Relatório da fase 1 do projeto de Engenharia de Software

Membros:

- Francisco Vasco - 61028
- Iago Paulo - 60198
- James Furtado - 61177
- João Oliveira - 61052
- Ricardo Gonçalo - 60519

Índece

Design Patterns

Abstract Factory Pattern

```
import ...

public abstract class CalendarFactory {
  public static interface LocaleApi {
    Locale getLocale();
    DateFormat getShortDateFormat();
  }

  private static LocaleApi ourLocaleApi;

  public static Calendar newCalendar() { return (Calendar) Calendar.getInstance(ourLocaleApi.getLocale()).clone(); }

  protected static void setLocaleApi(LocaleApi localeApi) { ourLocaleApi = localeApi; }

  public static GanttCalendar createGanttCalendar(Date date) { return new GanttCalendar(date, ourLocaleApi); }

  public static GanttCalendar createGanttCalendar(int year, int month, int date) {
    return new GanttCalendar(year, month, date, ourLocaleApi);
  }

  public static GanttCalendar createGanttCalendar() { return new GanttCalendar(ourLocaleApi); }
```

Ficheiro:biz.ganttproject.core\src\main\java\biz\ganttproject\core\time\CalendarFactory
Texto: Este cria um Calendario de Gant diferente do tipo GanttCalendar dependendo dos parâmetros usados ou uma nova instância de calendário do tipo Calendar.
Autor: Iago Paulo
Revisão: Estamos perante um Factory Pattern. Pois o construtor foi alterado para as funcões createGanttCalendar que recebem alguns parâmetros e devolvem um produto **by** João Oliveira

Builder Pattern

```
21  import ...
42
43
44  fun main(args: Array<String>) {
45    var builder = AppBuilder(args).withLogging().withWindowVisible().runBeforeUi {
46      RootLocalizer = SingleTranslationLocalizer(ResourceBundle.getBundle( baseName: "i18n"))
47      PluginManager.setCharts(listOf())
48      GanttLanguage.getInstance()
49    }
50    if (getCloudEnv() == GPCloudEnv.EMULATOR) {
51      builder = builder.withDocument( path: "cloud://asdfg/Test Team/Test Project")
52    }
53    builder.whenAppInitialized {  it: GanttProject
54      it.updater = DummyUpdater
55    }.launch()
56  }
57
58  val mainWindow = AtomicReference<GanttProject?>( initialValue: null)
59
60  /**
```

```
129       }
130       whenWindowOpened {  it: JFrame
131         Platform.runLater {
132           Thread.currentThread().uncaughtExceptionHandler = Unca
133               _, e -> GPLogger.log(e)
134           }
135         }
136       }
137       return this
138     }
139     fun withSplash(): AppBuilder {
140       val splashCloser = showAsync().get()
141       whenWindowOpened {  it: JFrame
142         try {
143           splashCloser.run()
144         } catch (ex: Exception) {
145           ex.printStackTrace()
146         }
147       }
148       return this
149     }
150     fun withWindowVisible(): AppBuilder {
151       whenAppInitialized { ganttProject ->
152         SwingUtilities.invokeLater { ganttProject.doShow() }
153       }
154       return this
155     }
```

Ficheiro:ganttproject\ganttproject\src\main\java\net\sourceforge\ganttproject\App.kt
Texto: Este é um builder da aplicação no qual em cada método retorna o proprio builder para poder usar mais metodos em sequencia de adições para o App e finalmente tem um launch para lançar o programa.
Autor: Iago Paulo

Revisão: Sim, dá os métodos necessários para construir a aplicação **by** Ricardo Gonçalo

```java
public class BottomUnitSceneBuilder extends AbstractSceneBuilder {

  public static interface InputApi {
    int getTopLineHeight();
    OffsetList getBottomUnitOffsets();
    TimeFormatter getFormatter(TimeUnit offsetUnit, Position lowerLine);
  }

  private final InputApi myInputApi;

  public BottomUnitSceneBuilder(Canvas timelineCanvas, InputApi inputApi) {
    super(timelineCanvas);
    myInputApi = inputApi;
  }

  @Override
  public void build() {
    Offset prevOffset = null;
    List<Offset> bottomOffsets = getBottomUnitOffsets();
    int xpos = bottomOffsets.get(0).getOffsetPixels();
    if (xpos > 0) {
      xpos = 0;
    }
    TimeFormatter formatter = null;
    TextGroup textGroup = null;

    for (Offset offset : bottomOffsets) {
      renderScaleMark(offset, prevOffset);
      if (formatter == null) {
```

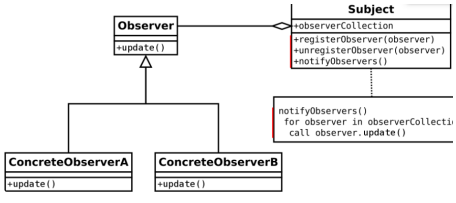
```java
public abstract class AbstractSceneBuilder implements SceneBuilder {
  private final Canvas myCanvas;
  private int myHeight;

  protected AbstractSceneBuilder() { myCanvas = new Canvas(); }

  protected AbstractSceneBuilder(Canvas canvas) { myCanvas = canvas; }

  public void setHeight(int height) { myHeight = height; }

  protected int getHeight() { return myHeight; }

  public Canvas getCanvas() { return myCanvas; }

  @Override
  public void reset(int sceneHeight) {
    getCanvas().clear();
    setHeight(sceneHeight);
  }
```

Ficheiro:biz.ganttproject.core/src/main/java/biz/ganttproject/core/chart/scene/AbstractSceneBuilder.java
Texto: Build pattern em AbstractSceneBuilder e SceneBuilder, que são base para coisas como ChartRendererBase, BottomUnitSceneBuilder, builders para Canvas
Autor: Ricardo Gonçalo
Revisão: Tem as características de de um builder pattern padrão e aparenta funcionar como tal **by** Iago Paulo

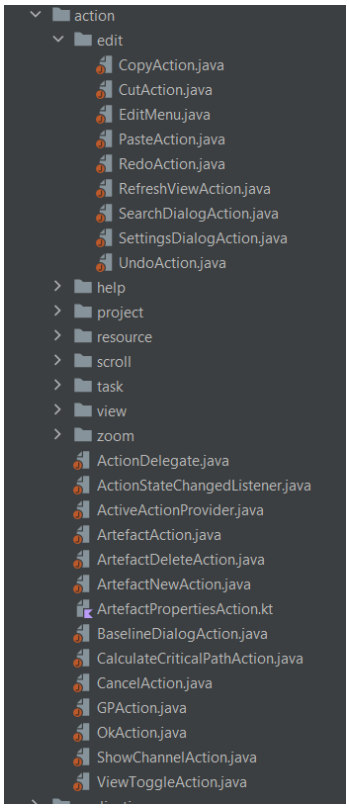
Command Pattern

```java
/**
 * @author bard
 */
public class ScrollingManagerImpl implements ScrollingManager {

  public ScrollingManagerImpl() {
  }

  @Override
  public void scrollBy(TimeDuration duration) {
    for (ScrollingListener l : myListeners) {
      l.scrollBy(duration);
    }
  }

  @Override
  public void scrollBy(int pixels) {
    for (ScrollingListener l : myListeners) {
      l.scrollBy(pixels);
    }
  }

  @Override
  public void scrollTo(Date date) {
    for (ScrollingListener l : myListeners) {
      l.scrollTo(date);
    }
  }

  @Override
  public void addScrollingListener(ScrollingListener listener) { myListeners.add(listener); }

  @Override
  public void removeScrollingListener(ScrollingListener listener) { myListeners.remove(listener); }

  List<ScrollingListener> myListeners = new ArrayList<~>();
}
```

Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/gui/scrolling/ ScrollingManagerImpl.java
Texto: A classe ScrollingManagerImpl serve como manager dos comandos de "scrolling".
Autor: Francisco Vasco
Revisão: Francisco diz ser um Command Pattern apesar de que para mim estar mais parecido com um Observer Pattern devido às semelhanças dos métodos com o que vimos na aula prática sobre Observer Pattern:

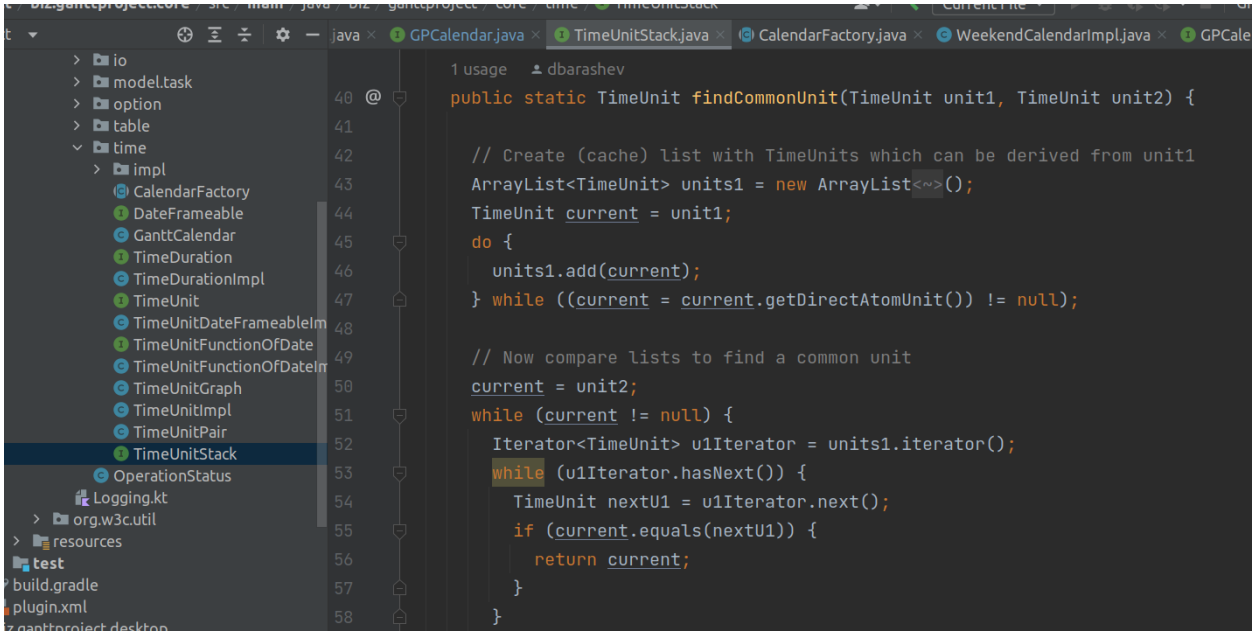


by Iago Paulo



Ficheiro:
ganttproject\src\main\java\net\sourceforge\ganttproject\action
Texto: Verifica-se um **Command** pattern, em que ações são classes em vez de apenas funções, verificado por exemplo na utilização destas ações em ganttproject/src/main/java/net/sourceforge/ganttproject/GanttProject.java
Autor: Ricardo Gonçalo
Revisão: As várias ações do programa, como o zoom, o scroll, etc, estão em classes separadas, encapsulando assim a informação necessária para executar cada uma das ações **by** João Oliveira

Iterator Pattern



Ficheiro:biz.ganttproject.core/src/main/java/biz/ganttproject/core/time/TimeUnitStack.java
Texto: A classe Util que é um inner class da Interface TimeUnitStack usa o patter Iterator no método estático findCommonUnit.
Autor: James Furtado
Revisão: De facto a classe tem um iterador e está a usa-lo para acessar elementos, logo é um Pattern Iterator **by** Iago Paulo

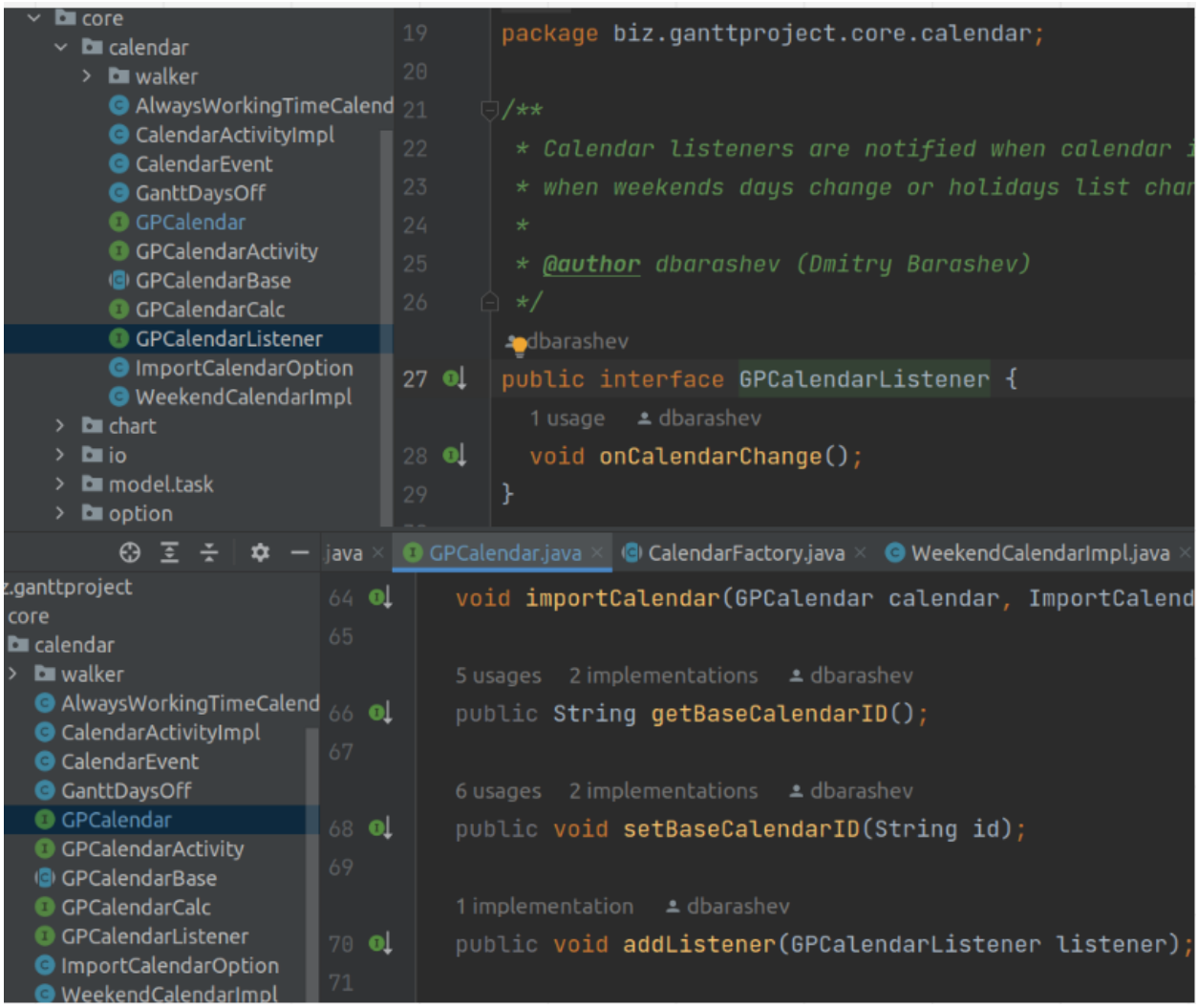
Memento Pattern

```
3 usages
38    private final Document myDocumentBefore;
39
```

```
dbarashev +2
@Override
public void undo() throws CannotUndoException {
  try {
    restoreDocument(myDocumentBefore);
    if (projectDatabaseTxn != null) {
      try {
        projectDatabaseTxn.undo();
      } catch (ProjectDatabaseException e) {
        GPLogger.log(e);
      }
    }
  } catch (DocumentException | IOException e) {
    undoRedoExceptionHandler(e);
  }
}
```

Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/undo/UndoableEditImpl.java
Texto: Guarda um objeto com o estado anterior do documento, tendo um método undo() que restaura o documento ao seu estado anterior.
Autor: João Oliveira
Revisão: Correto, é possível restaurar o estado anterior do documento a partir do método undo() ao manter uma snapshot do estado anterior **by** Ricardo Gonçalo

Observer Pattern



Ficheiro:biz.ganttproject.core/src/main/java/biz/ganttproject/core/calendar/
Texto: No package biz.ganttproject.core.calendar há um observer pattern onde a interface do Subject é o GPCalendar e a interface do Observer é GPCalenderListener.
Autor: James Furtado
Revisão: Implementações de GPCalendar (o Subject) mantêm uma coleção de GPCalendarListener (adicionados através do método addListener) e notificam cada um ao efetuar mudanças ao calendário, o padrão parece-me correto **by** Francisco Vasco

Singleton

```
79    private static final GanttLanguage ganttLanguage = new GanttLanguage();
```

```
      dbarashev
22    public static GanttLanguage getInstance() {
23      return ganttLanguage;
24    }
25
```

Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/parser/AbstractTagHandler.javaganttproject/src/main/java/net/sourceforge/ganttproject/language/GanttLanguage.java
Texto: Uso da função getInstance() que retorna uma única instância do objeto, que é inicializado uma única vez, através do uso de uma variável estática (linha 79)
Autor: João Oliveira
Revisão: Correto, implementação normal com Singleton Pattern **by** Francisco Vasco

```
95    public static synchronized GPCalendarProvider getInstance() {
96      if (ourInstance == null) {
97        List<GPCalendar> calendars = readCalendars();
98        Collections.sort(calendars, new Comparator<GPCalendar>() {
99          public int compare(GPCalendar o1, GPCalendar o2) { return o1.getName().compareTo(o2.getName()); }
102       });
103       ourInstance = new GPCalendarProvider(calendars);
104     }
105     return ourInstance;
106   }
```

Ficheiro:ganttproject\ganttproject\src\main\java\net\sourceforge\ganttproject\calendar\GPCalendarProvider.java
Texto: Utilização do método GetInstance implementado da maneira esperada para um pattern Singleton
Autor: Iago Paulo
Revisão: De facto este método em conjunto com a variável ourInstance assegura que apenas existe uma instância de GPCalendarProvider. **by** Francisco Vasco

```java
public interface RoleManager {
  public RoleSet createRoleSet(String name);

  public RoleSet[] getRoleSets();

  /** Clear the role list */
  public void clear();

  /** Return all roles except the default roles */
  // public String [] getRolesShort();
  public Role[] getProjectLevelRoles();

  public class Access {
    public static RoleManager getInstance() { return ourInstance; }

    private static RoleManager ourInstance = new RoleManagerImpl();
  }
```

Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/roles/RoleManager.java
Texto: A classe Access na linha 39-45 assegura a instanciação de apenas um RoleManager
Autor: Francisco Vasco
Revisão: A inner class Access da class RoleManager usa mesmo o pattern singleton **by** James Furtado

Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/language/GanttLanguage.java

Texto: A classe GanttLanguage tem um construtor privado e um método estático getIntance() que devolve a única instância da classe.
Autor: James Furtado
Revisão: Sim, não haveria melhor maneira de dizer que é singleton **by** Ricardo Gonçalo

Template method

Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/gui/options/OptionPageProviderBase.java
Texto: A classe abstrata que é estendida por 11 classes. 9 das classes reescrevem os metodos hasCustomComponent() e buildPageComponent()
Autor: João Oliveira
Revisão: Sim de facto isto é um template método. A classe é abstrata e as subclasses rescrevem esses e outros métodos **by** James Furtado

```
192
193    protected String getIconFilePrefix() { return null; }
196
```

```
45     @Override
46     protected String getIconFilePrefix() {
47       return isOn() ? ICON_PREFIX_ON : ICON_PREFIX_OFF;
48     }
```

```
34     @Override
35     protected String getIconFilePrefix() {
36       return "exit_";
37     }
```

Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/action/GPAction.java (linhas 138-151 e 193-195)
Texto: O método final createIcon(String iconSize) na classe abstrata GPAction chama o método getIconFilePrefix() cuja implementação concreta varia depois nas classes que estendem GPAction
Autor: Francisco Vasco
Revisão: Estamos perante um padrão de comportamento onde na classe está o método getIconFilePrefix() que é depois alterado por 19 classes que estendem a classe. **by** João Oliveira

Facade pattern

```
class UIFacadeImpl extends ProgressProvider implements UIFacade {
  6 usages
  private final JFrame myMainFrame;
  2 usages
  private final ScrollingManager myScrollingManager;
  2 usages
  private final ZoomManager myZoomManager;
  5 usages
  private final GanttStatusBar myStatusBar;
  17 usages
  private final UIFacade myFallbackDelegate;
  3 usages
  private final TaskSelectionManager myTaskSelectionManager;
  3 usages
  private final List<GPOptionGroup> myOptionGroups = Lists.newArrayList();
  5 usages
  private final GPOptionGroup myOptions;
  5 usages
  private final LafOption myLafOption;
  3 usages
  private final GPOptionGroup myLogoOptions;
  5 usages
  private final DefaultFileOption myLogoOption;
  3 usages
  private final NotificationManagerImpl myNotificationManager;
  1 usage
  private final TaskView myTaskView = new TaskView();
  2 usages
  private final DialogBuilder myDialogBuilder;
```

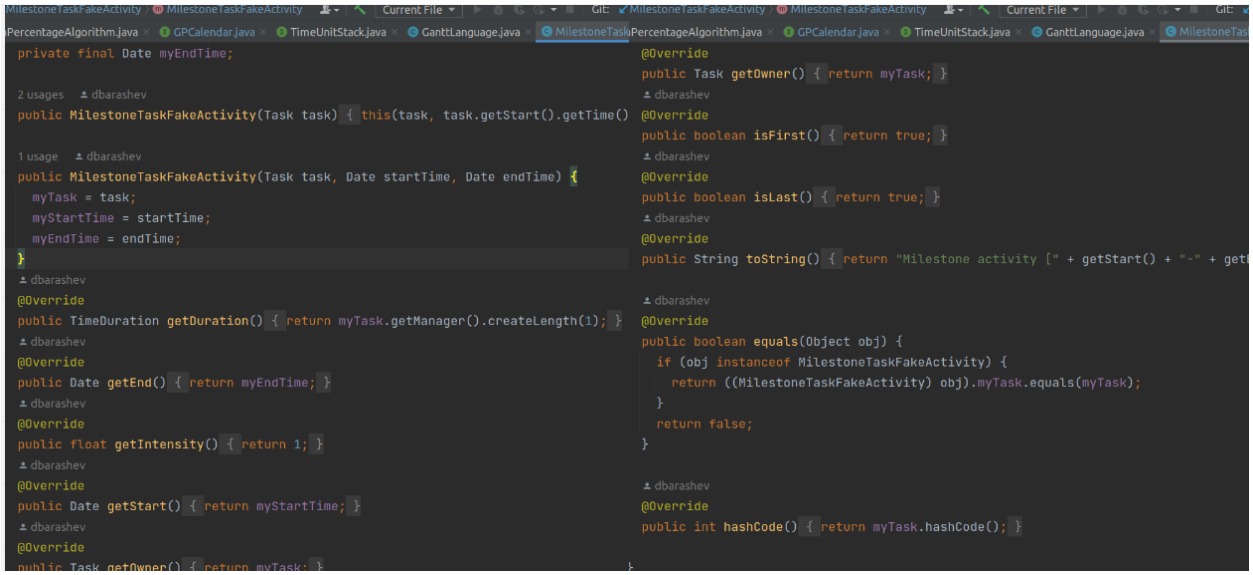
```
/** @returns an object containing the zoom related actions */
ZoomActionSet getZoomActionSet();

GPUndoManager getUndoManager();

void setLookAndFeel(GanttLookAndFeelInfo laf);

GanttLookAndFeelInfo getLookAndFeel();

Choice showConfirmationDialog(String message, String title);

void showPopupMenu(Component invoker, Action[] actions, int x, int y);

void showPopupMenu(Component invoker, Collection<Action> actions, int x, int y);

void showOptionDialog(int messageType, String message, Action[] actions);

Dialog createDialog(Component content, Action[] buttonActions, String title);

void setStatusText(String text);

void showErrorDialog(String errorMessage);

void showNotificationDialog(NotificationChannel channel, String message);

void showSettingsDialog(String pageID);
/**
 * Shows the given exception in an error dialog and also puts it into the log
 * file
```

Ficheiro:
ganttproject/src/main/java/net/sourceforge/ganttproject/UIFacadeImpl.java
ganttproject/src/main/java/net/sourceforge/ganttproject/gui/UIFacade.java

Texto: Representam um façade template, que tenta esconder toda a complexidade do UI
Autor: Ricardo Gonçalo
Revisão: O nome da classe é bem descritivo, tudo o que posso dizer é que concordo **by** James Furtado

Code Smells

Data Class



```
private final Date myEndTime;

2 usages  dbarashev
public MilestoneTaskFakeActivity(Task task) { this(task, task.getStart().getTime()

1 usage  dbarashev
public MilestoneTaskFakeActivity(Task task, Date startTime, Date endTime) {
  myTask = task;
  myStartTime = startTime;
  myEndTime = endTime;
}
dbarashev
@Override
public TimeDuration getDuration() { return myTask.getManager().createLength(1); }
dbarashev
@Override
public Date getEnd() { return myEndTime; }
dbarashev
@Override
public float getIntensity() { return 1; }
dbarashev
@Override
public Date getStart() { return myStartTime; }
dbarashev
@Override
public Task getOwner() { return myTask; }
dbarashev
@Override
public boolean isFirst() { return true; }
dbarashev
@Override
public boolean isLast() { return true; }
dbarashev
@Override
public String toString() { return "Milestone activity [" + getStart() + "-" + get
dbarashev
@Override
public boolean equals(Object obj) {
  if (obj instanceof MilestoneTaskFakeActivity) {
    return ((MilestoneTaskFakeActivity) obj).myTask.equals(myTask);
  }
  return false;
}
dbarashev
@Override
public int hashCode() { return myTask.hashCode(); }
```

Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/chart/MilestoneTaskFakeActivity.java
Texto: A class MilestoneTaskFakeActivity só tem getters.
Solução: Tentar dar alguma responsabilidade a classe.
Autor: James Furtado
Revisão: Confirma-se que a classe só tem "getters". A solução sugerida parece-me aceitável, outra solução seria guardar estas informações nas classes que as utilizam **by** Francisco Vasco

```java
public class RssUpdate {

  final String myVersion;
  final String myUrl;
  final String myDescription;

  public RssUpdate(String version, String url, String description) {
    myVersion = version;
    myUrl = url;
    myDescription = description;
  }

  public String getVersion() { return myVersion; }

  public String getUrl() { return myUrl; }

  public String getDescription() { return myDescription; }
```

Ficheiro: ganttproject/src/main/java/net/sourceforge/ganttproject/client/RssUpdate.java
Texto: Isto é uma Data Class, totalmente definida pelas 3 primitivas que tem, a sua existência isolada acrescenta complexidade desnecessária
Solução: Esta class podia muito bem estar nested na RssParser, já que é apenas aí que é usada
Autor: Ricardo Gonçalo
Revisão: Seria uma solução viável tendo em conta a redução de complexidade **by** Iago Paulo

Data Clump

```java
@Override
public void put(T data, int x, int y, int width, int height) {
  myRects.add(new Rect<T>(data, x, y, width, height));
  myValues.add(data);
}

@Override
public T get(int x, int y) {
  return get(x, xpadding: 0, y, ypadding: 0);
}

public T get(int x, int xpadding, int y, int ypadding) {
  for (Rect<T> r : myRects) {
```

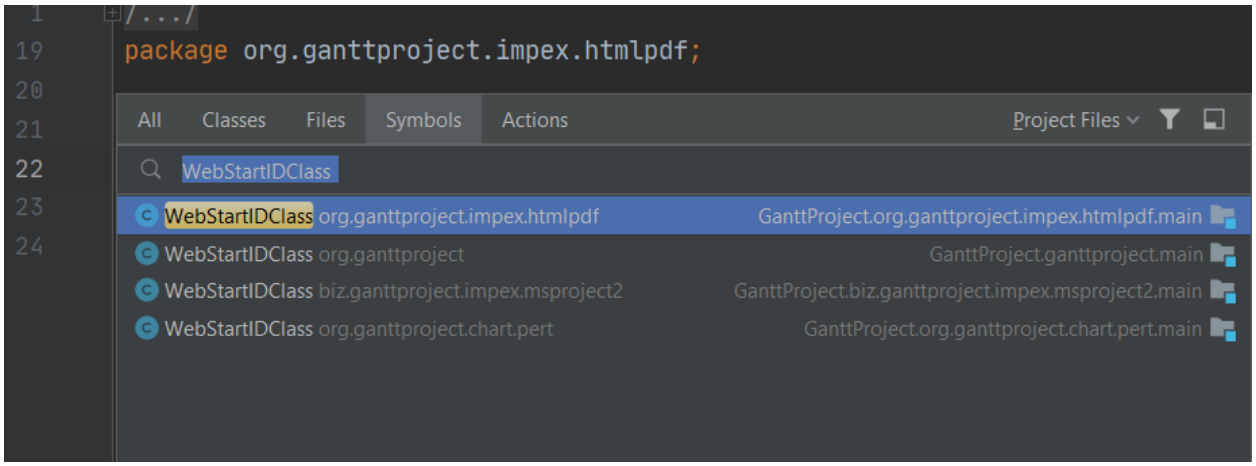
Ficheiro:ganttproject-master\biz.ganttproject.core\src\main\java\biz\ganttproject\core\chart\canvas\DummySpatialIndex.java
Texto: Lista de parâmetros muito longa.
Solução: Criar uma classe para agrupar alguns dos argumentos por exemplo o x e o y.
Autor: Iago Paulo
Revisão: É de facto uma data clump **by** João Oliveira

DeadCode

```java
/.../
package net.sourceforge.ganttproject.filter;

/**
 * Class to select a filter for the FileChooser object (*.gan and *.xml are
 * accepted)
 */
public class GanttXMLFileFilter extends ExtensionBasedFileFilter {
  public GanttXMLFileFilter() { super( fileExtension: "xml|gan", description: "GanttProject files (.gan, .xml)"); }
}
```

Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/filter/GanttXMLFileFilter.java
Texto: A classe GanttXMLFileFilter nunca é usada

Solução: Apagar a classe já que nunca é utilizada.
Autor: Francisco Vasco
Revisão: A classe GanttXMLFileFilter não está a ser usada pelo que pode ser apagada em segurança sem afetar o programa **by** João Oliveira

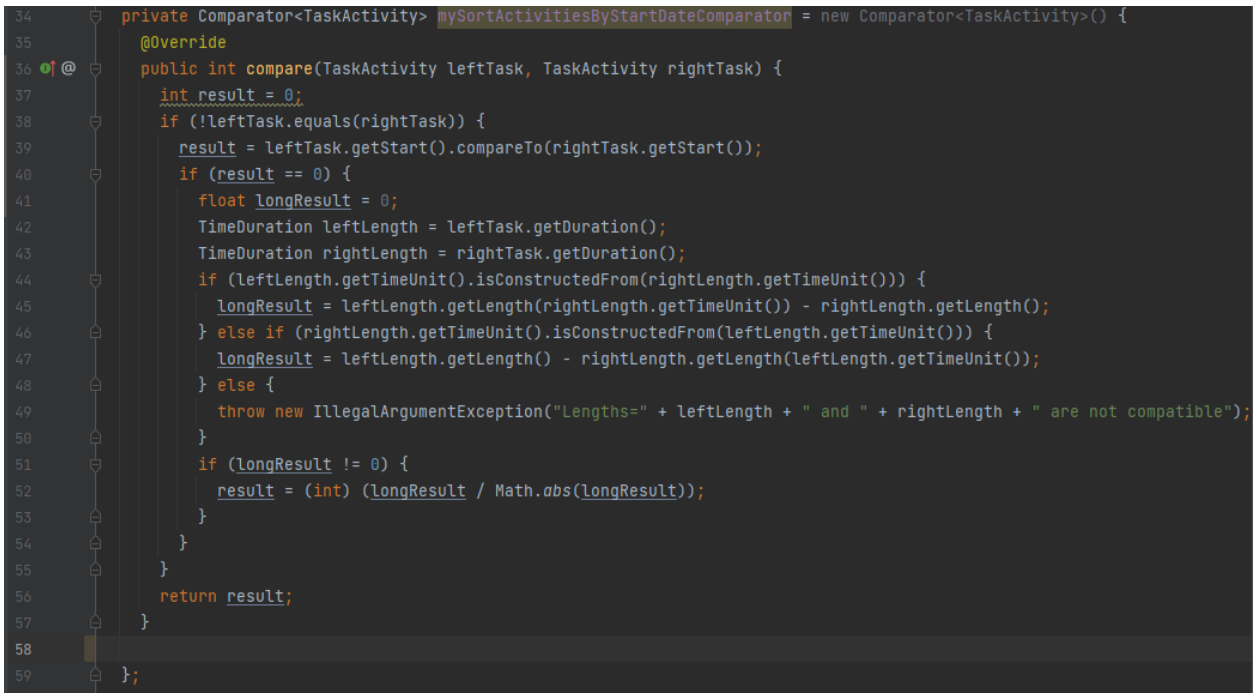


Ficheiro:gantProj\ganttproject\biz.ganttproject.impex.msproject2\src\main\java\biz\ganttproject\impex\msproject2\WebStartIDClass
Texto: A classe WebStartIDClass vazia (todas as classes com este nome também estão), talvez também se possa considerar Specular Generality dependendo do contexto.
Solução: Apagar as classes… Ou fazer uso delas
Autor: Iago Paulo
Revisão: Concordo que é DeadCode, não é utilizado e que pode ser também uma situação de preparação/"over-engineering" em antecipação de funcionalidades futuras **by** Francisco Basco

Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/action/ArtefactAction.java
Texto: A variável myDelegates (private final Action[] myDelegates) é inicializada, mas nunca usada.
Solução: Pode ser apagada sem causar nenhum problema.
Autor: João Oliveira
Revisão: Sim realmente é verdade e como é privada não é acessível de fora, a única coisa que poe estar a fazer é a impedir objetos de serem apanhados pelo garbage collector **by** Ricardo Gonçalo

Duplicated Code

```
private Comparator<Task> mySortTasksByStartDateComparator = new Comparator<Task>() {
  @Override
  public int compare(Task leftTask, Task rightTask) {
    int result = 0;
    if (!leftTask.equals(rightTask)) {
      result = leftTask.getStart().compareTo(rightTask.getStart());
      if (result == 0) {
        float longResult = 0;
        TimeDuration leftLength = leftTask.getDuration();
        TimeDuration rightLength = rightTask.getDuration();
        if (leftLength.getTimeUnit().isConstructedFrom(rightLength.getTimeUnit())) {
          longResult = leftLength.getLength(rightLength.getTimeUnit()) - rightLength.getLength();
        } else if (rightLength.getTimeUnit().isConstructedFrom(leftLength.getTimeUnit())) {
          longResult = leftLength.getLength() - rightLength.getLength(leftLength.getTimeUnit());
        } else {
          throw new IllegalArgumentException("Lengths=" + leftLength + " and " + rightLength + " are not compatible");
        }
        if (longResult != 0) {
          result = (int) (longResult / Math.abs(longResult));
        }
      }
    }
    return result;
  }

};
```



```
private Comparator<TaskActivity> mySortActivitiesByStartDateComparator = new Comparator<TaskActivity>() {
  @Override
  public int compare(TaskActivity leftTask, TaskActivity rightTask) {
    int result = 0;
    if (!leftTask.equals(rightTask)) {
      result = leftTask.getStart().compareTo(rightTask.getStart());
      if (result == 0) {
        float longResult = 0;
        TimeDuration leftLength = leftTask.getDuration();
        TimeDuration rightLength = rightTask.getDuration();
        if (leftLength.getTimeUnit().isConstructedFrom(rightLength.getTimeUnit())) {
          longResult = leftLength.getLength(rightLength.getTimeUnit()) - rightLength.getLength();
        } else if (rightLength.getTimeUnit().isConstructedFrom(leftLength.getTimeUnit())) {
          longResult = leftLength.getLength() - rightLength.getLength(leftLength.getTimeUnit());
        } else {
          throw new IllegalArgumentException("Lengths=" + leftLength + " and " + rightLength + " are not compatible");
        }
        if (longResult != 0) {
          result = (int) (longResult / Math.abs(longResult));
        }
      }
    }
    return result;
  }

};
```

Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/task/algorithm/SortTasksAlgorithm.java
Texto: Na classe SortTasksAlgorithm o código nas linhas 44-53 é idêntico ao código das linhas 71-80

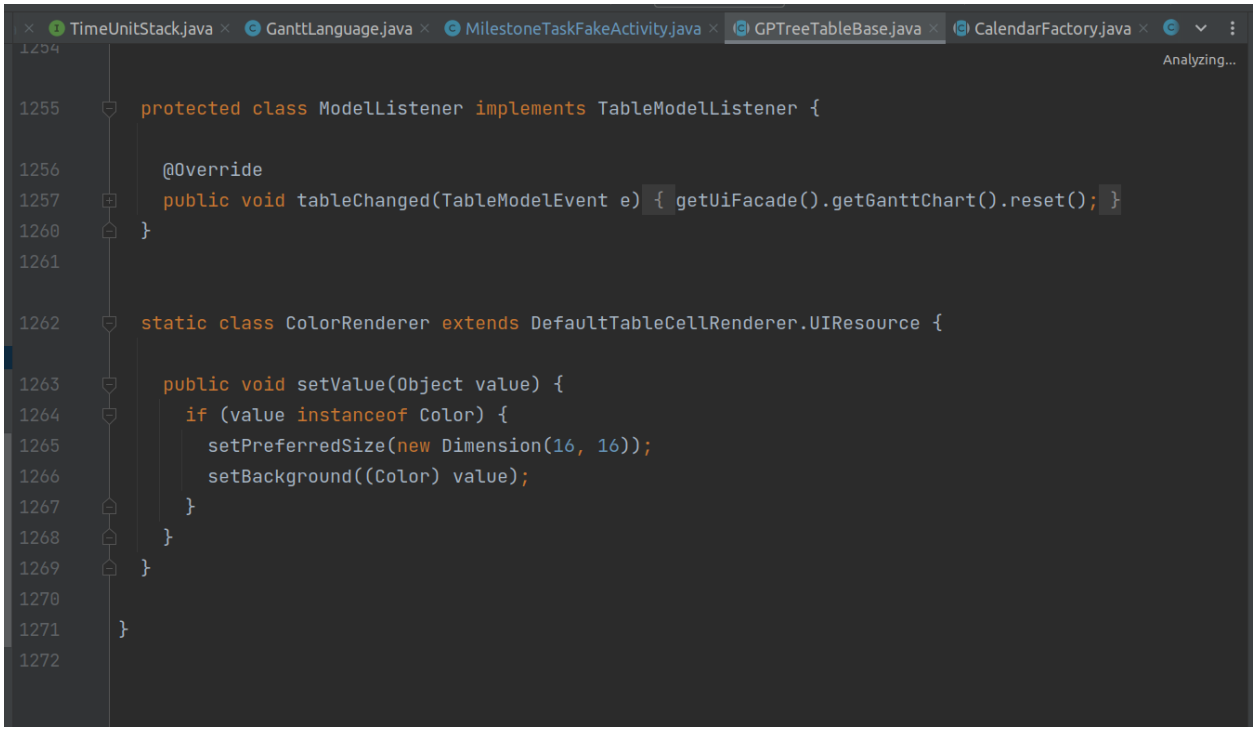
Solução: Extrair parte do código duplicado para um método auxiliar.
Autor: Francisco Vasco
Revisão: Estes 2 pedaços de código são de facto exatamente iguais, um método auxiliar seria uma solução viável **by** Iago Paulo

```
114     final UrlFetcher urlFetcher = new UrlFetcher() {
            4 usages  dbarashev
115         @Override
116         protected void onFetchComplete(File file) {
117             super.onFetchComplete(file);
118             onSelectedFileChange(file);
119         }
120     };
121     myUrlField = new JTextField();
122     Box urlBox = Box.createVerticalBox();
123     urlBox.add(myUrlField);
124     urlBox.add(myUrlLabel);
            dbarashev
125     myUrlField.getDocument().addDocumentListener(new DocumentListener() {
            dbarashev
126         @Override
127         public void removeUpdate(DocumentEvent e) { onChange(); }
130
            dbarashev
131         @Override
132         public void insertUpdate(DocumentEvent e) { onChange(); }
135
            dbarashev
136         @Override
137         public void changedUpdate(DocumentEvent e) { onChange(); }
140
            3 usages  dbarashev
141         private void onChange() { urlFetcher.setUrl(getSelectedUrl()); }
144     });
```

Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/gui/FileChooserPageBase.java e ganttproject/src/main/java/net/sourceforge/ganttproject/wizard/AbstractFileChooserPage.java

Texto: Código na linha 114 da primeira classe está duplicado em ambas as classes.
Solução: O código podia ser extraído para um método ou para uma classe auxiliar
Autor: João Oliveira
Revisão: Parece-me correta a identificação. Embora se trate de apenas uma linha podem surgir erros após um futuro refactoring que se evitam com esta solução. **by** Francisco Vasco

Large class



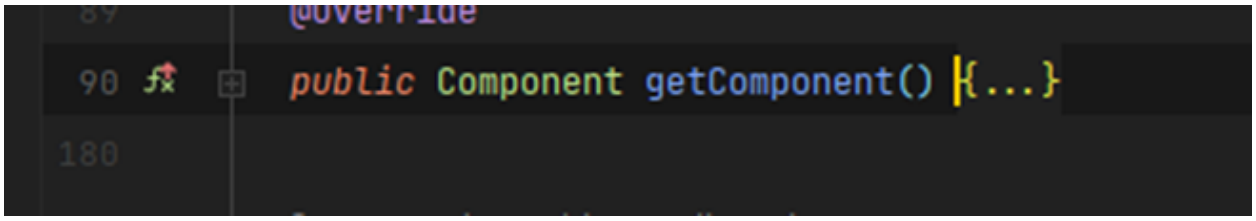
Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/GPTreeTableBase.java
Texto: A class GPTreeTableBase tem mais de 1000 linha de código além de ter muitas outras classes dentro dela mesma e métodos muito longos.
Solução: Dividir a class em outras classes mais pequenas.
Autor: James Furtado
Revisão: Classe tem mais de 1200 linhas e é composto por 6 classes dentro do ficheiro, de facto é uma classe longa. **by** Iago Paulo

Long Method

```
private void doSave(OutputStream out) throws Exception {
  final TransformerHandler handler = ((SAXTransformerFactory) SAXTransformer
  Transformer serializer = handler.getTransformer();
  serializer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
  serializer.setOutputProperty(OutputKeys.INDENT, "yes");
  serializer.setOutputProperty("{http://xml.apache.org/xslt}indent-amount",
  handler.setResult(new StreamResult(out));
  handler.startDocument();
  // handler.startDTD("ganttproject.sourceforge.net",
  // "-//GanttProject.org//DTD GanttProject-1.x//EN",
  // "http://ganttproject.sourceforge.net/dtd/ganttproject.dtd");
  // handler.endDTD();

  final AttributesImpl attrs = new AttributesImpl();
  addAttribute("version", GPVersion.getCurrentVersionNumber(), attrs);
  handler.startElement("", "ganttproject-options", "ganttproject-options",
```

```
  //addAttribute("category", "chart-main", attrs);
  //addAttribute("spec", getFontSpec(getUIConfiguration().getChartMainFont()), attrs);
  //emptyElement("font", attrs, handler);

  saveRoleSets(handler);
  new OptionSaver().saveOptionMap(myGPOptions.entrySet(), handler);
  savePreferences(myPluginPreferencesRootNode.node("/configuration"), handler);
  savePreferences(myPluginPreferencesRootNode.node("/instance"), handler);
  endElement("ganttproject-options", handler);

  GPLogger.log("[GanttOptions] save(): finished!!");
  handler.endDocument();
}
```

Ficheiro:
ganttproject/src/main/java/net/sourceforge/ganttproject/GanttOptions.java
Texto: Em no método doSave, linhas .210 - 358, tem-se método demasiado grande e código comentado
Solução: Simplificar as tarefas da função em subtarefas, para se tornar de mais fácil leitura e compreensão
Autor: Ricardo Gonçalo
Revisão: O método está de facto demasiado grande, apesar de ter todo o objetivo de inicializar a UI. Podiam ter sido usadas funções auxiliares para tornar mais claro o que é feito no método. Há também pedaços de código comentados que deviam ser apagados. **by** João Oliveira



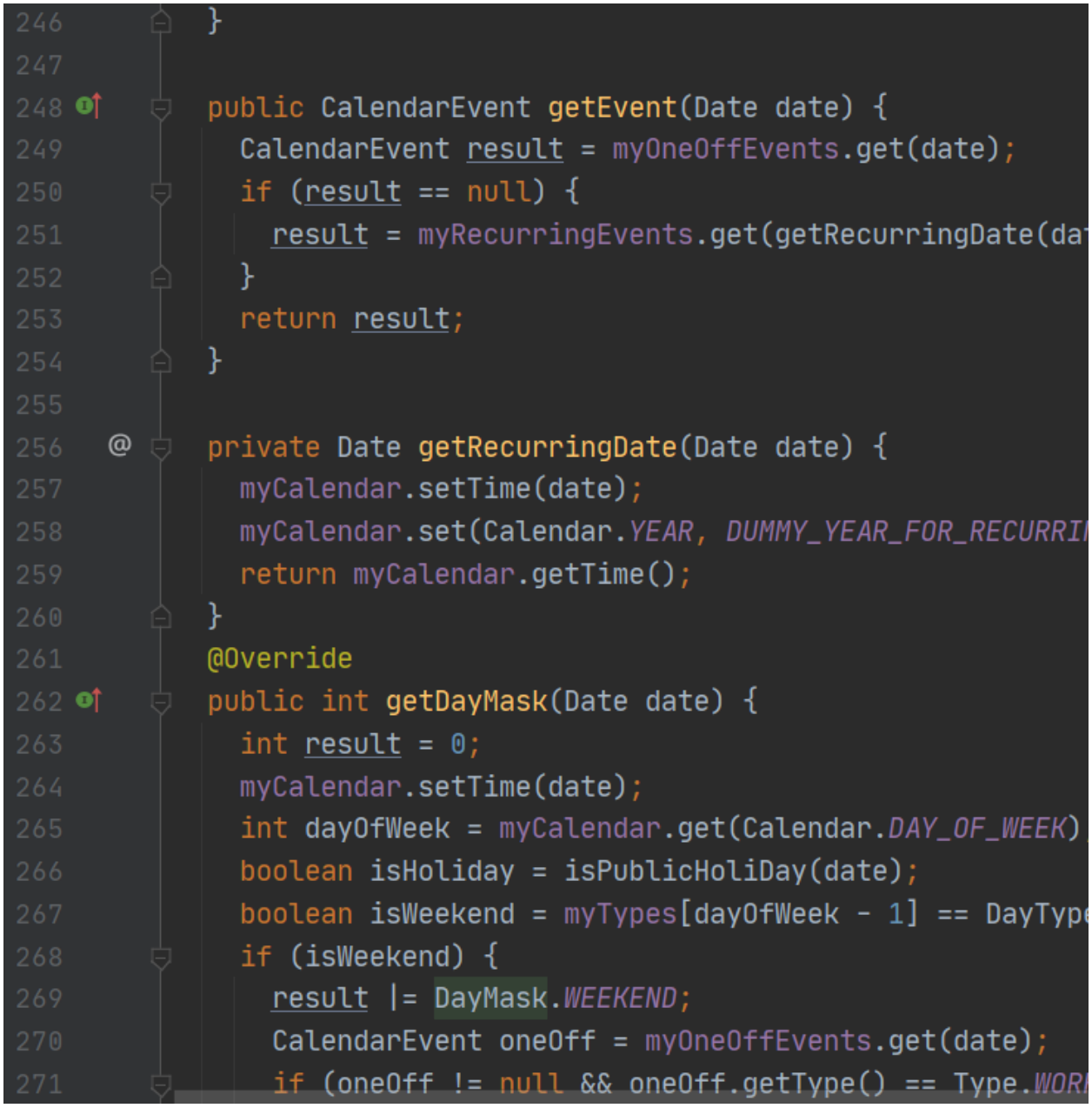
```
@Override
public Component getComponent() {...}
```

Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/gui/FileChooserPageBase.java
Texto: O método getComponent() é muito longo (90 linhas).
Solução: Devia ser criado métodos auxiliares.
Autor: João Oliveira
Revisão: Também acho que isto pode ser considerado um code smell **by** James Hertz

No comment

```java
/.../
package net.sourceforge.ganttproject.action;

import ...

public class BaselineDialogAction extends GPAction {
  private final IGanttProject myProject;
  private final UIFacade myUiFacade;
  private List<GanttPreviousState> myBaselines;
  private List<GanttPreviousState> myTrash = new ArrayList<~>();

  public BaselineDialogAction(IGanttProject project, UIFacade uiFacade) {
    super( name: "baseline.dialog");
    myProject = project;
    myUiFacade = uiFacade;
  }

  @Override
  public void actionPerformed(ActionEvent arg0) {
    myBaselines = new ArrayList<GanttPreviousState>(myProject.getBaselines());

    final EditableList<GanttPreviousState> list = new EditableList<~>(myBaselines,
        Collections.<GanttPreviousState> emptyList()) {

      @Override
      protected GanttPreviousState updateValue(GanttPreviousState newValue, GanttPreviousState curValue) {
        curValue.setName(newValue.getName());
        return curValue;
      }

      @Override
      protected GanttPreviousState createValue(GanttPreviousState prototype) {
        try {
          prototype.init();
          prototype.saveFile();
          return prototype;
        } catch (IOException e) {
          myUiFacade.showErrorDialog(e);
          return null;
        }
      }
```

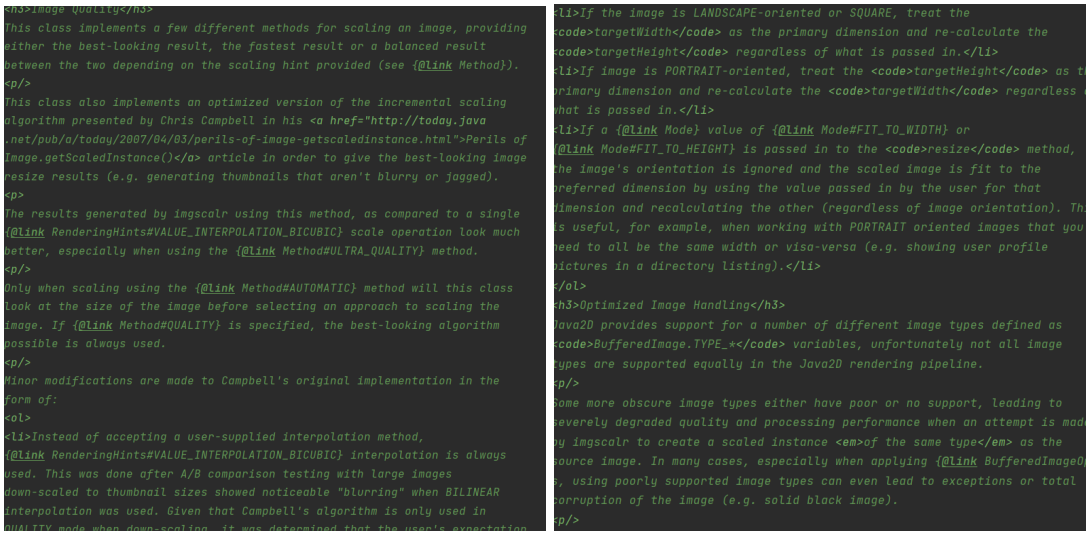
(a classe é grande por isso apresenta-se apenas uma parte aqui)
Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/action/BaselineDialogAction.java
Texto: A classe BaselineDialogAction sem comentários apesar da sua complexidade.
Solução:
Autor: Francisco Vasco
Revisão: A classe não tem nenhum comentário. Então eu concordo que isso é um code smell **by** James Hertz



```
246     }
247
248     public CalendarEvent getEvent(Date date) {
249       CalendarEvent result = myOneOffEvents.get(date);
250       if (result == null) {
251         result = myRecurringEvents.get(getRecurringDate(da
252       }
253       return result;
254     }
255
256     private Date getRecurringDate(Date date) {
257       myCalendar.setTime(date);
258       myCalendar.set(Calendar.YEAR, DUMMY_YEAR_FOR_RECURRI
259       return myCalendar.getTime();
260     }
261     @Override
262     public int getDayMask(Date date) {
263       int result = 0;
264       myCalendar.setTime(date);
265       int dayOfWeek = myCalendar.get(Calendar.DAY_OF_WEEK)
266       boolean isHoliday = isPublicHoliDay(date);
267       boolean isWeekend = myTypes[dayOfWeek - 1] == DayType
268       if (isWeekend) {
269         result |= DayMask.WEEKEND;
270         CalendarEvent oneOff = myOneOffEvents.get(date);
271         if (oneOff != null && oneOff.getType() == Type.WORK
```

Ficheiro:biz.ganttproject.core/src/main/java/biz/ganttproject/core/calendar/WeekendCalendarImpl.java
Texto: Essa classe não tem comentários.
Solução: Meter comentários
Autor: Iago Paulo
Revisão: Sim, para uma class de 380 linhas tem muitos poucos comentários **by** Ricardo Gonçalo

Over comment

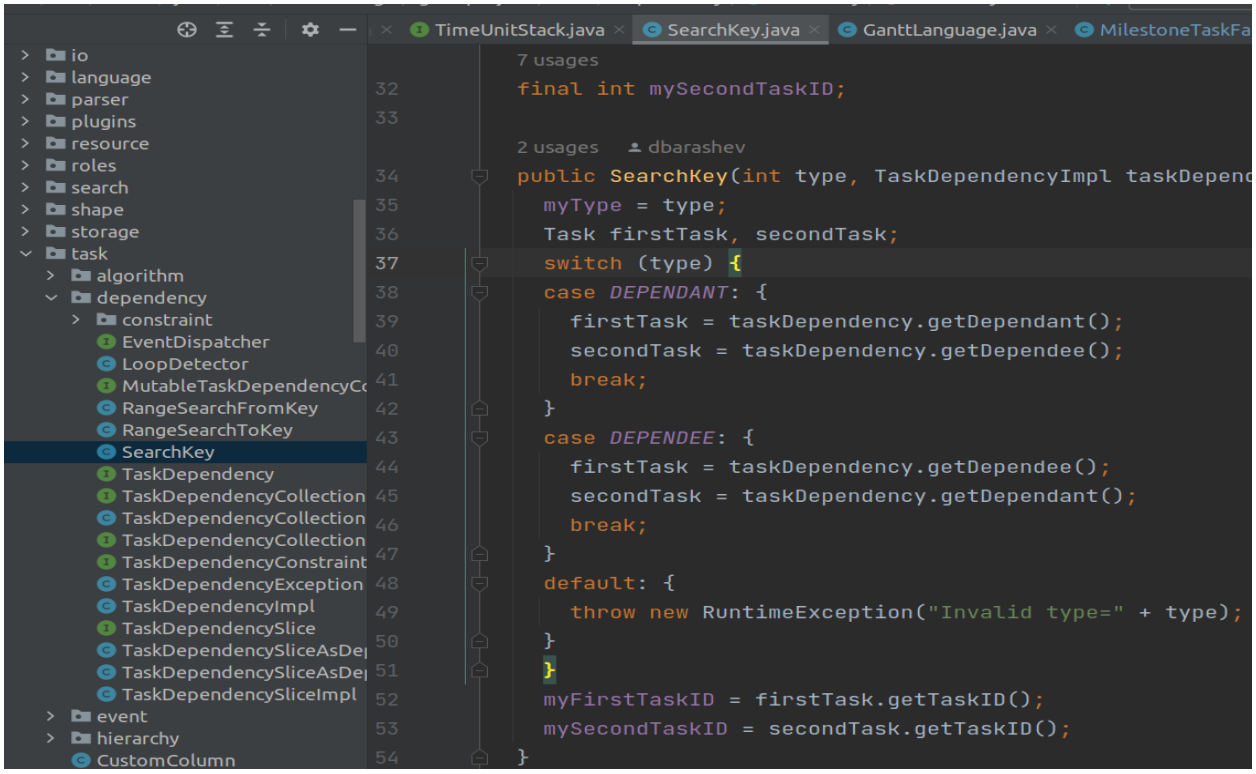


```
<h3>Image Quality</h3>
This class implements a few different methods for scaling an image, providing
either the best-looking result, the fastest result or a balanced result
between the two depending on the scaling hint provided (see {@link Method}).
<p/>
This class also implements an optimized version of the incremental scaling
algorithm presented by Chris Campbell in his <a href="http://today.java
.net/pub/a/today/2007/04/03/perils-of-image-getscaledinstance.html">Perils of
Image.getScaledInstance()</a> article in order to give the best-looking image
resize results (e.g. generating thumbnails that aren't blurry or jagged).
<p>
The results generated by imgscalr using this method, as compared to a single
{@link RenderingHints#VALUE_INTERPOLATION_BICUBIC} scale operation look much
better, especially when using the {@link Method#ULTRA_QUALITY} method.
<p/>
Only when scaling using the {@link Method#AUTOMATIC} method will this class
look at the size of the image before selecting an approach to scaling the
image. If {@link Method#QUALITY} is specified, the best-looking algorithm
possible is always used.
<p/>
Minor modifications are made to Campbell's original implementation in the
form of:
<ol>
<li>Instead of accepting a user-supplied interpolation method,
{@link RenderingHints#VALUE_INTERPOLATION_BICUBIC} interpolation is always
used. This was done after A/B comparison testing with large images
down-scaled to thumbnail sizes showed noticeable "blurring" when BILINEAR
interpolation was used. Given that Campbell's algorithm is only used in
QUALITY mode when down-scaling, it was determined that the user's expectation
```

```
<li>If the image is LANDSCAPE-oriented or SQUARE, treat the
<code>targetWidth</code> as the primary dimension and re-calculate the
<code>targetHeight</code> regardless of what is passed in.</li>
<li>If image is PORTRAIT-oriented, treat the <code>targetHeight</code> as t
primary dimension and re-calculate the <code>targetWidth</code> regardless
what is passed in.</li>
<li>If a {@link Mode} value of {@link Mode#FIT_TO_WIDTH} or
{@link Mode#FIT_TO_HEIGHT} is passed in to the <code>resize</code> method,
the image's orientation is ignored and the scaled image is fit to the
preferred dimension by using the value passed in by the user for that
dimension and recalculating the other (regardless of image orientation). Th
is useful, for example, when working with PORTRAIT oriented images that you
need to all be the same width or visa-versa (e.g. showing user profile
pictures in a directory listing).</li>
</ol>
<h3>Optimized Image Handling</h3>
Java2D provides support for a number of different image types defined as
<code>BufferedImage.TYPE_*</code> variables, unfortunately not all image
types are supported equally in the Java2D rendering pipeline.
<p/>
Some more obscure image types either have poor or no support, leading to
severely degraded quality and processing performance when an attempt is mad
by imgscalr to create a scaled instance <em>of the same type</em> as the
source image. In many cases, especially when applying {@link BufferedImageO
s, using poorly supported image types can even lead to exceptions or total
corruption of the image (e.g. solid black image).
<p/>
```

Ficheiro: ganttproject/src/main/java/org/imgscalr/Scalr.java
Texto: Tem comentários muito grandes. Por exemplo o que está logo em cima da classe tem mais de 130 linhas.
Solução: Simplificar estes comentários, há mesmo necessidade de explicações tão extensivas para o funcionamento desta classe? Claro que não, muita informação desnecessária
Autor: Ricardo Gonçalo
Revisão: Ao contrário do restante do código (que quase não tem comentários) essa classe tem comentários e são muitos, assim sendo, claramente é um code smell **by** James Furtado

Switch statement



Ficheiro:ganttproject/src/main/java/net/sourceforge/ganttproject/task/dependency/SearchKey.java
Texto: A classe tem um atributo myType que é um inteiro. As constantes DEPENDEE e DEPENDANT são usadas como tipos. E tem um switch statement no construtor para os dois casos e uma exceção caso contrário.
Solução: Fazer duas subclasses. Além de eliminar o atributo myType vai eliminar a exceção o que já é muito bom.
Autor: James Furtado
Revisão: Sim, no mínimo substituir por um Enum **by** Ricardo Gonçalo