight() { return myHeaderHeight; }  
  
  └─ usage └─ dbarashev
    public void setHeaderHeight(int headerHeight) { myHeaderHeight = headerHeight; }  
  
  └─ dbarashev
    public int getSpanningHeaderHeight() { return myHeaderHeight / 2; }  
  
  └─ dbarashev
    public Color getSpanningHeaderBackgroundColor() { return mySpanningHeaderBackgroundColor; }  
  
  └─ dbarashev
    public Color getHeaderBorderColor() { return myHeaderBorderColor; }
```

Neste caso a classe ChartUIConfiguration têm uma ausência de comentários muito notória e os comentários nas classes são muito importantes porque se outra pessoa for programar a mesma classe precisa de saber o que os métodos ou até a classe fazem, portanto deveríamos comentar esta.

Reviews

Francisco Silveira 60816

Review code smell nº3 :

Concordo com o que o colega disse. A falta de comentários torna o código pouco legível e torna o trabalho de quem vai trabalhar nele muito mais difícil. Os comentários são uma parte essencial do código.

Repeated Code - Daniel Eugénio 59797

```
234     getViewManager().createView(myGanttChartTabContent, new ImageIcon(getClass().getResource("/icons/tasks_16.gif")));
235     getViewManager().toggleVisible(myGanttChartTabContent);
236
237     myResourceChartTabContent = new ResourceChartTabContentPanel(getProject(), getUIFacade(), getResourcePanel(),
238         getResourcePanel().area);
239     getViewManager().createView(myResourceChartTabContent, new ImageIcon(getClass().getResource("/icons/res_16.gif")));
240     getViewManager().toggleVisible(myResourceChartTabContent);
```

- Neste trecho, é possível reparar que as linhas 234-235 e 239-240 são muito semelhantes, diferindo apenas no último argumento passado no método `toggleVisible()`.
- Uma solução possível seria fazer um método que receberia o argumento que queremos passar no `toggleVisible()`. Esta simples mudança tornaria o código nesta região mais limpo e legível.

-----> (GanttProject.java) <-----

Reviews

Miguel Agostinho 60677

Review do code smell nº1, concordo com o meu colega pois a repetição de código torna-o mais difícil de ler e entender e a solução que forneceu também me parece boa.

Long Parameter List - Daniel Eugénio 59797

```
140  
147     private void constructTopOffsets(TimeUnit timeUnit, List<Offset> topOffsets, List<Offset> bottomOffsets,  
148         int initialEnd, int baseUnitWidth) {
```

- Neste exemplo, os argumentos topOffsets e bottomOffsets podem ser atributos numa classe (Offsets.java, por exemplo) e, nesse cenário, estes dois argumentos podiam ser 1 apenas.

-----> (OffsetBuilderImpl.java) <-----

Reviews

Rafael Costa 60441

Review do long parameter list.

Concordo com o referido, é um bad smell code que devemos evitar fazer, pois torna o código confuso e denso.

Divergent Change - Daniel Eugénio 59797

```
85
86     Box colorLabels = Box.createHorizontalBox();
87     for (final Color c : myRecentColors) {
88         final JXLabel label = new JXLabel();
89         label.setBackgroundPainter(new Painter<JXLabel>() {
90             @Override
91             public void paint(Graphics2D g, JXLabel object, int width, int height) {
92                 g.setColor(c);
93                 g.fillRect(4, 4, width-8, height-8);
94             }
95         });
96         label.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
97         label.setFocusable(true);
98         label.setPreferredSize(new Dimension(20, 20));
99         label.setMaximumSize(new Dimension(20, 20));
100
101        final Border outsideFocusBorder = BorderFactory.createLineBorder(c.darker(), 2);
102        final Border outsideNoFocusBorder = BorderFactory.createEmptyBorder(2, 2, 2, 2);
103        label.setBorder(outsideNoFocusBorder);
104        label.addFocusListener(new FocusAdapter() {
105            @Override
106            public void focusGained(FocusEvent e) {
107                label.setBorder(outsideFocusBorder);
108                myChooserImpl.setColor(c);
109                mySelectedColor = c;
110            }
111            @Override
112            public void focusLost(FocusEvent e) { label.setBorder(outsideNoFocusBorder); }
113        });
114        label.addMouseListener(new MouseAdapter() {
115            @Override
116            public void mouseClicked(MouseEvent e) { label.requestFocus(); }
117        });
118        colorLabels.add(label);
119        colorLabels.add(Box.createHorizontalStrut(5));
120    }
121
122    colorLabels.setBorder(BorderFactory.createEmptyBorder(0, 7, 0, 0));
```

- Aqui é possível ver a completa ausência do uso de constantes, o que nos leva a ter de mexer em várias zonas do código na mesma classe. A utilização dos chamados *magic numbers* também dificulta a legibilidade do código por parte de pessoas que não estiveram envolvidas inicialmente no seu desenvolvimento.
- A solução seria utilizar constantes com nomes explicativos para podermos fatorizar várias zonas do código ao mesmo tempo.

-----> (GPCColorChooser.java) <-----

Reviews

Guilherme Abrantes 60971

Review do code smell Divergent Change, acho que foi um code smell bem encontrado sendo difícil a alteração do código pois vamos ter de alterar todos os números e também dificulta a legibilidade, a solução encontrada acho que é perfeita pois resolveria estes dois problemas mesmo que estes números só tenham sido usados neste método variáveis locais com nomes explicativos tornaria o código muito mais claro.

Speculative Generality - Rafael Costa 60441

```
ganttpj project / ganttpj project / src / main / java / net / sourceforge / ganttpj / PreferenceServiceImpl.java
```

```
1 package net.sourceforge.ganttpj;
2
3 import ...
4
5 public class PreferenceServiceImpl implements IPreferencesService {
6
7     @Override
8     public IStatus applyPreferences(IExportedPreferences preferences) throws CoreException {
9         // TODO Auto-generated method stub
10        return null;
11    }
12
13    @Override
14    public void applyPreferences(IEclipsePreferences node, IPreferenceFilter[] filters) throws CoreException {
15        // TODO Auto-generated method stub
16    }
17
18    @Override
19    public IStatus exportPreferences(IEclipsePreferences node, OutputStream output, String[] excludesList)
20        throws CoreException {
21        // TODO Auto-generated method stub
22        return null;
23    }
24
25    @Override
26    public void exportPreferences(IEclipsePreferences node, IPreferenceFilter[] filters, OutputStream output)
27        throws CoreException {
28        // TODO Auto-generated method stub
29    }
30
31    @Override
32    public String get(String key, String defaultValue, Preferences[] nodes) {
33        // TODO Auto-generated method stub
34        return null;
35    }
36}
```

Kotlin code style
Do you want to update your Kotlin code style...
Apply the code style Don't show again

É constituído por código que é genérico ou abstrato e o mais importante, não é realmente necessário hoje. Esse código está lá para apoiar o comportamento futuro, que pode ou não, ser necessário no futuro.

A solução será apenas criar os métodos e parâmetros à medida que é necessário.

<PreferencesServiceImpl.java>

Reviews

Guilherme Abrantes 60971

Review do code smell Speculative Generality, este code smell foi bem encontrado a explicação está bastante boa tenho apenas a acrescentar que além disto não existe qualquer tipo de comentário a explicar o que aquela classe poderia acrescentar no futuro, dificultando ainda mais o código.

Not Comment - Rafael Costa 60441

The screenshot shows an IDE interface with a dark theme. The left sidebar displays a project structure for a 'ganttpoint' project, listing various Java files and their contents. The main editor window shows the 'ErrorNotifier.java' file, which contains Java code for a class named 'ErrorNotifier'. The code includes imports for ArrayList and List, a constructor taking a UIFacadeImpl parameter, a method 'add' that adds an error message to a list, and a run method that iterates through the errors, appends them to a buffer, and shows an error dialog. A tooltip at the bottom right suggests updating Kotlin code style. The bottom navigation bar includes tabs for Git, TODO, Problems, Terminal, Profiler, Build, and Endpoints.

```
ganttpoint ganttpoint src main java net.sourceforge.ganttpoint ErrorNotifier
Project ion.java RsUpdate.java MultiDatePicker.kt CalendarEditorPanel.java BackgroundRendererImpl.java DesktopIntegration.java ErrorNotifier.java
File Edit View Insert Tools Help
1
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
311
312
313
314
315
315
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552

```

A classe apresenta-se sem comentários, o que é algo essencial para a interpretação da mesma.

Os comentários são essenciais uma vez que ao passar o código a outra pessoa, ou a trabalhar em equipa vai facilitar a interpretação do código pelos mesmos

<ErrorNotifier.java>

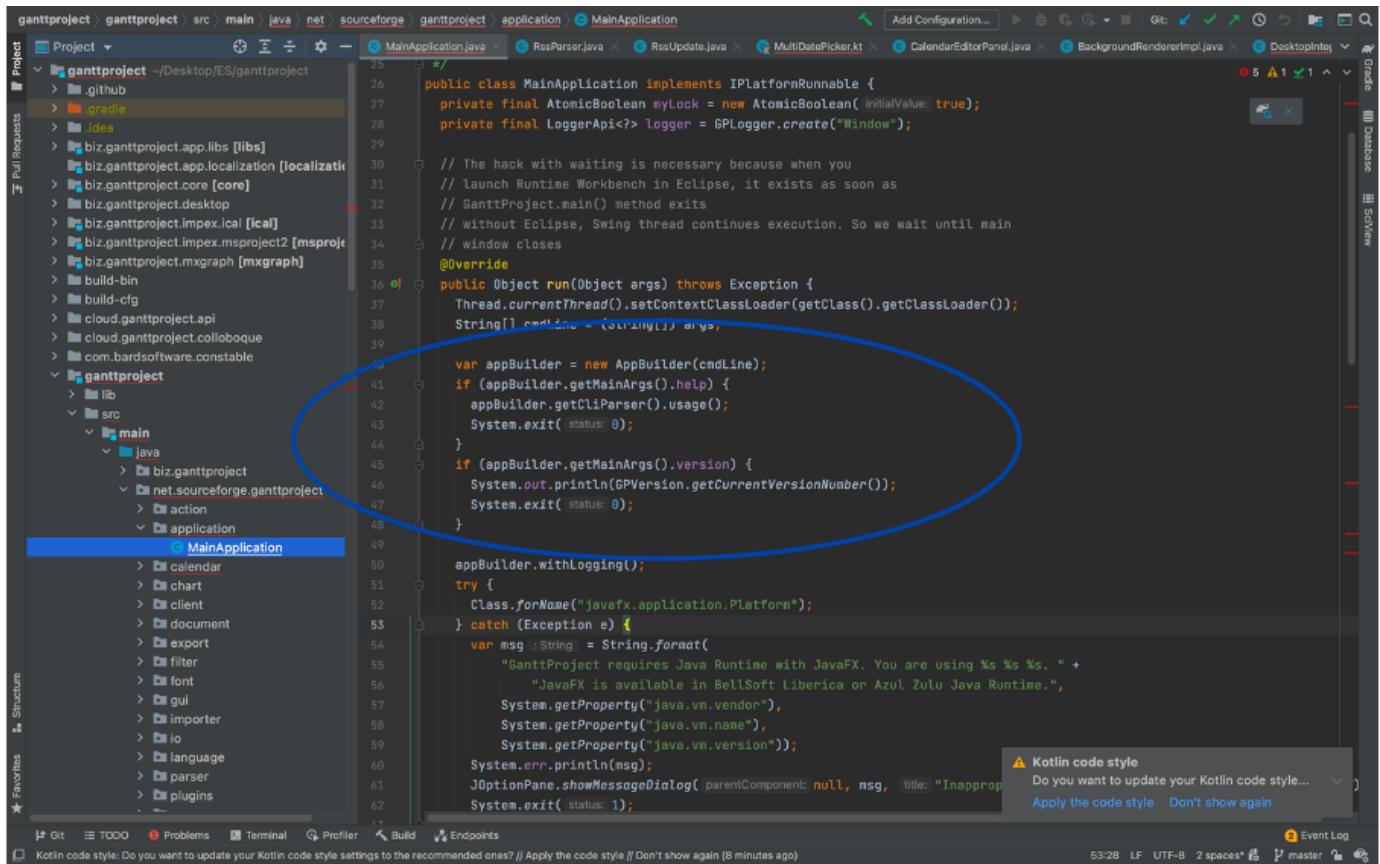
Reviews

Francisco Silveira 60816

Review Not Commented:

Concordo com o que o colega disse. A falta de comentários torna o código pouco legível e torna o trabalho de quem vai trabalhar nele muito mais difícil. Os comentários são uma parte essencial do código.

Duplicated Code - Rafael Costa 60441



The screenshot shows the IntelliJ IDEA interface with the MainApplication.java file open. A blue oval highlights the following code block:

```
        if (appBuilder.getMainArgs().help) {
            appBuilder.getClipParser().usage();
            System.exit( status: 0 );
        }
        if (appBuilder.getMainArgs().version) {
            System.out.println(GPVersion.getCurrentVersionNumber());
            System.exit( status: 0 );
        }
```

A tooltip at the bottom right of the oval says: "Kotlin code style. Do you want to update your Kotlin code style...".

Dentro de cada “if” é chamada uma função para chegar a um valor da mesma. Neste caso, para evitarmos estarmos sempre a chamar a mesma função, podemos fazer a chamada antes do “if” e guardar appBuilder.getMainArgs() em uma variável, com isso não necessitamos de estar a chamar 2 vezes o getMainArgs() da classe AppBuilder

<MainApplication.java>

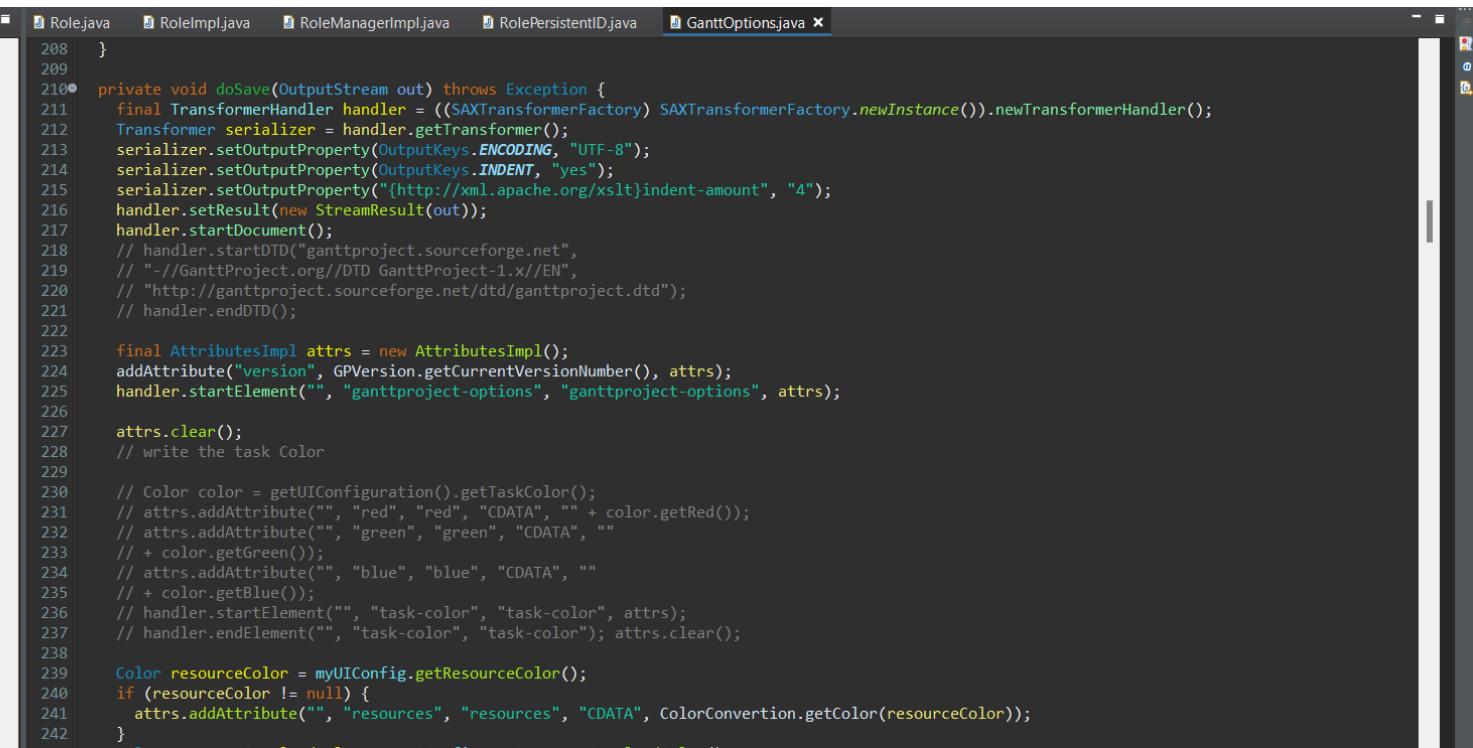
Reviews

Francisco Silveira 60816

Review duplicated code:

Concordo com o que o colega disse. Em alguns casos, guardar as variáveis que utilizamos várias vezes é extremamente importante pois, por exemplo, pode ser necessário correr grandes estruturas de dados para encontrar esses valores para além de retirar o código duplicado.

Comments that take on a “reminder” nature - Guilherme Abrantes 60971



A screenshot of an IDE showing the GanttOptions.java file. The code is annotated with several comments that serve as reminders:

```
208 }
209
210• private void doSave(OutputStream out) throws Exception {
211     final TransformerHandler handler = ((SAXTransformerFactory) SAXTransformerFactory.newInstance()).newTransformerHandler();
212     Transformer serializer = handler.getTransformer();
213     serializer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
214     serializer.setOutputProperty(OutputKeys.INDENT, "yes");
215     serializer.setOutputProperty("{http://xml.apache.org/xslt}indent-amount", "4");
216     handler.setResult(new StreamResult(out));
217     handler.startDocument();
218     // handler.startDTD("ganttproject.sourceforge.net",
219     // "-//GanttProject.org//DTD GanttProject-1.x//EN",
220     // "http://ganttproject.sourceforge.net/dtd/ganttproject.dtd");
221     // handler.endDTD();
222
223     final AttributesImpl attrs = new AttributesImpl();
224     addAttribute("version", GPVersion.getCurrentVersionNumber(), attrs);
225     handler.startElement("", "ganttproject-options", "ganttproject-options", attrs);
226
227     attrs.clear();
228     // write the task Color
229
230     // Color color = getUIConfiguration().getTaskColor();
231     // attrs.addAttribute("", "red", "red", "CDATA", "" + color.getRed());
232     // attrs.addAttribute("", "green", "green", "CDATA", ""
233     // + color.getGreen());
234     // attrs.addAttribute("", "blue", "blue", "CDATA", ""
235     // + color.getBlue());
236     // handler.startElement("", "task-color", "task-color", attrs);
237     // handler.endElement("", "task-color", "task-color"); attrs.clear();
238
239     Color resourceColor = myUIConfig.getResourceColor();
240     if (resourceColor != null) {
241         attrs.addAttribute("", "resources", "resources", "CDATA", ColorConversion.getColor(resourceColor));
242     }
```

Aqui vemos um comentário que evidencia que algo precisa de ser feito no futuro, o que aplica alterações em outros métodos para resolver devíamos ter implementado logo esta funcionalidade para não resultar em mais problemas.

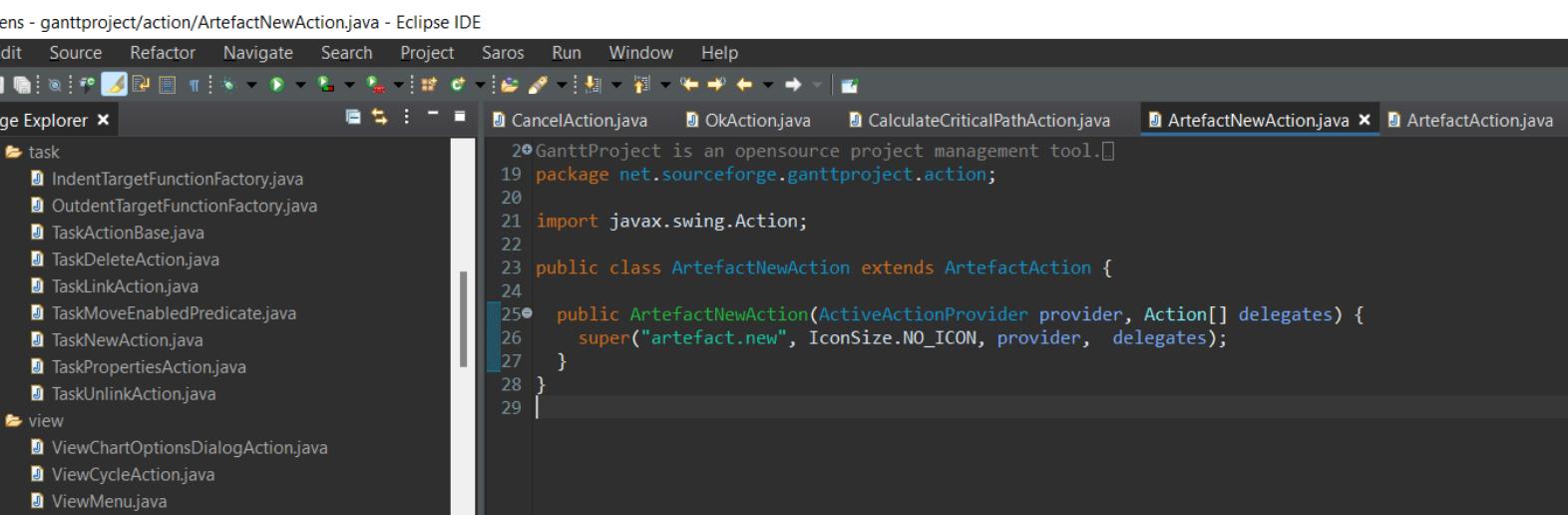
-----> (GanttOptions.java) <-----

Reviews

Rafael Costa 60441

O exposto no code smell nº1, a meu ver pode não evidenciar que pertence algo preciso de ser feito no futuro, poderá também ser algo que simplesmente deixou de ser necessário de utilizar e para "jogar pelo seguro" o troço de código foi mantido em comentário.

Data Class - Guilherme Abrantes 60971



The screenshot shows the Eclipse IDE interface with the title "ens - ganttproject/action/ArtefactNewAction.java - Eclipse IDE". The menu bar includes "Edit", "Source", "Refactor", "Navigate", "Search", "Project", "Saros", "Run", "Window", and "Help". The toolbar has various icons for file operations. The left sidebar shows the "File Explorer" with a tree view of Java files under "task" and "view" packages. The right pane displays the source code for ArtefactNewAction.java:

```
2 GanttProject is an opensource project management tool.
19 package net.sourceforge.ganttproject.action;
20
21 import javax.swing.Action;
22
23 public class ArtefactNewAction extends ArtefactAction {
24
25     public ArtefactNewAction(ActiveActionProvider provider, Action[] delegates) {
26         super("artefact.new", IconSize.NO_ICON, provider, delegates);
27     }
28 }
29
```

Aqui vemos o Code smell Dataclass sendo esta classe bastante desnecessária pois apenas tem o construtor, esta classe seria facilmente substituída por um simples método na classe ArtefactAction.

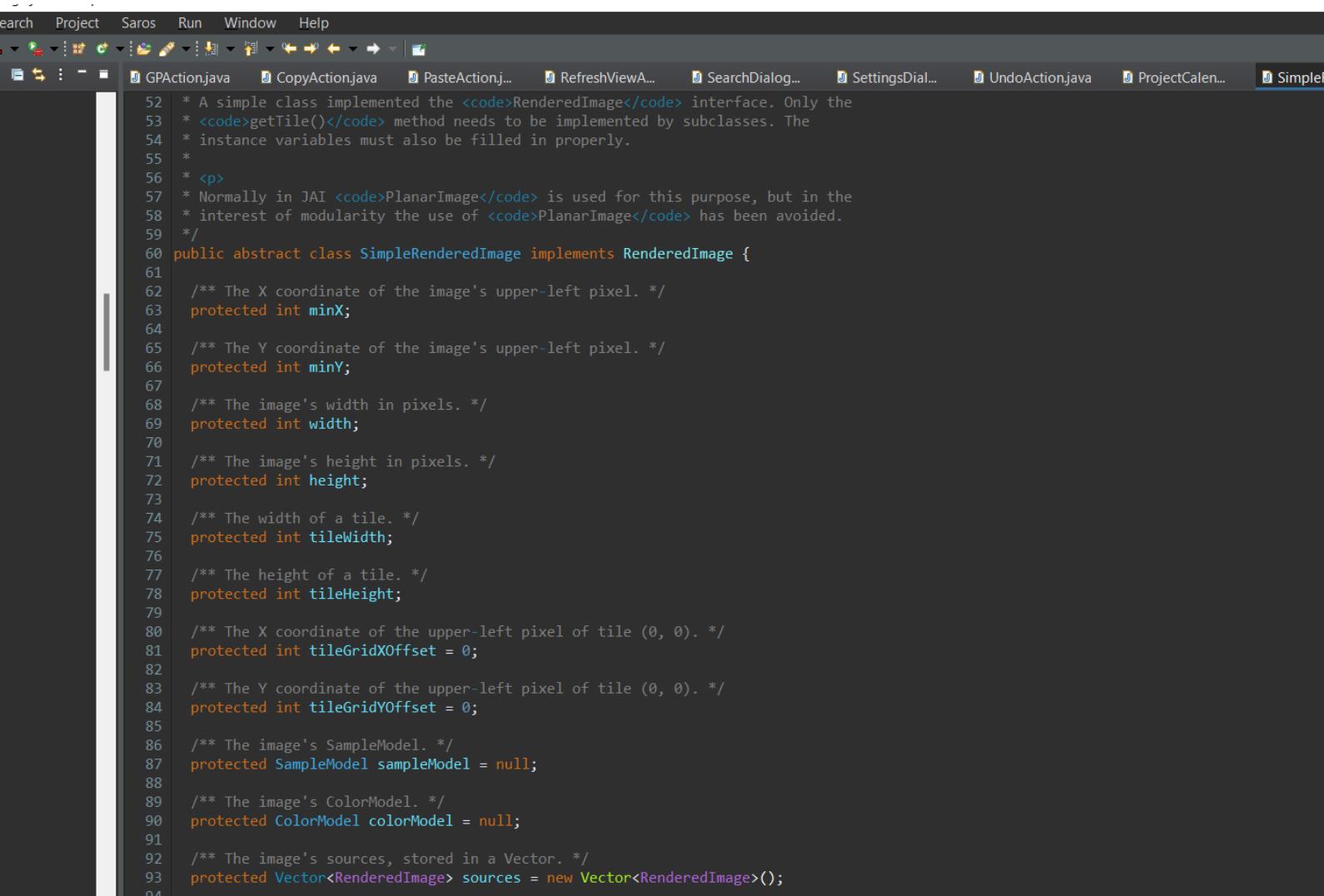
-----> (ArtefactNewAction.java) <-----

Reviews

Miguel Agostinho 60677

Concordo com o code smell nº2 pois, também acho desnecessário a criação desta classe porque apenas tem um construtor sem nenhum método.

Data Clumps - Guilherme Abrantes 60971



A screenshot of a Java IDE interface, likely Eclipse, showing the code for `SimpleRenderedImage.java`. The code is an abstract class that implements the `RenderedImage` interface. It contains several protected integer fields representing coordinates and dimensions, along with their respective getters and setters. The code is well-commented, explaining the purpose of each field and method.

```
52 * A simple class implemented the <code>RenderedImage</code> interface. Only the
53 * <code>getTile()</code> method needs to be implemented by subclasses. The
54 * instance variables must also be filled in properly.
55 *
56 * <p>
57 * Normally in JAI <code>PlanarImage</code> is used for this purpose, but in the
58 * interest of modularity the use of <code>PlanarImage</code> has been avoided.
59 */
60 public abstract class SimpleRenderedImage implements RenderedImage {
61
62     /** The X coordinate of the image's upper-left pixel. */
63     protected int minX;
64
65     /** The Y coordinate of the image's upper-left pixel. */
66     protected int minY;
67
68     /** The image's width in pixels. */
69     protected int width;
70
71     /** The image's height in pixels. */
72     protected int height;
73
74     /** The width of a tile. */
75     protected int tileSize;
76
77     /** The height of a tile. */
78     protected int tileHeight;
79
80     /** The X coordinate of the upper-left pixel of tile (0, 0). */
81     protected int tileGridXOffset = 0;
82
83     /** The Y coordinate of the upper-left pixel of tile (0, 0). */
84     protected int tileGridYOffset = 0;
85
86     /** The image's SampleModel. */
87     protected SampleModel sampleModel = null;
88
89     /** The image's ColorModel. */
90     protected ColorModel colorModel = null;
91
92     /** The image's sources, stored in a Vector. */
93     protected Vector<RenderedImage> sources = new Vector<RenderedImage>();
```

Podemos observar nesta classe bastantes inteiros que representam coordenadas e dimensões, é possível então ter uma classe com estas constantes e fazendo nessa classe alguns dos cálculos tornando tudo mais claro e fácil de perceber.

-----> (SimpleRenderedImage.java) <-----

Reviews

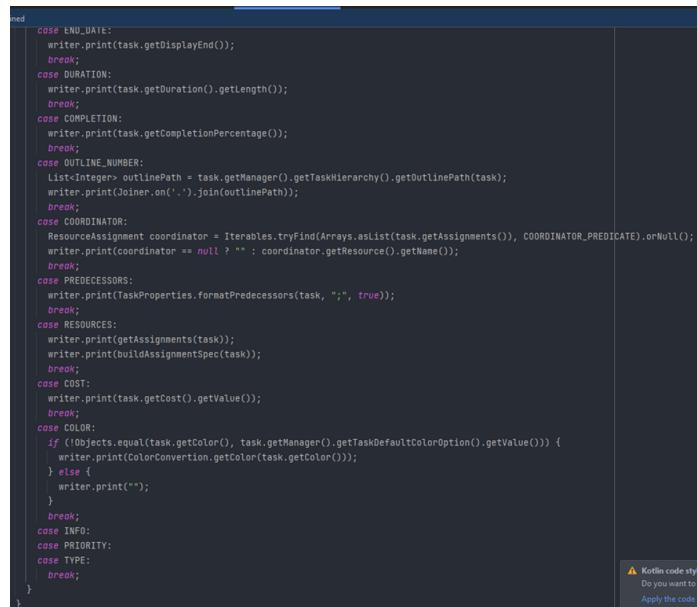
Daniel Eugénio 59797

Concordo com o code smell nº3, penso que o code smell mencionado é corretamente identificado. No entanto, penso que também poderia ser considerado Primitive Obsession, pelo uso excessivo de atributos com o tipo inteiro.

Switch statement - Francisco Silveira 60816

(GanttCSVExport.java)

Neste programa podemos ver que existem cases dentro do switch sem qualquer tipo de código, ou seja, não servem para nada. Neste caso, deveríamos remover esses cases ou em alguns casos específicos criar um case “Default” que trataria de todos os cases que não é suposto executar código.



```
med
    case END_DATE:
        writer.print(task.getDisplayEnd());
        break;
    case DURATION:
        writer.print(task.getDuration().getLength());
        break;
    case COMPLETION:
        writer.print(task.getCompletionPercentage());
        break;
    case OUTLINE_NUMBER:
        List<Integer> outlinePath = task.getManager().getTaskHierarchy().getOutlinePath(task);
        writer.print(joiner.on('.').join(outlinePath));
        break;
    case COORDINATOR:
        ResourceAssignment coordinator = Iterables.tryFind(Arrays.asList(task.getAssignments()), COORDINATOR_PREDICATE).orNull();
        writer.print(coordinator == null ? "" : coordinator.getResource().getName());
        break;
    case PREDECESSORS:
        writer.print(taskProperties.formatPredecessors(task, ";", true));
        break;
    case RESOURCES:
        writer.print(getAssignments(task));
        writer.print(buildAssignmentSpec(task));
        break;
    case COST:
        writer.print(task.getCost().getValue());
        break;
    case COLOR:
        if (!Objects.equals(task.getColor(), task.getManager().getTaskDefaultColorOption().getValue())) {
            writer.print(ColorConversion.getColor(task.getColor()));
        } else {
            writer.print("");
        }
        break;
    case INFO:
    case PRIORITY:
    case TYPE:
        break;
}
```

Reviews

Rafael Costa 60441

Concordo com o code smell nº1, pois não nos serve de nada ter condições no switch nas quais não utilizamos, para isso deveria ser inserido default no final do switch ou eliminado completamente aqueles "cases" que não estão a ser utilizados

(TaskDisplayColumnsTagHandler.java) - Francisco Silveira 60816

```
@Override
protected boolean onStartElement(Attributes attrs) {
    if (!isEnabled) {
        return false;
    }
    loadTaskDisplay(attrs);
    return true;
}
```

Como podemos ver neste pedaço de código no primeiro “if” tratamos primeiro da negação de um método booleano. Deveríamos testar a condição “isEnabled” dentro do “if” e executar “loadTaskDisplay()” e retornar “true”. De seguida depois do “if” retornamos “false”.

Reviews

Miguel Agostinho 60677

Concordo com o code smell nº2 pois, a negação de uma condição no primeiro if pode causar uma dificuldade de percepção do código.

No Comments (CalendarEditorPanel) - Francisco Silveira 60816

Esta Classe esta sem comentários, o que é essencial para a interpretação da mesma. Sem comentários, alguém ou alguma equipa que fosse trabalhar no nosso código iria ter bastante dificuldade em entender o programa. O que dificultaria o trabalho da equipa e faria com que demorasse muito mais tempo.

Reviews

Daniel Eugénio 59797

Review do code smell Not Commented, concordo com a opinião do colega. Apesar desta classe ser relativamente simples, os comentários são boas adições ao código e podem ser cruciais para o entendimento de pessoas novas, tanto ao código como ao mundo da programação.

Padrões

**Prototype Pattern (ProjectRolesOptionPageProvider.java) - Miguel Agostinho
60677**

```
  @Override
protected Role createValue(Role prototype) {
    RoleSet projectRoles = getRoleManager().getProjectRoleSet();
    return projectRoles.createRole(prototype.getName());
}
```

Neste padrão estamos a criar um objeto copiando informações de um outro, para o podermos alterar sem danificar o original.

Reviews

Guilherme Abrantes 60971

Review do pattern Prototype Pattern, concordo com a identificação deste padrão
acho que o meu colega deu uma boa explicação

Iterator Pattern (TableRowSkinBase.java) - Miguel Agostinho 60677

```
2 usages ▾ Dmitry Barashev
private void recreateCells() {
    if (cellsMap != null) {
        Collection<Reference<R>> cells = cellsMap.values();
        Iterator<Reference<R>> cellsIter = cells.iterator();
        while (cellsIter.hasNext()) {
            Reference<R> cellRef = cellsIter.next();
            R cell = cellRef.get();
            if (cell != null) {
                cell.updateIndex(-1);
                cell.getSkin().dispose();
                cell.setSkin(null);
            }
        }
        cellsMap.clear();
    }

    ObservableList<? extends TableColumnBase/*<T, ?*>/> columns = getVisibleLeafColumns();

    cellsMap = new WeakHashMap<>(columns.size());
    fullRefreshCounter = DEFAULT_FULL_REFRESH_COUNTER;
    getChildren().clear();

    for (TableColumnBase col : columns) {
        if (cellsMap.containsKey(col)) {
            continue;
        }

        // create a TableCell for this column and store it in the cellsMap
        // for future use
        createCellAndCache(col);
    }
}
```

Padrão identificado pela criação de um iterador para percorrer elementos “Reference”.

Reviews

Daniel Eugénio 59797

Review do pattern Iterator, concordo com o que foi dito pelo colega, é bastante evidente a utilização deste padrão e não tenho nada a acrescentar.

Template Method Pattern (TreeTableCells.kt) - Miguel Agostinho 60677

```
▲ Dmitry Barashev +2
override fun startEdit() {
    if (this.index == -1) {
        return
    }
    if (!isEditable) {
        onEditingCompleted()
        return
    }
    super.startEdit()
    contentDisplay = ContentDisplay.GRAPHIC_ONLY
    disclosureNode?.let {
        it.isVisible = false
    }

    if (isEditing) {
        treeTableView.requestFocus()
        doStartEdit()
    } else {
        onEditingCompleted()
    }
}
```

Este padrão consiste em fazer “override” de um método da classe super chamando esse mesmo método e adicionando mais algumas funcionalidades.

Reviews

Francisco Silveira 60816

Review do Template Method Pattern:

Concordo com o que o colega disse. Facilmente identificamos que este método chama o método da classe super e adiciona novo código.

Iterador - Daniel Eugénio 59797

```
112     private DefaultMutableTreeNode buildTree() {  
113  
114         DefaultMutableTreeNode root = new DefaultMutableTreeNode();  
115         List<HumanResource> listResources = myResourceManager.getResources();  
116         Iterator<HumanResource> itRes = listResources.iterator();  
117  
118         while (itRes.hasNext()) {  
119             HumanResource hr = itRes.next();  
120             ResourceNode rnRes = new ResourceNode(hr); // the first for the resource  
121             root.add(rnRes);  
122         }  
123         return root;  
124     }
```

- Este padrão é facilmente identificado pela criação de um objeto do tipo Iterator. É acompanhado com um ciclo while para percorrer os elementos da lista

-----> (ResourceTreeTableModel.java) <-----

Reviews

Miguel Agostinho 60677

Concordo com o pattern nº1 pois, conseguimos ver facilmente a criação de um iterador com o intuito de percorrer uma coleção de objetos.

Prototype - Daniel Eugénio 59797

```
141     public TaskBuilder withPrototype(Task prototype) {  
142         myPrototype = prototype;  
143         return this;  
144     }
```

- Este pedaço de código permite criar uma cópia dum objeto Task que nele seja passado. O método em questão é usado na classe ClipboardTaskProcessor, para criar uma cópia de uma task.

-----> (TaskManager.java) <-----

Reviews

Francisco Silveira 60816

Prototype: Neste padrão podemos facilmente encontrar um prototype. É recebido um objeto e retorna outro com os mesmos dados. Concordo com tudo o que o meu colega disse.

Façade - Daniel Eugénio 59797

```
102  class UIFacadeImpl extends ProgressProvider implements UIFacade {
103      private final JFrame myMainFrame;
104      private final ScrollingManager myScrollingManager;
105      private final ZoomManager myZoomManager;
106      private final GanttStatusBar myStatusBar;
107      private final UIFacade myFallbackDelegate;
108      private final TaskSelectionManager myTaskSelectionManager;
109      private final List<GPOptionGroup> myOptionGroups = Lists.newArrayList();
110      private final GPOptionGroup myOptions;
111      private final LafOption myLafOption;
112      private final GPOptionGroup myLogoOptions;
113      private final DefaultFileOption myLogoOption;
114      private final NotificationManagerImpl myNotificationManager;
115      private final TaskView myTaskView = new TaskView();
116      private final DialogBuilder myDialogBuilder;
117      private final Map<String, Font> myOriginalFonts = Maps.newHashMap();
118      private final List<Runnable> myOnUpdateComponentTreeUiCallbacks = Lists.newArrayList();
119      private float myLastScale = 0;
120
121      private static Map<FontSpec.Size, String> getSizeLabels() {
122          Map<FontSpec.Size, String> result = Maps.newHashMap();
123          for (FontSpec.Size size : FontSpec.Size.values()) {
124              result.put(size, GanttLanguage.getInstance().getText("optionValue.ui.appFontSpec." + size.toString() + ".label"));
125          }
126      }
127  }
```

- Aqui está implementado o padrão Façade. Neste caso, o padrão é aplicado para simplificar o API ProgressProvider.

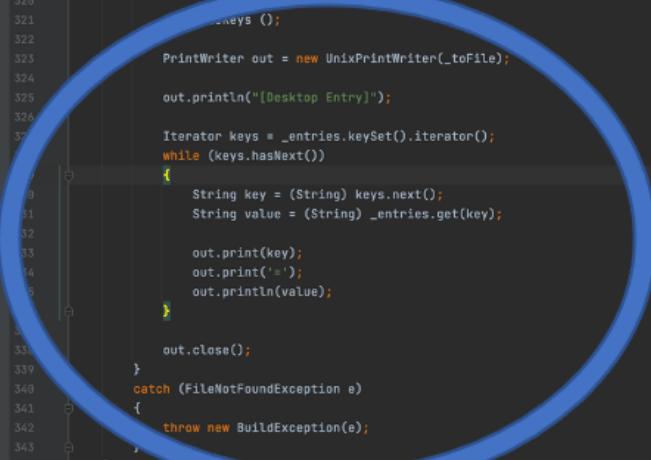
-----> (UIFacadeImpl.java) <-----

Reviews

Guilherme Abrantes 60971

Review do pattern Façade , temos aqui um padrão bastante explícito e muito bem encontrado que não levanta dúvidas, claramente aquela classe é uma façade que vai tornar as subclasses mais fáceis de usar

Iterator pattern - Rafael Costa 60441



```
ganttproject build-bin build-deb src java com googlecode ant_deb_task DesktopEntry execute
Project .gradle GantTask.java GPLLogger.java PreferenceServiceImpl.java ResourceTreeTable.java GPTreeTableBase.java DesktopEntry.java
  ganttproject ~-/Desktop/ES/ganttproject
    .github
    .gradle
    .idea
    biz.ganttproject.app.libs [libs]
    biz.ganttproject.app.localization [localization]
    biz.ganttproject.core [core]
    biz.ganttproject.desktop
    biz.ganttproject.ical [ical]
    biz.ganttproject.impex.msproject2 [msproject]
    biz.ganttproject.mxgraph [mxgraph]
    build-bin
    build-cfg
    cloud.ganttproject.api
    cloud.ganttproject.colloboque
    com.bardsoftware.constable
  ganttproject
    lib
  src
    main
      java
        biz.ganttproject
        app
        customproperty
        ganttreeview
        impex.csv
        lib
        platform
        print
        resource
        storage
        task
        FXUtil.kt
    net.sourceforge.ganttproject
      action
      application MainApplication
      calendar
      chart
      ...
  DesktopEntry.java
  310
  {
    _toFile.getParentFile().mkdirs();
    log("Generating desktop entry to: " + _toFile.getAbsolutePath());
  311
  312
  313
  314
  315
  {
    LocalizedEntry localizedEntry = (LocalizedEntry) _localizedEntries.get(i);
  316
    _entries.put(localizedEntry.toString(), localizedEntry.getValue());
  317
  318
  319
  320
  321
  322
  323
  324
  325
  326
  327
  328
  329
  330
  331
  332
  333
  334
  335
  336
  337
  338
  339
  340
  341
  342
  343
  344
  345
  346
  347
  348
  349
  350
  351
  352
  353
  354
  355
  356
  357
  358
  359
  360
  361
  362
  363
  364
  365
  366
  367
  368
  369
  370
  371
  372
  373
  374
  375
  376
  377
  378
  379
  380
  381
  382
  383
  384
  385
  386
  387
  388
  389
  390
  391
  392
  393
  394
  395
  396
  397
  398
  399
  400
  401
  402
  403
  404
  405
  406
  407
  408
  409
  410
  411
  412
  413
  414
  415
  416
  417
  418
  419
  420
  421
  422
  423
  424
  425
  426
  427
  428
  429
  430
  431
  432
  433
  434
  435
  436
  437
  438
  439
  440
  441
  442
  443
  444
  445
  446
  447
  448
  449
  450
  451
  452
  453
  454
  455
  456
  457
  458
  459
  460
  461
  462
  463
  464
  465
  466
  467
  468
  469
  470
  471
  472
  473
  474
  475
  476
  477
  478
  479
  480
  481
  482
  483
  484
  485
  486
  487
  488
  489
  490
  491
  492
  493
  494
  495
  496
  497
  498
  499
  500
  501
  502
  503
  504
  505
  506
  507
  508
  509
  510
  511
  512
  513
  514
  515
  516
  517
  518
  519
  520
  521
  522
  523
  524
  525
  526
  527
  528
  529
  530
  531
  532
  533
  534
  535
  536
  537
  538
  539
  540
  541
  542
  543
  544
  545
  546
  547
  548
  549
  550
  551
  552
  553
  554
  555
  556
  557
  558
  559
  560
  561
  562
  563
  564
  565
  566
  567
  568
  569
  570
  571
  572
  573
  574
  575
  576
  577
  578
  579
  580
  581
  582
  583
  584
  585
  586
  587
  588
  589
  590
  591
  592
  593
  594
  595
  596
  597
  598
  599
  600
  601
  602
  603
  604
  605
  606
  607
  608
  609
  610
  611
  612
  613
  614
  615
  616
  617
  618
  619
  620
  621
  622
  623
  624
  625
  626
  627
  628
  629
  630
  631
  632
  633
  634
  635
  636
  637
  638
  639
  640
  641
  642
  643
  644
  645
  646
  647
  648
  649
  650
  651
  652
  653
  654
  655
  656
  657
  658
  659
  660
  661
  662
  663
  664
  665
  666
  667
  668
  669
  670
  671
  672
  673
  674
  675
  676
  677
  678
  679
  680
  681
  682
  683
  684
  685
  686
  687
  688
  689
  690
  691
  692
  693
  694
  695
  696
  697
  698
  699
  700
  701
  702
  703
  704
  705
  706
  707
  708
  709
  710
  711
  712
  713
  714
  715
  716
  717
  718
  719
  720
  721
  722
  723
  724
  725
  726
  727
  728
  729
  730
  731
  732
  733
  734
  735
  736
  737
  738
  739
  740
  741
  742
  743
  744
  745
  746
  747
  748
  749
  750
  751
  752
  753
  754
  755
  756
  757
  758
  759
  760
  761
  762
  763
  764
  765
  766
  767
  768
  769
  770
  771
  772
  773
  774
  775
  776
  777
  778
  779
  780
  781
  782
  783
  784
  785
  786
  787
  788
  789
  790
  791
  792
  793
  794
  795
  796
  797
  798
  799
  800
  801
  802
  803
  804
  805
  806
  807
  808
  809
  810
  811
  812
  813
  814
  815
  816
  817
  818
  819
  820
  821
  822
  823
  824
  825
  826
  827
  828
  829
  830
  831
  832
  833
  834
  835
  836
  837
  838
  839
  840
  841
  842
  843
  844
  845
  846
  847
  848
  849
  850
  851
  852
  853
  854
  855
  856
  857
  858
  859
  860
  861
  862
  863
  864
  865
  866
  867
  868
  869
  870
  871
  872
  873
  874
  875
  876
  877
  878
  879
  880
  881
  882
  883
  884
  885
  886
  887
  888
  889
  890
  891
  892
  893
  894
  895
  896
  897
  898
  899
  900
  901
  902
  903
  904
  905
  906
  907
  908
  909
  910
  911
  912
  913
  914
  915
  916
  917
  918
  919
  920
  921
  922
  923
  924
  925
  926
  927
  928
  929
  930
  931
  932
  933
  934
  935
  936
  937
  938
  939
  940
  941
  942
  943
  944
  945
  946
  947
  948
  949
  950
  951
  952
  953
  954
  955
  956
  957
  958
  959
  960
  961
  962
  963
  964
  965
  966
  967
  968
  969
  970
  971
  972
  973
  974
  975
  976
  977
  978
  979
  980
  981
  982
  983
  984
  985
  986
  987
  988
  989
  990
  991
  992
  993
  994
  995
  996
  997
  998
  999
  1000
  1001
  1002
  1003
  1004
  1005
  1006
  1007
  1008
  1009
  1010
  1011
  1012
  1013
  1014
  1015
  1016
  1017
  1018
  1019
  1020
  1021
  1022
  1023
  1024
  1025
  1026
  1027
  1028
  1029
  1030
  1031
  1032
  1033
  1034
  1035
  1036
  1037
  1038
  1039
  1040
  1041
  1042
  1043
  1044
  1045
  1046
  1047
  1048
  1049
  1050
  1051
  1052
  1053
  1054
  1055
  1056
  1057
  1058
  1059
  1060
  1061
  1062
  1063
  1064
  1065
  1066
  1067
  1068
  1069
  1070
  1071
  1072
  1073
  1074
  1075
  1076
  1077
  1078
  1079
  1080
  1081
  1082
  1083
  1084
  1085
  1086
  1087
  1088
  1089
  1090
  1091
  1092
  1093
  1094
  1095
  1096
  1097
  1098
  1099
  1100
  1101
  1102
  1103
  1104
  1105
  1106
  1107
  1108
  1109
  1110
  1111
  1112
  1113
  1114
  1115
  1116
  1117
  1118
  1119
  1120
  1121
  1122
  1123
  1124
  1125
  1126
  1127
  1128
  1129
  1130
  1131
  1132
  1133
  1134
  1135
  1136
  1137
  1138
  1139
  1140
  1141
  1142
  1143
  1144
  1145
  1146
  1147
  1148
  1149
  1150
  1151
  1152
  1153
  1154
  1155
  1156
  1157
  1158
  1159
  1160
  1161
  1162
  1163
  1164
  1165
  1166
  1167
  1168
  1169
  1170
  1171
  1172
  1173
  1174
  1175
  1176
  1177
  1178
  1179
  1180
  1181
  1182
  1183
  1184
  1185
  1186
  1187
  1188
  1189
  1190
  1191
  1192
  1193
  1194
  1195
  1196
  1197
  1198
  1199
  1200
  1201
  1202
  1203
  1204
  1205
  1206
  1207
  1208
  1209
  1210
  1211
  1212
  1213
  1214
  1215
  1216
  1217
  1218
  1219
  1220
  1221
  1222
  1223
  1224
  1225
  1226
  1227
  1228
  1229
  1230
  1231
  1232
  1233
  1234
  1235
  1236
  1237
  1238
  1239
  1240
  1241
  1242
  1243
  1244
  1245
  1246
  1247
  1248
  1249
  1250
  1251
  1252
  1253
  1254
  1255
  1256
  1257
  1258
  1259
  1260
  1261
  1262
  1263
  1264
  1265
  1266
  1267
  1268
  1269
  1270
  1271
  1272
  1273
  1274
  1275
  1276
  1277
  1278
  1279
  1280
  1281
  1282
  1283
  1284
  1285
  1286
  1287
  1288
  1289
  1290
  1291
  1292
  1293
  1294
  1295
  1296
  1297
  1298
  1299
  1300
  1301
  1302
  1303
  1304
  1305
  1306
  1307
  1308
  1309
  1310
  1311
  1312
  1313
  1314
  1315
  1316
  1317
  1318
  1319
  1320
  1321
  1322
  1323
  1324
  1325
  1326
  1327
  1328
  1329
  1330
  1331
  1332
  1333
  1334
  1335
  1336
  1337
  1338
  1339
  1340
  1341
  1342
  1343
  1344
  1345
  1346
  1347
  1348
  1349
  1350
  1351
  1352
  1353
  1354
  1355
  1356
  1357
  1358
  1359
  1360
  1361
  1362
  1363
  1364
  1365
  1366
  1367
  1368
  1369
  1370
  1371
  1372
  1373
  1374
  1375
  1376
  1377
  1378
  1379
  1380
  1381
  1382
  1383
  1384
  1385
  1386
  1387
  1388
  1389
  1390
  1391
  1392
  1393
  1394
  1395
  1396
  1397
  1398
  1399
  1400
  1401
  1402
  1403
  1404
  1405
  1406
  1407
  1408
  1409
  1410
  1411
  1412
  1413
  1414
  1415
  1416
  1417
  1418
  1419
  1420
  1421
  1422
  1423
  1424
  1425
  1426
  1427
  1428
  1429
  1430
  1431
  1432
  1433
  1434
  1435
  1436
  1437
  1438
  1439
  1440
  1441
  1442
  1443
  1444
  1445
  1446
  1447
  1448
  1449
  1450
  1451
  1452
  1453
  1454
  1455
  1456
  1457
  1458
  1459
  1460
  1461
  1462
  1463
  1464
  1465
  1466
  1467
  1468
  1469
  1470
  1471
  1472
  1473
  1474
  1475
  1476
  1477
  1478
  1479
  1480
  1481
  1482
  1483
  1484
  1485
  1486
  1487
  1488
  1489
  1490
  1491
  1492
  1493
  1494
  1495
  1496
  1497
  1498
  1499
  1500
  1501
  1502
  1503
  1504
  1505
  1506
  1507
  1508
  1509
  1510
  1511
  1512
  1513
  1514
  1515
  1516
  1517
  1518
  1519
  1520
  1521
  1522
  1523
  1524
  1525
  1526
  1527
  1528
  1529
  1530
  1531
  1532
  1533
  1534
  1535
  1536
  1537
  1538
  1539
  1540
  1541
  1542
  1543
  1544
  1545
  1546
  1547
  1548
  1549
  1550
  1551
  1552
  1553
  1554
  1555
  1556
  1557
  1558
  1559
  1560
  1561
  1562
  1563
  1564
  1565
  1566
  1567
  1568
  1569
  1570
  1571
  1572
  1573
  1574
  1575
  1576
  1577
  1578
  1579
  1580
  1581
  1582
  1583
  1584
  1585
  1586
  1587
  1588
  1589
  1590
  1591
  1592
  1593
  1594
  1595
  1596
  1597
  1598
  1599
  1600
  1601
  1602
  1603
  1604
  1605
  1606
  1607
  1608
  1609
  1610
  1611
  1612
  1613
  1614
  1615
  1616
  1617
  1618
  1619
  1620
  1621
  1622
  1623
  1624
  1625
  1626
  1627
  1628
  1629
  1630
  1631
  1632
  1633
  1634
  1635
  1636
  1637
  1638
  1639
  1640
  1641
  1642
  1643
  1644
  1645
  1646
  1647
  1648
  1649
  1650
  1651
  1652
  1653
  1654
  1655
  1656
  1657
  1658
  1659
  1660
  1661
  1662
  1663
  1664
  1665
  1666
  1667
  1668
  1669
  1670
  1671
  1672
  1673
  1674
  1675
  1676
  1677
  1678
  1679
  1680
  1681
  1682
  1683
  1684
  1685
  1686
  1687
  1688
  1689
  1690
  1691
  1692
  1693
  1694
  1695
  1696
  1697
  1698
  1699
  1700
  1701
  1702
  1703
  1704
  1705
  1706
  1707
  1708
  1709
  1710
  1711
  1712
  1713
  1714
  1715
  1716
  1717
  1718
  1719
  1720
  1721
  1722
  1723
  1724
  1725
  1726
  1727
  1728
  1729
  1730
  1731
  1732
  1733
  1734
  1735
  1736
  1737
  1738
  1739
  1740
  1741
  1742
  1743
  1744
  1745
  1746
  1747
  1748
  1749
  1750
  1751
  1752
  1753
  1754
  1755
  1756
  1757
  1758
  1759
  1750
  1751
  1752
  1753
  1754
  1755
  1756
  1757
  1758
  1759
  1760
  1761
  1762
  1763
  1764
  1765
  1766
  1767
  1768
  1769
  1770
  1771
  1772
  1773
  1774
  1775
  1776
  1777
  1778
  1779
  1770
  1771
  1772
  1773
  1774
  1775
  1776
  1777
  1778
  1779
  1780
  1781
  1782
  1783
  1784
  1785
  1786
  1787
  1788
  1789
  1780
  1781
  1782
  1783
  1784
  1785
  1786
  1787
  1788
  1789
  1790
  1791
  1792
  1793
  1794
  1795
  1796
  1797
  1798
  1799
  1790
  1791
  1792
  1793
  1794
  1795
  1796
  1797
  1798
  1799
  1800
  1801
  1802
  1803
  1804
  1805
  1806
  1807
  1808
  1809
  1800
  1801
  1802
  1803
  1804
  1805
  1806
  1807
  1808
  1809
  1810
  1811
  1812
  1813
  1814
  1815
  1816
  1817
  1818
  1819
  1810
  1811
  1812
  1813
  1814
  1815
  1816
  1817
  1818
  1819
  1820
  1821
  1822
  1823
  1824
  1825
  1826
  1827
  1828
  1829
  1820
  1821
  1822
  1823
  1824
  1825
  1826
  1827
  1828
  1829
  1830
  1831
  1832
  1833
  1834
  1835
  1836
  1837
  1838
  1839
  1830
  1831
  1832
  1833
  1834
  1835
  1836
  1837
  1838
  1839
  1840
  1841
  1842
  1843
  1844
  1845
  1846
  1847
  1848
  1849
  1840
  1841
  1842
  1843
  1844
  1845
  1846
 
```

Decorator pattern - Rafael Costa 60441

The screenshot shows an IDE interface with the following details:

- Project View:** Shows the project structure under "ganttproject".
- Code Editor:** Displays the `GPCloudStatusBar.kt` file. The code implements the Decorator pattern by adding new behaviors to objects at runtime. It uses `OnlineDocumentMode` as a base class and `OnlineDocumentMode.MIRROR` and `OnlineDocumentMode.OFFLINE_ONLY` as subclasses.
- Toolbars and Menus:** Standard Java development tools like Git, TODO, Problems, Terminal, and Profiler are visible.
- Status Bar:** Shows the current commit hash (299-23), file type (LF), encoding (UTF-8), and workspace (master).
- Notifications:** A tooltip for "Kotlin code style" is present, asking if the user wants to update the code style.

Este padrão permite anexar novos comportamentos, funcionalidades ou estados extra a um objeto em tempo de execução colocando esses objetos dentro de objetos especiais que contêm os comportamentos.

<GPCloudStatusBar.kt>

Reviews

Miguel Agostinho 60677

Review do pattern nº2, aqui podemos observer um padrão "Decorator" bem identificado pois, podemos ver claramente o objeto "this"

a ser enviado com o método addDecoration para obter um novo comportamento.

Prototype Pattern - Rafael Costa 60441

The screenshot shows a Java code editor with the file `EditableList.java` open. The code implements the Prototype pattern to manage a list of objects. It includes methods for adding new objects, setting values at specific indices, and removing objects from the list. A tooltip for 'Kotlin code style' is visible, asking if the user wants to update their Kotlin code style.

```
233     }
234     return EditableList.this.isEditable(myValues.get(row));
235   }
236 
237   @Override
238   public void setValueAt(Object value, int row, int col) {
239     assert col == 0;
240     if (value == null) {
241       deleteValue(myValues.get(row));
242       myValues.remove(row);
243       fireTableRowsDeleted(row, row);
244       return;
245     }
246     T prototype = createPrototype(value);
247     if (row >= myValues.size()) {
248       if (prototype != null) {
249         T newValue = createValue(prototype);
250         if (newValue != null) {
251           myValues.add(newValue);
252           fireTableRowsInserted(myValues.size(), myValues.size());
253         }
254       } else if (row >= 0) {
255         if (prototype != myValues.get(row)) {
256           T updatedValue = updateValue(prototype, myValues.get(row));
257           myValues.set(row, updatedValue);
258           fireTableRowsUpdated(row, row);
259         }
260       } else {
261         throw new IllegalArgumentException("I can't set data in row=" + row);
262       }
263     }
264   }
265 
266   class ComboItem {
267     final String myText;
```

Através do prototype podemos criar uma cópia dum objeto que nele seja passado.

<EditableList.java>

Reviews

Daniel Eugénio 59797

Review do pattern nº3 (Prototype), concordo com o colega, é evidente o uso deste padrão, mais concretamente na linha 246. Podemos ainda ver o protótipo a ser usado mais à frente, na linha 257.

Singleton Pattern - Guilherme Abrantes 60971

```
70 private static GPCalendarProvider ourInstance;
71
72● static GPCalendar readCalendar(File resource) {
73     WeekendCalendarImpl calendar = new WeekendCalendarImpl();
74
75     HolidayTagHandler holidayHandler = new HolidayTagHandler(calendar);
76     CalendarTagHandler calendarHandler = new CalendarTagHandler(calendar, holidayHandler);
77     XmlParser parser = new XmlParser(
78         ImmutableList.<TagHandler>of(calendarHandler, holidayHandler),
79         ImmutableList.<ParsingListener>of());
80
81     try {
82         parser.parse(new BufferedInputStream(new FileInputStream(resource)));
83         return calendar;
84     } catch (IOException e) {
85         GPLogger.logToLogger("Failed to parse file "+resource.getAbsolutePath());
86         GPLogger.logToLogger(e);
87         return null;
88     }
89 }
90
91● private static List<GPCalendar> readCalendars() {
92     return HolidayCalendarKt.loadCalendars();
93 }
94
95● public static synchronized GPCalendarProvider getInstance() {
96     if (ourInstance == null) {
97         List<GPCalendar> calendars = readCalendars();
98         Collections.sort(calendars, new Comparator<GPCalendar>() {
99            ▲ 99●     public int compare(GPCalendar o1, GPCalendar o2) {
100                 return o1.getName().compareTo(o2.getName());
101             }
102         });
103         ourInstance = new GPCalendarProvider(calendars);
104     }
105
106     return ourInstance;
107 }
```

Este padrão é identificável pela criação de uma instância única OurInstance e pelo método public static synchronized que fornece acesso global a esta instância.

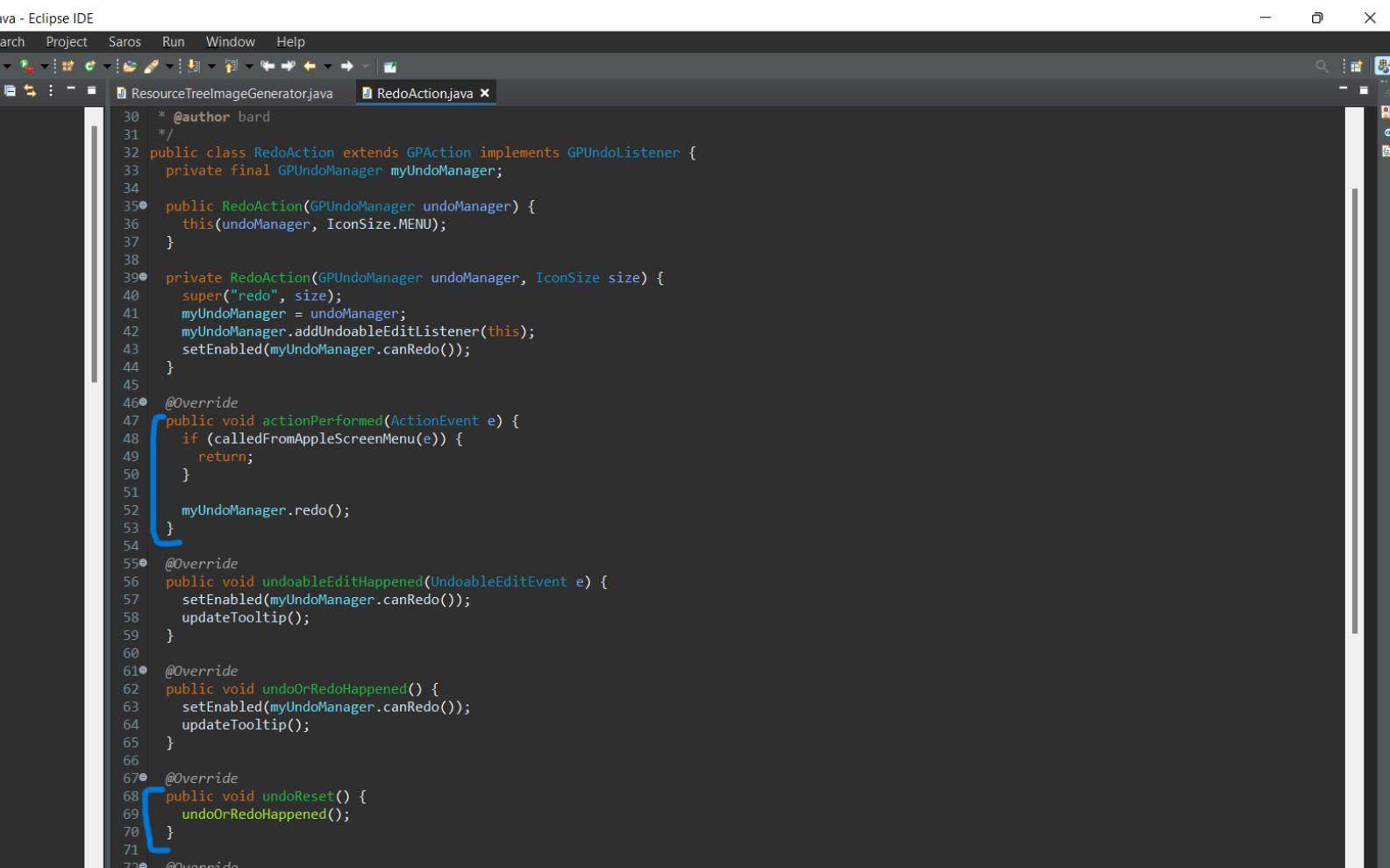
-----> (GPCalenderProvider.java) <-----

Reviews

Rafael Costa 60441

Singleton Pattern: Concordo com a identificação do padrão, mas penso que não é explícito a função do padrão

Memento Pattern - Guilherme Abrantes 60971



The screenshot shows the Eclipse IDE interface with the title "Java - Eclipse IDE". The menu bar includes "Search", "Project", "Saros", "Run", "Window", and "Help". The toolbar has icons for file operations like New, Open, Save, and Run. The left sidebar shows a project structure with "ResourceTreeImageGenerator.java" and "RedoAction.java". The main editor window displays the Java code for "RedoAction.java". The code implements the Memento pattern by using a GPUTUndoManager to store the state of edits before they are undone or redone. It overrides methods like actionPerformed, undoableEditHappened, undoOrRedoHappened, and undoReset to manage the redo functionality.

```
30  * @author bard
31  */
32 public class RedoAction extends GPAAction implements GPUTUndoListener {
33     private final GPUTUndoManager myUndoManager;
34
35     public RedoAction(GPUTUndoManager undoManager) {
36         this(undoManager, IconSize.MENU);
37     }
38
39     private RedoAction(GPUTUndoManager undoManager, IconSize size) {
40         super("redo", size);
41         myUndoManager = undoManager;
42         myUndoManager.addUndoableEditListener(this);
43         setEnabled(myUndoManager.canRedo());
44     }
45
46     @Override
47     public void actionPerformed(ActionEvent e) {
48         if (calledFromAppleScreenMenu(e)) {
49             return;
50         }
51
52         myUndoManager.redo();
53     }
54
55     @Override
56     public void undoableEditHappened(UndoableEditEvent e) {
57         setEnabled(myUndoManager.canRedo());
58         updateTooltip();
59     }
60
61     @Override
62     public void undoOrRedoHappened() {
63         setEnabled(myUndoManager.canRedo());
64         updateTooltip();
65     }
66
67     @Override
68     public void undoReset() {
69         undoOrRedoHappened();
70     }
71
72     @Override
```

Aqui podemos observar uma classe que permite que a ação corrente volte atrás para a ação antiga a partir do método undoReset().

-----> (RedoAction.java) <-----

Reviews

Miguel Agostinho 60677

Review do pattern nº2, este pattern foi bem identificado e o meu colega fez uma boa explicação do mesmo pois é notório o uso de um "undo" para voltar atrás.

Iterator Pattern - Guilherme Abrantes 60971

```
66● private Dimension calculateDimension() {  
67    int width = 0;  
68    int assignmentsCount = 0;  
69    final BufferedImage testImage = new BufferedImage(10, 10, BufferedImage.TYPE_INT_RGB);  
70    final Graphics2D g = (Graphics2D) testImage.getGraphics();  
71    final int tabSize = 5;  
72    final List<HumanResource> users = myResourceManager.getResources();  
73    for (Iterator<HumanResource> user = users.iterator(); user.hasNext();) {  
74        HumanResource hr = user.next();  
75        int nameWidth = TextLengthCalculatorImpl.getTextLength(g, hr.getName());  
76        if (nameWidth > width) {  
77            width = nameWidth;  
78        }  
79        ResourceAssignment[] assignments = hr.getAssignments();  
80        if (assignments != null) {  
81            for (int i = 0; i < assignments.length; i++) {  
82                if (isAssignmentVisible(assignments[i])) {  
83                    int taskWidth = tabSize + TextLengthCalculatorImpl.getTextLength(g, assignments[i].getTask().getName());  
84                    if (taskWidth > width) {  
85                        width = taskWidth;  
86                    }  
87                    assignmentsCount++;  
88                }  
89            }  
90        }  
91    }  
92    width += 20;  
93    int height = (assignmentsCount + users.size()) * getRowHeight() + 90;  
94    return new Dimension(width, height);  
95}  
96
```

Este padrão é facilmente identificado pela criação de um objeto do tipo Iterator. É acompanhado com um ciclo for para percorrer os elementos da lista.

-----> (ResourceTreeImageGenerator.java) <-----

Reviews

Francisco Silveira 60816

Concordo com o que o colega disse. Facilmente identificamos um padrão pela criação de um iterador.

Template Method Pattern - Francisco Silveira 60816

(ShortDateFormatOption.java & DefaultStringOption.java)

```
    @Override  
    public void LoadPersistentValue(String value) {  
        super.LoadPersistentValue(value);  
        GanttLanguage.getInstance().setShortDateFormat(myDateFormat);  
    }
```

```
    @Override  
    public void loadPersistentValue(String value) { setValue(value); }  
}
```

Neste padrão podemos observar que existe um “override” do método “loadPersistentValue”.

Isto permite modificar ou fazer algumas alterações ao método da superclasse pois as classes que estendem a super classe podem por exemplo ter uma análise e processamento de dados idêntica mas lidar com os vários formatos de dados de maneira diferente.

Reviews

Rafael Costa 60441

Sobre o pattern nº1, padrão bem identificado.

Iterator pattern - Francisco Silveira 60816

(CustumColumnsStorage.java)

```
public void addCustomColumnsListener(CustomPropertyListener listener) { myListeners.add(listener); }

private void fireCustomColumnsChange(CustomPropertyEvent event) {
    Iterator<CustomPropertyListener> it = myListeners.iterator();
    while (it.hasNext()) {
        CustomPropertyListener listener = it.next();
        listener.customPropertyChanged(event);
    }
}
```

Neste pedaço de código conseguimos facilmente identificar a criação de um iterador. Estes são utilizados para percorrer uma lista ou outro tipo de estrutura de dados com maior eficiência.

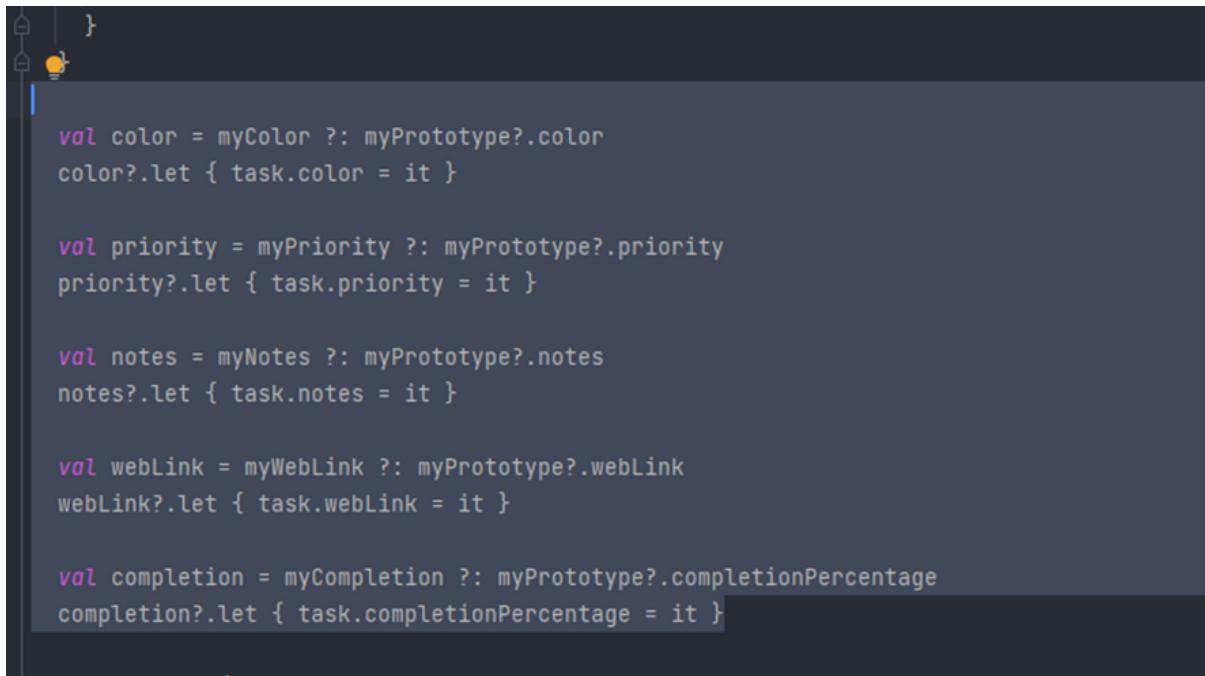
Reviews

Daniel Eugénio 59797

Sobre o pattern nº2 (iterador), concordo com o padrão identificado, no entanto, a meu ver, o iterador serve mais o propósito de percorrer uma estrutura de dados sem mexer na estrutura em si, algo que não acontece com um ciclo for ou while, e não tanto com a eficiência.

Prototype pattern - Francisco Silveira 60816

(tasManagerImpl.kt)



```
    }
}

val color = myColor ?: myPrototype?.color
color?.let { task.color = it }

val priority = myPriority ?: myPrototype?.priority
priority?.let { task.priority = it }

val notes = myNotes ?: myPrototype?.notes
notes?.let { task.notes = it }

val webLink = myWebLink ?: myPrototype?.webLink
webLink?.let { task.webLink = it }

val completion = myCompletion ?: myPrototype?.completionPercentage
completion?.let { task.completionPercentage = it }
```

Aqui podemos ver que é utilizado um protótipo de um objeto já existente para ser comparado com outros valores, ou seja, utilizamos a cópia de um objeto para podermos trabalhar nele à vontade.

Isto permite que não seja possível alterar o objeto inicial criando uma maior segurança no nosso código e dar mais liberdade ao programador pois este pode fazer os testes e alterações que ache necessário para o seu programa no protótipo.

Reviews

Rafael Costa 60441

Sobre o pattern nº3, concordo na plenitude com o exposto acerca do padrão prototype, diria completo e explícito

Complexity Metrics Miguel Agostinho (60677)

Complexity metrics	quarta	30 nov. 2022 10:57:30 WET		
Package	v(G)avg	v(G)tot		
net.sourceforge.ganttproject	1,77	1 949		
net.sourceforge.ganttproject.action	1,89	123		
net.sourceforge.ganttproject.action.edit	1,45	61		
net.sourceforge.ganttproject.action.help	1,43	20		
net.sourceforge.ganttproject.action.project	1,48	86		
net.sourceforge.ganttproject.action.resource	1,47	78		
net.sourceforge.ganttproject.action.scroll	1,38	18		
net.sourceforge.ganttproject.action.task	1,59	126		
net.sourceforge.ganttproject.action.view	1,54	20		
net.sourceforge.ganttproject.action.zoom	1,18	13		
net.sourceforge.ganttproject.application	2	2		
net.sourceforge.ganttproject.calendar	2,02	87		
net.sourceforge.ganttproject.chart	1,64	535		
net.sourceforge.ganttproject.chart.export	1,2	18		
net.sourceforge.ganttproject.chart.gantt	2,03	138		
net.sourceforge.ganttproject.chart.item	1	10		
net.sourceforge.ganttproject.chart.mouse	1,72	110		
net.sourceforge.ganttproject.chart.overview	1,7	63		
net.sourceforge.ganttproject.client	2,71	57		
net.sourceforge.ganttproject.document	1,51	214		
net.sourceforge.ganttproject.document.webdav	2,36	337		
net.sourceforge.ganttproject.excel	2	12		
net.sourceforge.ganttproject.export	1,75	210		
net.sourceforge.ganttproject.filter	1,67	10		
net.sourceforge.ganttproject.gui	1,95	805		
net.sourceforge.ganttproject.gui.about	1,33	12		
net.sourceforge.ganttproject.gui.options	1,95	332		
net.sourceforge.ganttproject.gui.options.model	12	12		
net.sourceforge.ganttproject.gui.projectwizard	1,5	93		
net.sourceforge.ganttproject.gui.scrolling	1,5	9		
net.sourceforge.ganttproject.gui.tableView	2	54		
net.sourceforge.ganttproject.gui.taskproperties	2,1	172		
net.sourceforge.ganttproject.gui.view	1,37	26		
net.sourceforge.ganttproject.gui.window	1,14	8		
net.sourceforge.ganttproject.gui.zoom	1,42	27		
net.sourceforge.ganttproject.importer	1,3	86		
net.sourceforge.ganttproject.io	2,34	208		
net.sourceforge.ganttproject.language	1,93	145		
net.sourceforge.ganttproject.parser	2,11	260		
net.sourceforge.ganttproject.plugins	3	9		
net.sourceforge.ganttproject.print	2,32	72		
net.sourceforge.ganttproject.resource	1,6	217		
net.sourceforge.ganttproject.roles	1,57	69		
net.sourceforge.ganttproject.search	1,59	59		
net.sourceforge.ganttproject.shape	1,6	8		
net.sourceforge.ganttproject.task	1,74	794		
net.sourceforge.ganttproject.task.algorithm	2,37	406		
net.sourceforge.ganttproject.task.dependency	1,6	147		
net.sourceforge.ganttproject.task.dependency.constraint	1,68	52		
net.sourceforge.ganttproject.task.event	1	24		
net.sourceforge.ganttproject.task.hierarchy	2,22	20		
net.sourceforge.ganttproject.undo	1,4	35		
net.sourceforge.ganttproject.util	3,36	84		
net.sourceforge.ganttproject.util.collect	1	4		
net.sourceforge.ganttproject.wizard	1,81	76		
project	1,81	8 622		

Obs: Para esta métrica optamos por agrupá-la por packages pois, caso contrário ficaria muito extenso.

Cyclomatic Complexity - A Cyclomatic Complexity consiste no número de decisões mais 1, sendo este número de decisões if, if ... else, switch , for loop, while loop

v(G)avg - A Average Cyclomatic Complexity é a complexidade ciclomática por função de um ficheiro/package. Portanto, esta é a soma das complexidades ciclomáticas de todas as funções a dividir pelo número de funções presentes nesse mesmo ficheiro/package.

v(G)tot - A Total Cyclomatic Complexity é a soma das complexidades ciclomáticas de funções presentes num ficheiro/package.

Pontos problemáticos:

Nesta métrica podemos observar que na package net.sourceforge.ganttproject.gui.options.model obtemos um valor extremo de Cyclomatic Complexity sendo esse valor 12 para **v(G)avg** ou seja, a média da complexidade de cada função nesta package é 12. Sendo este um valor elevado deveríamos rever o código desta package para tentar reduzir esta mesma complexidade pois, funções com grande cyclomatic complexity são difíceis de entender e também de obter cobertura total do código em testes unitários.

Por outro lado encontramos também outros extremos como por exemplo na package net.sourceforge.ganttproject.gui pois conseguimos observar que o valor de **v(G)tot** é 805 o que significa que a Cyclomatic Complexity tem um valor total de 805 nesta package mas, sendo o valor de **v(G)avg** baixo quando comparado com outras packages.

Relação com code smells:

Como explicado acima, o valor desta métrica depende do número de decisões, portanto, sempre que temos um número elevado isto pode significar que podemos encontrar uma grande sequência de if's o que representará um code smell “Long Method” (Um método que contém muitas linhas de código).

Reviews

Francisco Silveira 60816

Primeiramente, penso que o colega tenha feito um excelente trabalho na explicação de cada parâmetro, pois para além de explicar o significado de cada um ainda deu breves exemplos bastante esclarecedores. Concordo com os code smells do meu colega pois com esta métrica conseguimos facilmente identificar “Long Method”’s assim como o excesso de if’s.

MARTIN PACKAGING METRICS

RAFAEL COSTA (60441)

Martin packaging metrics	A	Ca	Ce	D	I
Package					
net.sourceforge.ganttproject	0,17	0	0	0,83	1
net.sourceforge.ganttproject.action	0,33	0	0	0,67	1
net.sourceforge.ganttproject.action.edit	0	0	0	1	1
net.sourceforge.ganttproject.action.help	0	0	0	1	1
net.sourceforge.ganttproject.action.project	0,07	0	0	0,93	1
net.sourceforge.ganttproject.action.resource	0,1	0	0	0,9	1
net.sourceforge.ganttproject.action.scroll	0	0	0	1	1
net.sourceforge.ganttproject.action.task	0,2	0	0	0,8	1
net.sourceforge.ganttproject.action.view	0	0	0	1	1
net.sourceforge.ganttproject.action.zoom	0	0	0	1	1
net.sourceforge.ganttproject.application	0	0	0	1	1
net.sourceforge.ganttproject.calendar	0,12	0	0	0,88	1
net.sourceforge.ganttproject.chart	0,34	0	0	0,66	1
net.sourceforge.ganttproject.chart.export	0,25	0	0	0,75	1
net.sourceforge.ganttproject.chart.gantt	0	0	0	1	1
net.sourceforge.ganttproject.chart.item	0	0	0	1	1
net.sourceforge.ganttproject.chart.mouse	0,38	0	0	0,62	1
net.sourceforge.ganttproject.chart.overview	0	0	0	1	1
net.sourceforge.ganttproject.client	0	0	0	1	1
net.sourceforge.ganttproject.document	0,37	0	0	0,63	1
net.sourceforge.ganttproject.document.webdav	0,11	0	0	0,89	1
net.sourceforge.ganttproject.excel	0	0	0	1	1
net.sourceforge.ganttproject.export	0,25	0	0	0,75	1
net.sourceforge.ganttproject.filter	0	0	0	1	1
net.sourceforge.ganttproject.font	0	0	0	1	1
net.sourceforge.ganttproject.gui	0,27	0	0	0,73	1
net.sourceforge.ganttproject.gui.about	0	0	0	1	1
net.sourceforge.ganttproject.gui.options	0,22	0	0	0,78	1
net.sourceforge.ganttproject.gui.options.model	1	0	0	0	1
net.sourceforge.ganttproject.gui.projectwizard	0,15	0	0	0,85	1
net.sourceforge.ganttproject.gui.scrolling	0,67	0	0	0,33	1
net.sourceforge.ganttproject.gui.tableView	0	0	0	1	1
net.sourceforge.ganttproject.gui.taskproperties	0,17	0	0	0,83	1
net.sourceforge.ganttproject.gui.view	0,5	0	0	0,5	1
net.sourceforge.ganttproject.gui.window	0	0	0	1	1
net.sourceforge.ganttproject.gui.zoom	0,25	0	0	0,75	1
net.sourceforge.ganttproject.importer	0,27	0	0	0,73	1
net.sourceforge.ganttproject.iq	0,06	0	0	0,94	1
net.sourceforge.ganttproject.language	0,29	0	0	0,71	1
net.sourceforge.ganttproject.parser	0,2	0	0	0,8	1
net.sourceforge.ganttproject.plugins	0	0	0	1	1
net.sourceforge.ganttproject.print	0	0	0	1	1
net.sourceforge.ganttproject.resource	0,35	0	0	0,65	1
net.sourceforge.ganttproject.roles	0,4	0	0	0,6	1
net.sourceforge.ganttproject.search	0,33	0	0	0,67	1
net.sourceforge.ganttproject.shape	0	0	0	1	1
net.sourceforge.ganttproject.task	0,33	0	0	0,67	1
net.sourceforge.ganttproject.task.algorithm	0,31	0	0	0,69	1
net.sourceforge.ganttproject.task.dependency	0,42	0	0	0,58	1
net.sourceforge.ganttproject.task.dependency.constraint	0,17	0	0	0,83	1
net.sourceforge.ganttproject.task.event	0,33	0	0	0,67	1
net.sourceforge.ganttproject.task.hierarchy	0	0	0	1	1
net.sourceforge.ganttproject.undo	0,5	0	0	0,5	1
net.sourceforge.ganttproject.util	0,11	0	0	0,89	1
net.sourceforge.ganttproject.util.collect	0	0	0	1	1
net.sourceforge.ganttproject.wizard	0,5	0	0	0,5	1

Acoplamentos aferentes (Afferent couplings - Ca):

O número de classes em outros pacotes que dependem de classes dentro de um pacote é um indicador da responsabilidade do pacote.

Acoplamentos eferentes (Efferent couplings - Ce):

O número de classes em outros pacotes dos quais as classes em um pacote dependem é um indicador da dependência de pacotes externos.

Abstração (A):

A relação entre o número de classes abstratas (e interfaces) e o número total de classes no pacote analisado.

O intervalo para esta métrica varia entre 0 e 1, sendo A=0 a indicação de que o pacote é completamente concreto e A=1 a indicação de que o pacote é completamente abstrato.

Instabilidade (I):

A relação entre o acoplamento eferente (Ce) e o acoplamento total (Ce + Ca) tal que $I = Ce / (Ce + Ca)$.

Esta métrica é um indicador da resiliência do pacote à mudança.

O intervalo para esta métrica varia entre 0 e 1, sendo I=0 a indicação de que o pacote é completamente estável e I=1 a indicação de que o pacote é completamente instável.

Distância da sequência principal (D):

A distância perpendicular de um pacote da linha idealizada $A + I = 1$. D é calculado como o módulo de $(A + I - 1)$ ou seja $|A + I - 1|$.

Esta métrica é um indicador do equilíbrio do pacote entre abstração e estabilidade. Um pacote diretamente na sequência principal é perfeitamente equilibrado em relação à sua abstração e estabilidade, ou seja, os pacotes ideais são completamente **abstratos e estáveis (I=0, A=1)** ou completamente **concretos e instáveis (I=1, A=0)**.

O intervalo para essa métrica varia entre 0 e 1, com D=0 indicando um pacote que coincide com a sequência principal e D=1 indicando um pacote que está o mais longe possível da sequência principal.

Pontos problemáticos:

A **instabilidade dos pacotes é visível** em todos os pacotes presentes na tabela acima representada como se pode ver através da coluna **I** e contudo através da coluna **D** a existência de pacotes ideais é elevada e muitos pacotes apresentam valores de D próximos ao que seria o valor idealizado 1, uma vez que apesar de todos os pacotes serem instáveis muitos são concretos, o que fazem com que o indicador D seja quase ideal em muitos casos. Apenas no caso do pacote net.sourceforge.ganttproject.gui.options.model o valor do indicador D apresentado é 0, ou seja não é ideal.

Code Smells:

Como explicado acima, os valores de **Afferent couplings (Ca)** e **Efferent couplings (Ce)**, são relativos ao número de classes que dependem ou estão dependentes de outras classes de outros pacotes, e daí, encontramos relação com o Code Smell Inappropriate Intimacy, uma vez que este é referente ao uso de campos e métodos de outras classes.

Reviews

Daniel Eugénio 59797

Primeiramente, penso que tenha feito um bom resumo explicativo dos parâmetros da métrica de forma concisa. Quanto à análise dos dados, penso que foi acertada, de facto há muitos pacotes quase ideais, com um valor de D perto de 1, sendo poucos os que têm um valor mais baixo. O Code Smell é também bem identificado, visto que a *inappropriate intimacy* está relacionada com acoplamento e dependência entre classes, fatores esses que podem ser analisados pelos parâmetros da métrica em causa.

MOOD Metrics Guilherme Abrantes (60971)

MOOD metrics		quarta	30 nov. 2022 10:56:14 WET					
Project	project	AHF	AIF	CF	MHF	MIF	PF	
		89,63%		68,73%	0,20%	45,86%	0,00%	100,00%

As mood metrics são constituídas por 6 métricas diferentes: **MHF**, **AHF**, **MIF**, **AIF**, **PF** e **CF**.

AHF e MHF são métricas relacionadas com a visibilidade.

AHF = 1 - Attributes Visible;

MHF = 1 - Methods Visible;

MHF tem em conta o número de métodos visíveis no nosso projeto podemos ver um MHF de 45,86% o que indica que existem quase tantos métodos públicos como privados o que pode resultar em vários métodos especializados que não podem ser reutilizados (45,86% é um número relativamente alto quando comparado com outros projetos).

AHF tem em conta o número de atributos visíveis, sendo que o nosso projeto possui um AHF de 89,63% acho um número bastante aceitável onde a maioria dos atributos são privados.

MIF e AIF são métricas relacionadas com a herança de métodos.

MIF = inherited methods / total methods available in classes

AIF = inherited attributes / total attributes available in classes

Podemos então constatar um **MIF** de 0 que pode estar a ser causado por as classes que herdam métodos dos pais estarem a redefinir todos os métodos ou então acrescentar novos, este MIF é considerado bastante mau.

O **AIF** deste projeto é de 68,73% o que sugere que bastantes atributos são herdados pelas classes filhas, o que pode não ser o ideal.

PF Polymorphism Factor:

PF = overrides / sum for each class(new methods * descendants)

A **PF** indica o grau de redefinição de métodos durante a herança de classes, podemos então observar um PF de 100% o que indica que nos estamos a dar override a tudo o que condiz com o facto do MIF ser 0%.

CF Coupling Factor:

CF = Actual couplings / Maximum possible couplings

CF mede as copulações que existem, para existir uma população de A para B, A tem de chamar métodos ou variáveis que existem em B, neste projeto este valor é de 0,20% o que indica que quase não existem classes copuladas.

Pontos problemáticos e code smells:

Sobre os pontos problemáticos temos o PF e o MIF que se completam e indicam que grande parte dos métodos são redefinidos, o que leva às classes filhas a não aproveitarem os métodos que herdam dos pais, o que pode levar ao code Smell Divergent Change, como não existem métodos comuns sempre que queremos adicionar algo temos de reescrever todos os métodos que já existem.

Reviews

Rafael Costa 60441

A descrição de cada parâmetro da métrica, penso que tenha sido simples e intuitiva, em relação ao Code Smell, realmente não é fácil de identificar um Code Smell para a métrica em questão, mas talvez como o colega descreveu o code Smell Divergent Change seja o adequado.

Lines of Code Metrics Daniel Eugénio (59797)

Lines of code metrics														
Package	CLOC	CLOC(rec)	JLOC	JLOC(rec)	LOC	LOC(rec)	LOCp	LOCp(rec)	LOCt	LOCt(rec)	NCLOC	NCLOCp	NCLOCp(rec)	NCLOCt
n/a	8 720	n/a	3 124	n/a	62 376	n/a	62 376	n/a	0	n/a	n/a	53 372	n/a	n/a
biz	n/a	97	n/a	61	n/a	1 169	n/a	1 169	n/a	0	n/a	n/a	1 072	n/a
biz.ganttp	n/a	97	n/a	61	n/a	1 169	n/a	1 169	n/a	0	n/a	n/a	1 072	n/a
biz.ganttp.impex	n/a	97	n/a	61	n/a	1 169	n/a	1 169	n/a	0	n/a	n/a	1 072	n/a
biz.ganttp.impex.csv	0	97	61	61	1 169	1 169	1 169	1 169	0	0	1 072	1 072	1 072	0
net	n/a	8 623	n/a	3 063	n/a	61 204	n/a	61 204	n/a	0	n/a	n/a	52 297	n/a
net.sourceforge	n/a	8 623	n/a	3 063	n/a	61 204	n/a	61 204	n/a	0	n/a	n/a	52 297	n/a
net.sourceforge.ganttp	0	8 623	458	3 063	12 258	61 204	12 258	61 204	0	0	10 993	10 993	52 297	0
net.sourceforge.ganttp.action	0	1 142	48	151	894	4 687	894	4 687	0	0	676	676	3 545	0
net.sourceforge.ganttp.action.edit	0	140	10	10	556	556	556	556	0	0	416	416	416	0
net.sourceforge.ganttp.action.help	0	16	5	5	186	186	186	186	0	0	170	170	170	0
net.sourceforge.ganttp.action.project	0	229	21	21	786	786	786	786	0	0	557	557	557	0
net.sourceforge.ganttp.action.resource	0	152	16	16	652	652	652	652	0	0	500	500	500	0
net.sourceforge.ganttp.action.scroll	0	76	0	0	214	214	214	214	0	0	138	138	138	0
net.sourceforge.ganttp.action.task	0	221	36	36	1 087	1 087	1 087	1 087	0	0	866	866	866	0
net.sourceforge.ganttp.action.view	0	45	6	6	182	182	182	182	0	0	137	137	137	0
net.sourceforge.ganttp.action.zoom	0	45	9	9	130	130	130	130	0	0	85	85	85	0
net.sourceforge.ganttp.application	0	8	3	3	44	44	44	44	0	0	36	36	36	0
net.sourceforge.ganttp.calendar	0	31	11	11	593	593	593	593	0	0	563	563	563	0
net.sourceforge.ganttp.chart	0	998	356	455	3 846	6 638	3 846	6 638	0	0	3 376	3 376	5 654	0
net.sourceforge.ganttp.chart.export	0	65	0	0	225	225	225	225	0	0	164	164	164	0
net.sourceforge.ganttp.chart.gantt	0	102	42	42	925	925	925	925	0	0	823	823	823	0
net.sourceforge.ganttp.chart.item	0	56	27	27	146	146	146	146	0	0	90	90	90	0
net.sourceforge.ganttp.chart.mouse	0	211	19	19	914	914	914	914	0	0	703	703	703	0
net.sourceforge.ganttp.chart.overview	0	84	11	11	582	582	582	582	0	0	498	498	498	0
net.sourceforge.ganttp.client	0	58	5	5	388	388	388	388	0	0	330	330	330	0
net.sourceforge.ganttp.document	0	425	181	328	1 666	4 115	1 666	4 115	0	0	1 433	1 433	3 367	0
net.sourceforge.ganttp.document.webdav	0	192	147	147	2 449	2 449	2 449	2 449	0	0	1 934	1 934	1 934	0
net.sourceforge.ganttp.excel	0	22	2	2	111	111	111	111	0	0	89	89	89	0
net.sourceforge.ganttp.export	0	241	24	24	1 462	1 462	1 462	1 462	0	0	1 221	1 221	1 221	0
net.sourceforge.ganttp.filter	0	30	11	11	83	83	83	83	0	0	53	53	53	0
net.sourceforge.ganttp.font	0	15	5	5	35	35	35	35	0	0	21	21	21	0
net.sourceforge.ganttp.gui	0	1 569	324	566	6 637	12 408	6 637	12 408	0	0	5 840	5 840	10 845	0
net.sourceforge.ganttp.gui.about	0	15	0	0	122	122	122	122	0	0	107	107	107	0
net.sourceforge.ganttp.gui.options	0	359	107	107	2 493	2 676	2 493	2 676	0	0	2 180	2 180	2 321	0
net.sourceforge.ganttp.gui.options.model	0	42	0	0	183	183	183	183	0	0	141	141	141	0
net.sourceforge.ganttp.gui.projectwizard	0	125	7	7	770	770	770	770	0	0	645	645	645	0
net.sourceforge.ganttp.gui.scrolling	0	9	25	25	88	88	88	88	0	0	79	79	79	0
net.sourceforge.ganttp.gui.tableView	0	24	20	20	418	418	418	418	0	0	394	394	394	0
net.sourceforge.ganttp.gui.taskproperties	0	136	36	36	1 200	1 200	1 200	1 200	0	0	1 064	1 064	1 064	0
net.sourceforge.ganttp.gui.view	0	61	16	16	260	260	260	260	0	0	199	199	199	0
net.sourceforge.ganttp.gui.window	0	15	6	6	73	73	73	73	0	0	58	58	58	0
net.sourceforge.ganttp.gui.zoom	0	26	25	25	164	164	164	164	0	0	138	138	138	0
net.sourceforge.ganttp.importer	0	179	19	19	784	784	784	784	0	0	605	605	605	0
net.sourceforge.ganttp.io	0	256	33	33	1 425	1 425	1 425	1 425	0	0	1 175	1 175	1 175	0
net.sourceforge.ganttp.language	0	76	186	186	942	942	942	942	0	0	866	866	866	0
net.sourceforge.ganttp.parser	0	295	83	83	1 687	1 687	1 687	1 687	0	0	1 393	1 393	1 393	0
net.sourceforge.ganttp.plugins	0	14	5	5	63	63	63	63	0	0	49	49	49	0
net.sourceforge.ganttp.print	0	109	2	2	808	808	808	808	0	0	699	699	699	0
net.sourceforge.ganttp.resource	0	158	57	57	1 200	1 200	1 200	1 200	0	0	1 042	1 042	1 042	0
net.sourceforge.ganttp.roles	0	95	24	24	509	509	509	509	0	0	414	414	414	0
net.sourceforge.ganttp.search	0	116	26	26	593	593	593	593	0	0	477	477	477	0
net.sourceforge.ganttp.shape	0	31	3	3	90	90	90	90	0	0	59	59	59	0
net.sourceforge.ganttp.task	0	1 194	247	443	4 708	8 725	4 708	8 725	0	0	4 223	4 223	7 532	0
net.sourceforge.ganttp.task.algorithm	0	256	112	112	2 176	2 176	2 176	2 176	0	0	1 921	1 921	1 921	0
net.sourceforge.ganttp.task.dependency	0	346	47	72	1 065	1 504	1 065	1 504	0	0	804	804	1 158	0
net.sourceforge.ganttp.task.dependency.constrai	0	85	25	25	439	439	439	439	0	0	354	354	354	0
net.sourceforge.ganttp.task.event	0	78	12	12	220	220	220	220	0	0	142	142	142	0
net.sourceforge.ganttp.task.hierarchy	0	29	0	0	117	117	117	117	0	0	88	88	88	0
net.sourceforge.ganttp.undo	0	60	14	14	284	284	284	284	0	0	224	224	224	0
net.sourceforge.ganttp.util	0	179	110	113	631	666	631	666	0	0	466	466	487	0
net.sourceforge.ganttp.util.collect	0	14	3	3	35	35	35	35	0	0	21	21	21	0
net.sourceforge.ganttp.wizard	0	48	36	36	606	606	606	606	0	0	558	558	558	0
org	n/a	n/a	n/a	n/a	3	n/a	3	n/a	0	0	3	n/a	3	n/a
org.ganttp	0	0	0	0	3	3	3	3	0	0	3	3	3	0

Alguns constituintes da métrica Lines of Code:

LOC - Lines Of Code

- Todas linhas de código, incluindo comentários

NCLOC - Non-Comment Lines of Code

- Linhas de código excluindo os comentários

LOCp - Lines of product code

- Linhas de código “eficaz” (source code). Exclui comentários e linhas em branco.

Pontos problemáticos:

Esta métrica é difícil de avaliar, visto ser mais eficaz quando avaliada ao nível da classe, no entanto, avaliar um projeto desta dimensão à classe demoraria muito tempo, pelo que achámos melhor ver os valores ao nível da package. A nível da package, a package com mais linhas é a package “mãe”, pelo que se justifica o valor de linhas apresentado, por esta ter várias packages em si.

net.sourceforge.ganttproject.gui.GanttTaskPropertiesBean	27	16	466
net.sourceforge.ganttproject.gui.UIUtil	19	4	416
Média de linhas por classe = 73,8726			

Aqui estão dois exemplos, ao nível da classe, de valores que estão fora do normal para a média da aplicação, o que pode implicar dificuldades na leitura e interpretação destas classes e que podiam ser divididas em classes mais pequenas.

Code Smells

Apesar de difícil de avaliar, esta métrica pode ser extremamente útil para identificar code smells como Large Class ou Duplicate Code, no caso de uma classe se destacar das outras no número de linhas. Outro Code Smell é o No Comments, detetável em classes onde o número de linhas seja igual, independentemente do parâmetro ser LOC ou NCLOC.

Em relação à análise do meu colega, concordo que esta métrica é difícil de ser avaliada no formato de packages, em relação aos code smells concordo que possa ajudar a identificar os code smells referidos mas também o code smell “Comments” que é identificado quando uma classe tem excesso de comentários.

Reviews

Miguel Agostinho 60677

Em relação à análise do meu colega, concordo que esta métrica é difícil de ser avaliada no formato de packages, em relação aos code smells concordo que possa ajudar a identificar os code smells referidos mas também o code smell “Comments” que é identificado quando uma classe tem excesso de comentários.

Dependency Metrics Francisco Silveira (60816)

Package	Cyclic	PDcy	PDpt	PDpt*
net.sourceforge.ganttproject	50	53	52	64
net.sourceforge.ganttproject.action	50	8	21	64
net.sourceforge.ganttproject.action.edit	50	12	2	64
net.sourceforge.ganttproject.action.help	50	8	1	64
net.sourceforge.ganttproject.action.project	50	13	2	64
net.sourceforge.ganttproject.action.resource	50	10	1	64
net.sourceforge.ganttproject.action.scroll	50	7	1	64
net.sourceforge.ganttproject.action.task	50	10	2	64
net.sourceforge.ganttproject.action.view	50	8	1	64
net.sourceforge.ganttproject.action.zoom	50	2	5	64
net.sourceforge.ganttproject.application	0	1	0	1
net.sourceforge.ganttproject.calendar	50	12	2	64
net.sourceforge.ganttproject.chart	50	27	22	64
net.sourceforge.ganttproject.chart.export	50	5	5	64
net.sourceforge.ganttproject.chart.gantt	50	18	1	64
net.sourceforge.ganttproject.chart.item	50	1	4	64
net.sourceforge.ganttproject.chart.mouse	50	18	2	64
net.sourceforge.ganttproject.chart.overview	50	7	1	64
net.sourceforge.ganttproject.client	50	7	1	64
net.sourceforge.ganttproject.document	50	17	12	64
net.sourceforge.ganttproject.document.webdav	50	8	1	64
net.sourceforge.ganttproject.excel	50	1	1	64
net.sourceforge.ganttproject.export	50	17	7	64
net.sourceforge.ganttproject.filter	0	0	3	65
net.sourceforge.ganttproject.font	0	0	3	65
net.sourceforge.ganttproject.gui	50	30	39	64
net.sourceforge.ganttproject.gui.about	50	4	2	64
net.sourceforge.ganttproject.gui.options	50	19	19	64
net.sourceforge.ganttproject.gui.options.model	50	4	8	64
net.sourceforge.ganttproject.gui.projectwizard	50	8	4	64
net.sourceforge.ganttproject.gui.scrolling	0	1	6	65
net.sourceforge.ganttproject.gui.tableView	50	7	1	64
net.sourceforge.ganttproject.gui.taskproperties	50	17	2	64
net.sourceforge.ganttproject.gui.view	50	7	4	64
net.sourceforge.ganttproject.gui.window	50	1	1	64
net.sourceforge.ganttproject.gui.zoom	0	1	9	65
net.sourceforge.ganttproject.importer	50	20	6	64
net.sourceforge.ganttproject.io	50	19	9	64
net.sourceforge.ganttproject.language	50	5	35	64
net.sourceforge.ganttproject.parser	50	16	7	64
net.sourceforge.ganttproject.plugins	50	3	6	64
net.sourceforge.ganttproject.print	50	8	2	64
net.sourceforge.ganttproject.resource	50	10	21	64
net.sourceforge.ganttproject.roles	50	1	16	64
net.sourceforge.ganttproject.search	50	7	2	64
net.sourceforge.ganttproject.shape	0	0	1	65
net.sourceforge.ganttproject.task	50	22	36	64
net.sourceforge.ganttproject.task.algorithm	50	12	14	64
net.sourceforge.ganttproject.task.dependency	50	8	20	64
net.sourceforge.ganttproject.task.dependency.constraint	50	7	10	64
net.sourceforge.ganttproject.task.event	50	3	4	64
net.sourceforge.ganttproject.task.hierarchy	50	1	1	64
net.sourceforge.ganttproject.undo	50	6	13	64
net.sourceforge.ganttproject.util	50	3	15	64
net.sourceforge.ganttproject.util.collect	0	0	12	65
net.sourceforge.ganttproject.wizard	50	5	3	64

Dependency Metrics

Uma métrica dependente é uma métrica em que suas variáveis de influência são explicitadas. Um exemplo é a escalabilidade da taxa de transferência em relação ao número de clientes que acessam uma conexão de dados. Aqui, uma métrica (a taxa de transferência) é fornecida em dependência de outra variável (o número de clientes).

Cyclic:

- número de dependências cyclic por package. Esta ocorre quando um package A depende de outro package B, e o B também depende do A.

PDcy:

- número de dependências por package, ou seja, por quantos packages está dependente o package em questão.

PDpt:

- número de packages dependentes, ou seja, quantos packages estão dependentes do package em questão.

PDpt*:

- número de packages dependentes transitivamente. Considerando três packages: A, B, C. Se C for dependente de B e B for dependente de A então C é uma dependência transitiva para A.

Pontos Problemáticos:

Como podemos ver nas colunas “Cyclic” e “PDpt*” Os valores apresentados são praticamente os mesmos, ou seja, não conseguimos observar algo em concreto apenas que estes packages dependem quase todos uns dos outros.

Code Smells:

Esta métrica pode ajudar-nos a identificar vários tipos de code smells como por exemplo o Shotgun Surgery no caso de querermos alterar algo simples no programa mas temos que alterar várias classes para o fazermos. Outro code smell que podemos identificar é o Feature Envy no caso de existir um método que está mais preocupado em manipular dados de outra classe em vez da sua. Estes code smells podem ser identificados observando as suas dependências. No caso do Feature Envy podemos observá-lo se existir uma classe com poucas dependências e muitos dependentes.

Atenção estes exemplos podem ocorrer (ou ocorrer de outra maneira dependendo do programa em questão) e não existir nenhum code smell apenas são algumas maneiras de podermos identificá-los.

Reviews

Guilherme Abrantes 60971

Em relação à análise do meu colega, acho que o colega conseguiu descrever bem e explicar o que consiste cada ponto desta métrica porém podia ter pegado num exemplo em específico e ter dado a sua análise, em relação aos code smells concordo com o que o meu colega disse é difícil identificar um em específico mas certos valores de dependências podem levar a certos smells.
