Software Engineering Project Report

GanttProject

Members:

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

Table of Contents

Phase 1 Sprint 1

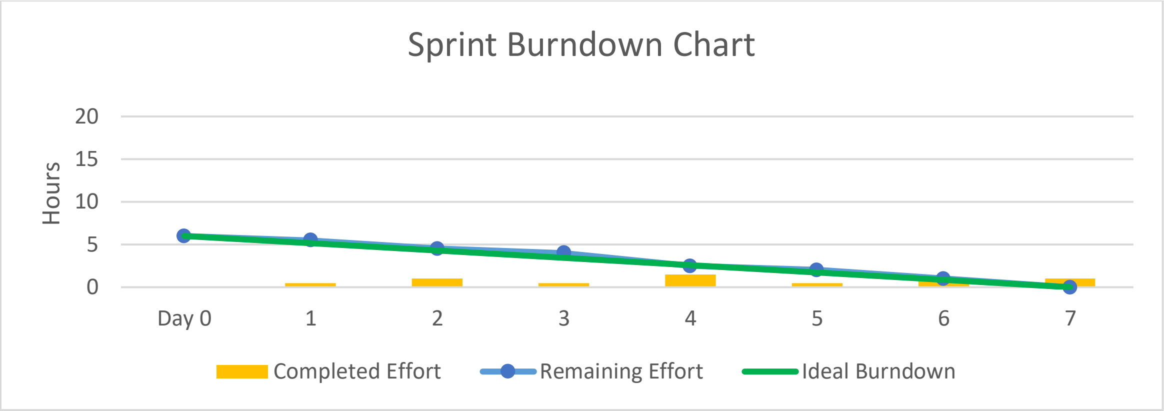
In this sprint we basically did the assignment of the first phase and the **scrum master** was João Oliveira.

Sprint backlog

Todo	In Progress	Reviewing	Done
			Identificar design patterns
			Identificar code smells
			Rever design patterns
			Rever code smells
			Corrigir eventuais erros

Burndown chart

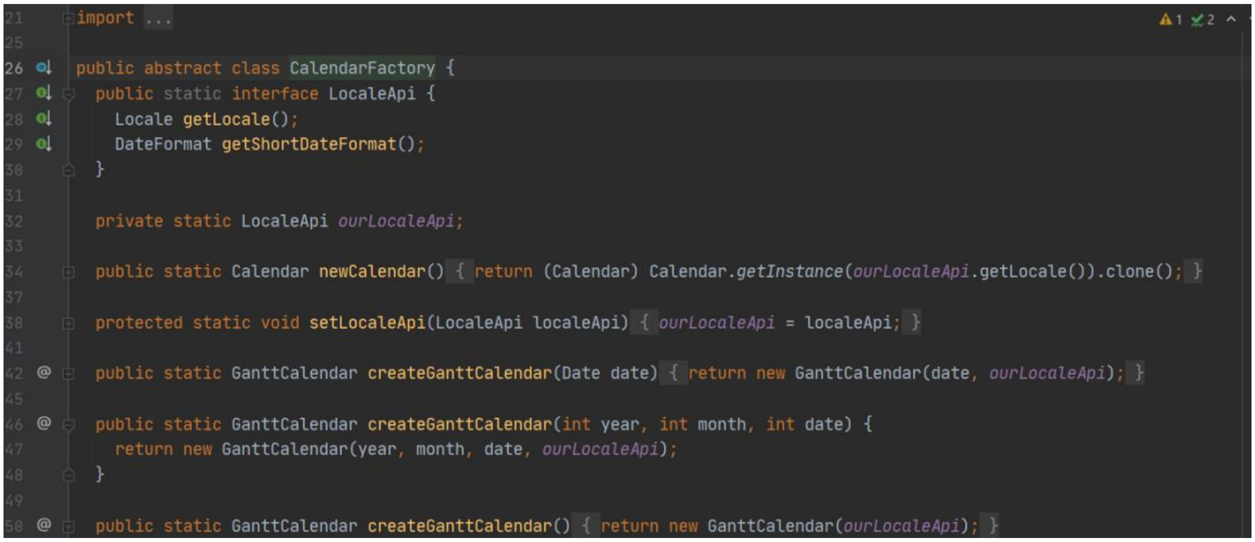
Task ID	Task Description	Initial Estimate	15/Oct/22	16/Oct/22	17/Oct/22	18/Oct/22	19/Oct/22	20/Oct/22	21/Oct/22
		Day 0	1	2	3	4	5	6	7
1	Identificar design patterns	1	0.5	0.5					
2	Identificar code smells	1		0.5	0.5				
3	Rever design patterns	1				1			
4	Rever code smells	2				0.5	0.5		1
5	Corrigir eventuais erros	1						1	
Completed Effort		0	0.5	1	0.5	1.5	0.5	1	1
Remaining Effort		6	5.5	4.5	4.0	2.5	2.0	1.0	0.0
Ideal Burndown		6	5.1	4.3	3.4	2.6	1.7	0.9	0.0


Sprint Burndown Chart
Hours
20
15
10
5
0
Day 0
1
2
3
4
5
6
7
Completed Effort
Remaining Effort
Ideal Burndown

Produced content

Design Patterns

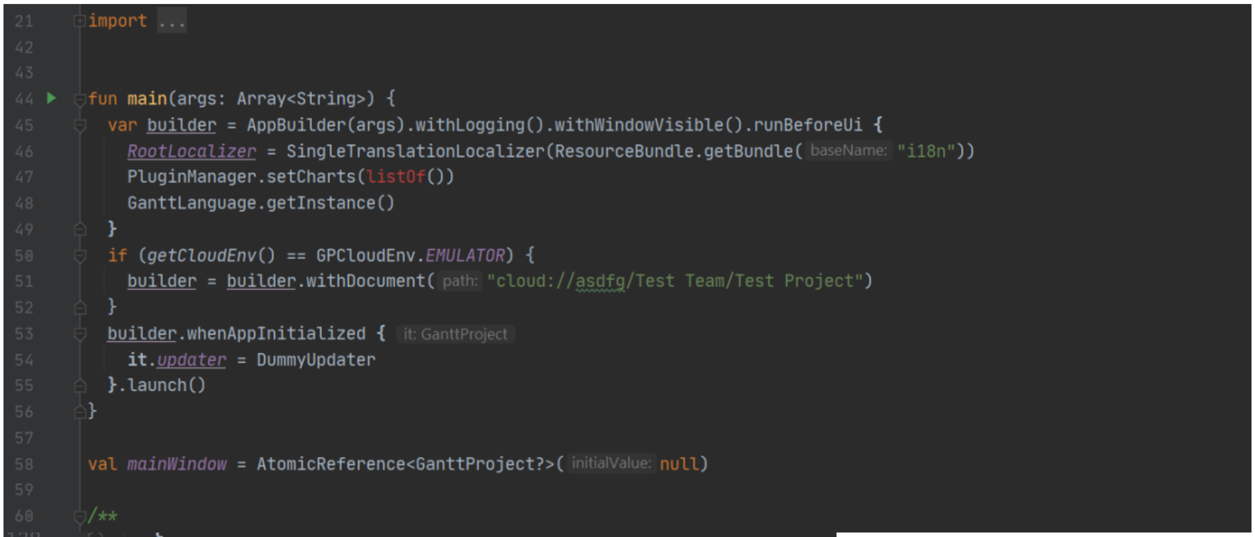
Abstract Factory Pattern



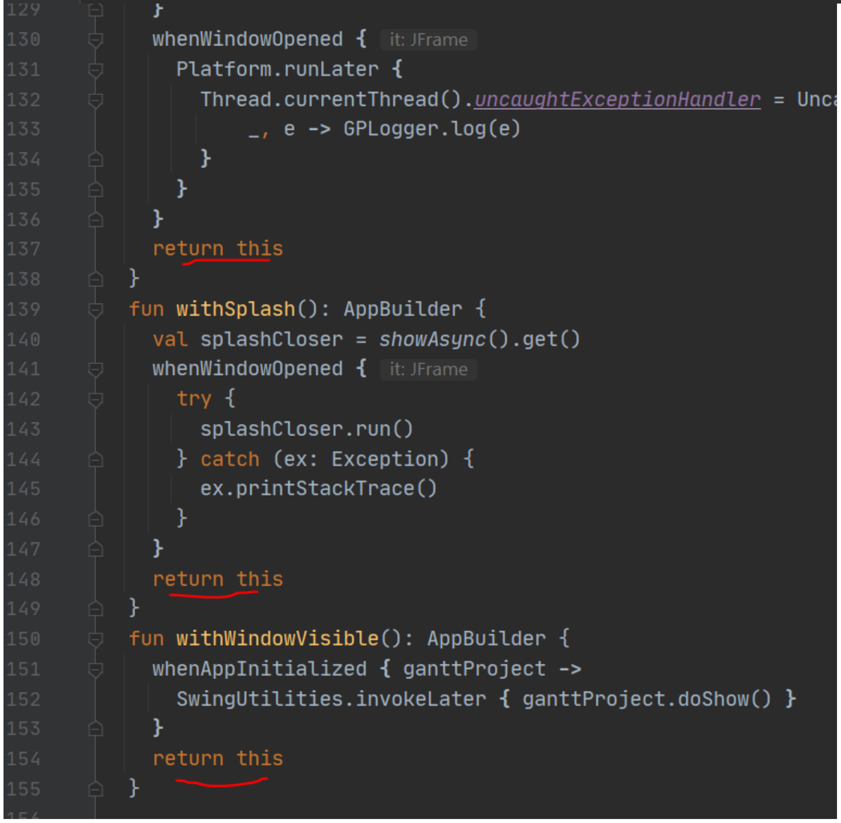
```
import ...

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); }
```

File:biz.ganttproject.core\src\main\java\biz\ganttproject\core\time\CalendarFactory
Text: Allows the creation of different GanttCalendars depending on the parameters used or a new Calendar instance.
Author: Iago Paulo
Review: Yes this is a factory pattern since the constructor was altered for the methods createGanttCalendar that receive a number of parameters and return a product **by** João Oliveira

Builder Pattern

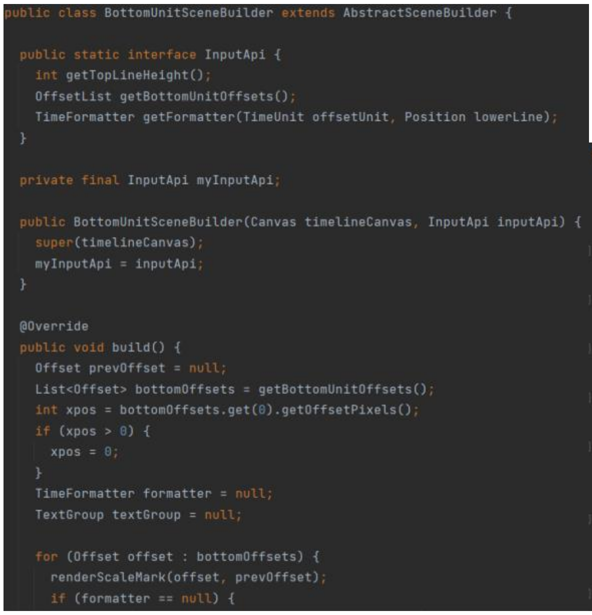


```
import ...


fun main(args: Array<String>) {
  var builder = AppBuilder(args).withLogging().withWindowVisible().runBeforeUi {
    RootLocalizer = SingleTranslationLocalizer(ResourceBundle.getBundle( baseName: "i18n"))
    PluginManager.setCharts(listOf())
    GanttLanguage.getInstance()
  }
  if (getCloudEnv() == GPCloudEnv.EMULATOR) {
    builder = builder.withDocument( path: "cloud://asdfg/Test Team/Test Project")
  }
  builder.whenAppInitialized {  it: GanttProject
    it.updater = DummyUpdater
  }.launch()
}

val mainWindow = AtomicReference<GanttProject?>( initialValue: null)

/**
```

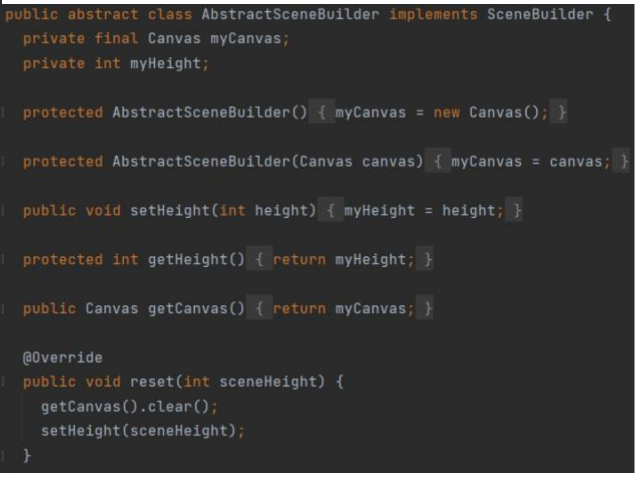


```
    }
    whenWindowOpened {  it: JFrame
      Platform.runLater {
        Thread.currentThread().uncaughtExceptionHandler = Unca
            _, e -> GPLogger.log(e)
        }
      }
    }
    return this
  }
  fun withSplash(): AppBuilder {
    val splashCloser = showAsync().get()
    whenWindowOpened {  it: JFrame
      try {
        splashCloser.run()
      } catch (ex: Exception) {
        ex.printStackTrace()
      }
    }
    return this
  }
  fun withWindowVisible(): AppBuilder {
    whenAppInitialized { ganttProject ->
      SwingUtilities.invokeLater { ganttProject.doShow() }
    }
    return this
  }
```

File:ganttproject\ganttproject\src\main\java\net\sourceforge\ganttproject\App.kt
Text: This is a builder where each method returns the builder itself, allows step-by-step additions to the the app and finally launches the program.
Author: Iago Paulo
Review: Agreed, it provides the necessary methods to build the app **by** Ricardo Gonçalo



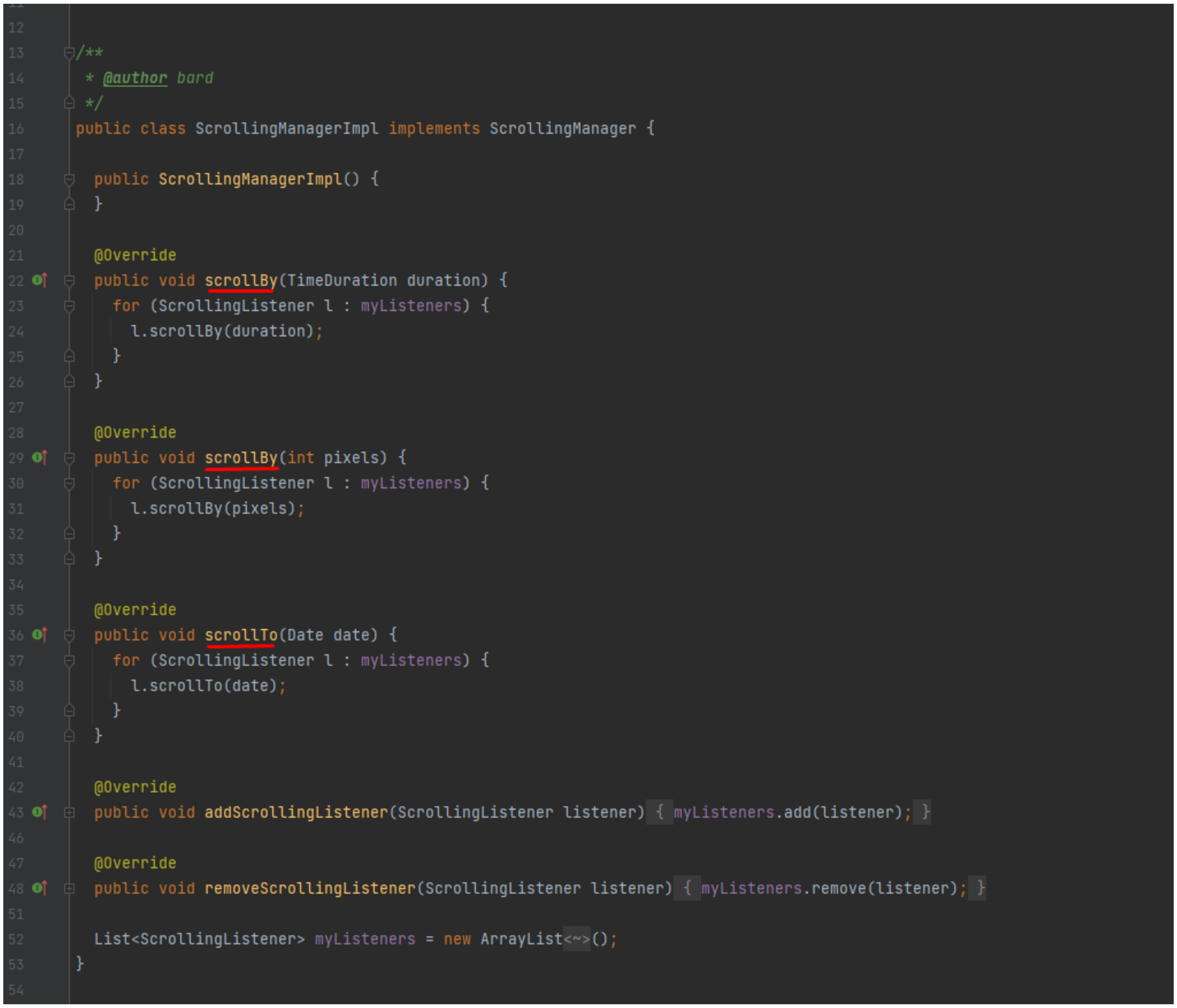
```java
public class BottomUnitSceneBuilder extends AbstractSceneBuilder {

  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);
  }
```

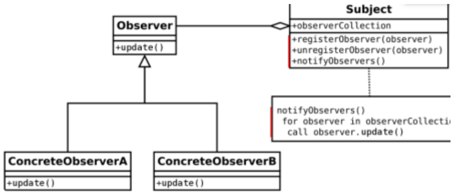
File:biz.ganttproject.core/src/main/java/biz/ganttproject/core/chart/scene/AbstractSceneBuilder.java
Text: Build pattern in AbstractSceneBuilder and SceneBuilder,which are the foundation for classes like ChartRendererBase and BottomUnitSceneBuilder, builders for Canvas
Author: Ricardo Gonçalo
Review: Has the characteristics of a standard builder pattern and appears to be working as such **by** Iago Paulo

Command Pattern

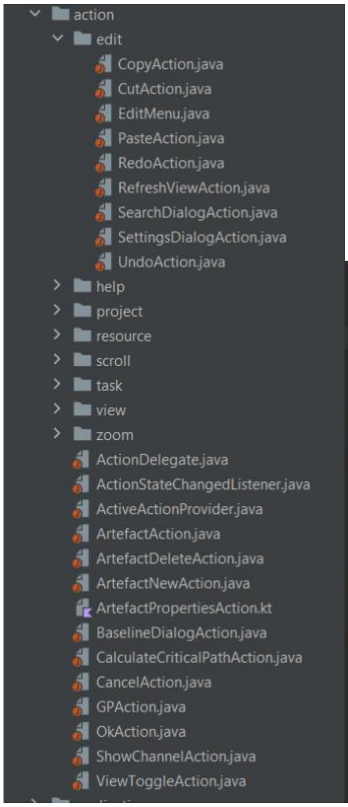


```java
/**
 * @author bard
 */
public class ScrollingManagerImpl implements ScrollingManager {

  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<~>();
}
```

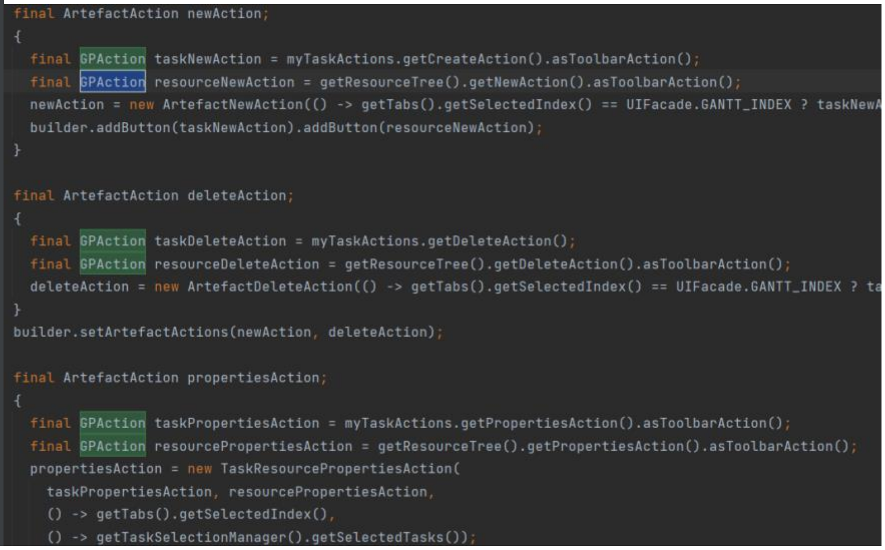
File:ganttproject/src/main/java/net/sourceforge/ganttproject/gui/scrolling/ ScrollingManagerImpl.java
Text: The class Scrolling ManagerImpl serves as a command manager for "scrolling".
Author: Francisco Vasco
Review: Francisco says it's a Command Pattern although to me it looks more like an Observer Pattern due to the similarities to what we saw in the practical lessons:



by Iago Paulo



```
action
  edit
    CopyAction.java
    CutAction.java
    EditMenu.java
    PasteAction.java
    RedoAction.java
    RefreshViewAction.java
    SearchDialogAction.java
    SettingsDialogAction.java
    UndoAction.java
  help
  project
  resource
  scroll
  task
  view
  zoom
  ActionDelegate.java
  ActionStateChangedListener.java
  ActiveActionProvider.java
  ArtefactAction.java
  ArtefactDeleteAction.java
  ArtefactNewAction.java
  ArtefactPropertiesAction.kt
  BaselineDialogAction.java
  CalculateCriticalPathAction.java
  CancelAction.java
  GPAction.java
  OkAction.java
  ShowChannelAction.java
  ViewToggleAction.java
```



```
final ArtefactAction newAction;
{
  final GPAction taskNewAction = myTaskActions.getCreateAction().asToolbarAction();
  final GPAction resourceNewAction = getResourceTree().getNewAction().asToolbarAction();
  newAction = new ArtefactNewAction(() -> getTabs().getSelectedIndex() == UIFacade.GANTT_INDEX ? taskNewA
  builder.addButton(taskNewAction).addButton(resourceNewAction);
}

final ArtefactAction deleteAction;
{
  final GPAction taskDeleteAction = myTaskActions.getDeleteAction();
  final GPAction resourceDeleteAction = getResourceTree().getDeleteAction().asToolbarAction();
  deleteAction = new ArtefactDeleteAction(() -> getTabs().getSelectedIndex() == UIFacade.GANTT_INDEX ? ta
}
builder.setArtefactActions(newAction, deleteAction);

final ArtefactAction propertiesAction;
{
  final GPAction taskPropertiesAction = myTaskActions.getPropertiesAction().asToolbarAction();
  final GPAction resourcePropertiesAction = getResourceTree().getPropertiesAction().asToolbarAction();
  propertiesAction = new TaskResourcePropertiesAction(
    taskPropertiesAction, resourcePropertiesAction,
    () -> getTabs().getSelectedIndex(),
    () -> getTaskSelectionManager().getSelectedTasks());
```

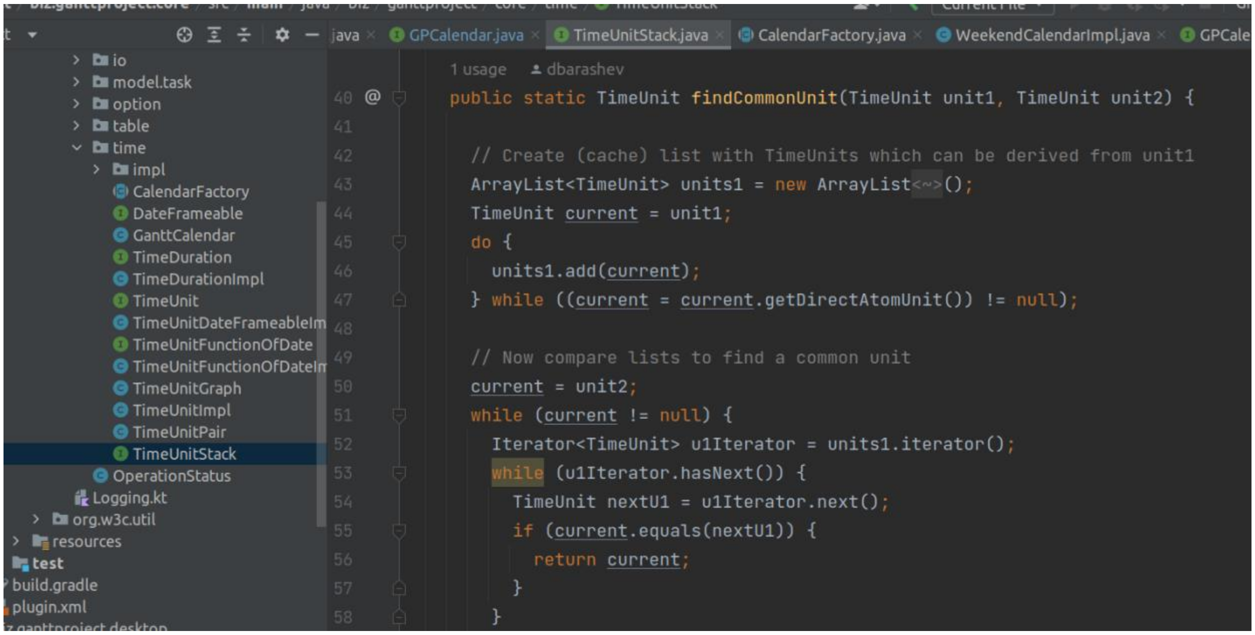
File: *ganttproject\src\main\java\net\sourceforge\ganttproject\action*

Text: This is a **Command** pattern, where the actions are classes instead of functions, verified for example in their usage in ganttproject/src/main/java/net/sourceforge/ganttproject/GanttProject.java

Author: Ricardo Gonçalo

Review: Several actions like "zooming", "scrolling", etc, are separated in classes, encapsulating the necessary information for executing each action **by** João Oliveira

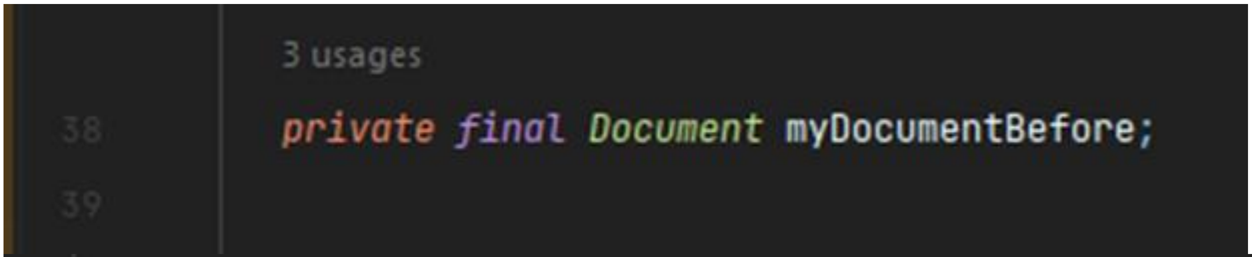
Iterator Pattern



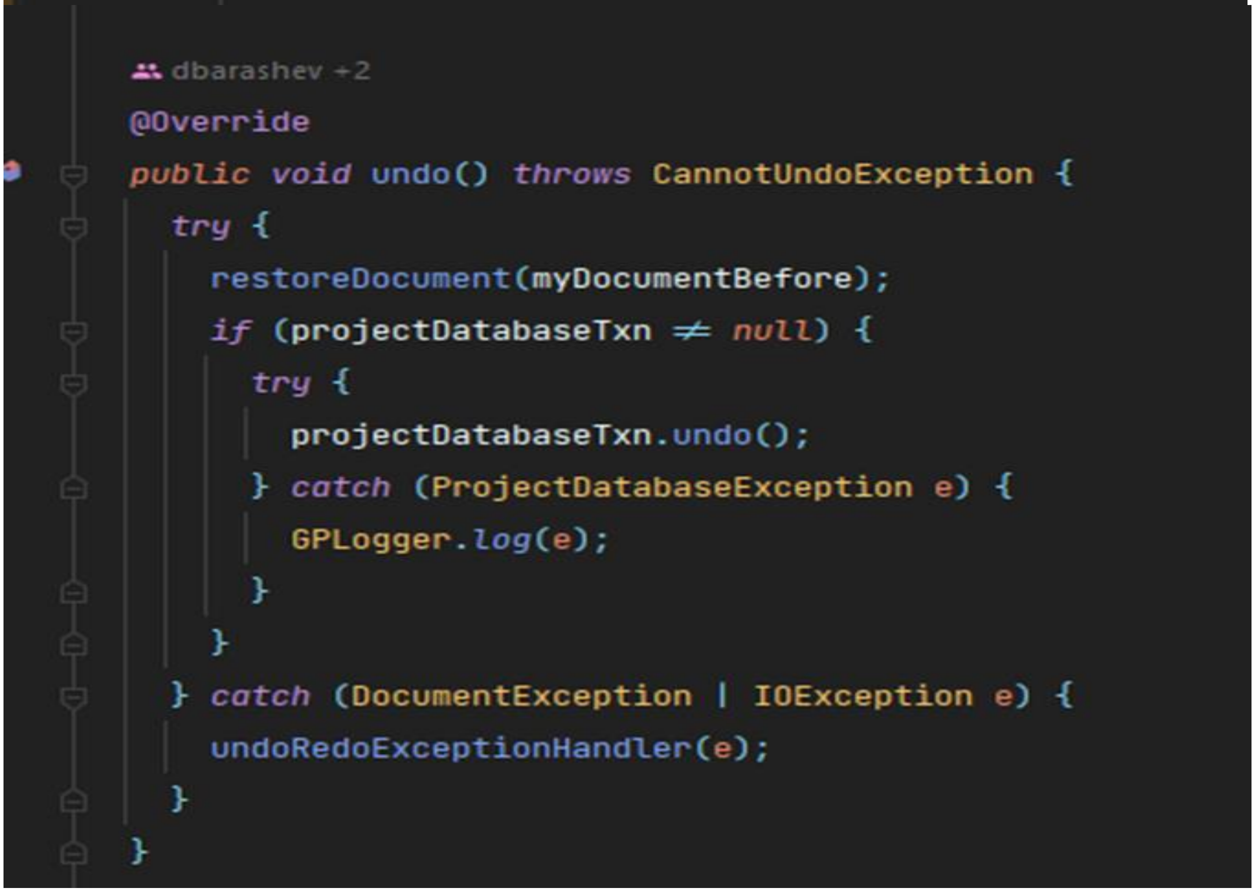
```
1 usage   dbarashev
public static TimeUnit findCommonUnit(TimeUnit unit1, TimeUnit unit2) {

  // Create (cache) list with TimeUnits which can be derived from unit1
  ArrayList<TimeUnit> units1 = new ArrayList<~>();
  TimeUnit current = unit1;
  do {
    units1.add(current);
  } while ((current = current.getDirectAtomUnit()) != null);

  // Now compare lists to find a common unit
  current = unit2;
  while (current != null) {
    Iterator<TimeUnit> u1Iterator = units1.iterator();
    while (u1Iterator.hasNext()) {
      TimeUnit nextU1 = u1Iterator.next();
      if (current.equals(nextU1)) {
        return current;
      }
    }
```

File:biz.ganttproject.core/src/main/java/biz/ganttproject/core/time/TimeUnitStack.java
Text: The Util class is an inner class for the TimeUnitStack Interface and uses the Iterator Pattern on the static method findCommonUnit.
Author: James Furtado
Review: The class does in fact have an iterator and uses it to access elements, so it is an Iterator Pattern **by** Iago Paulo

Memento Pattern



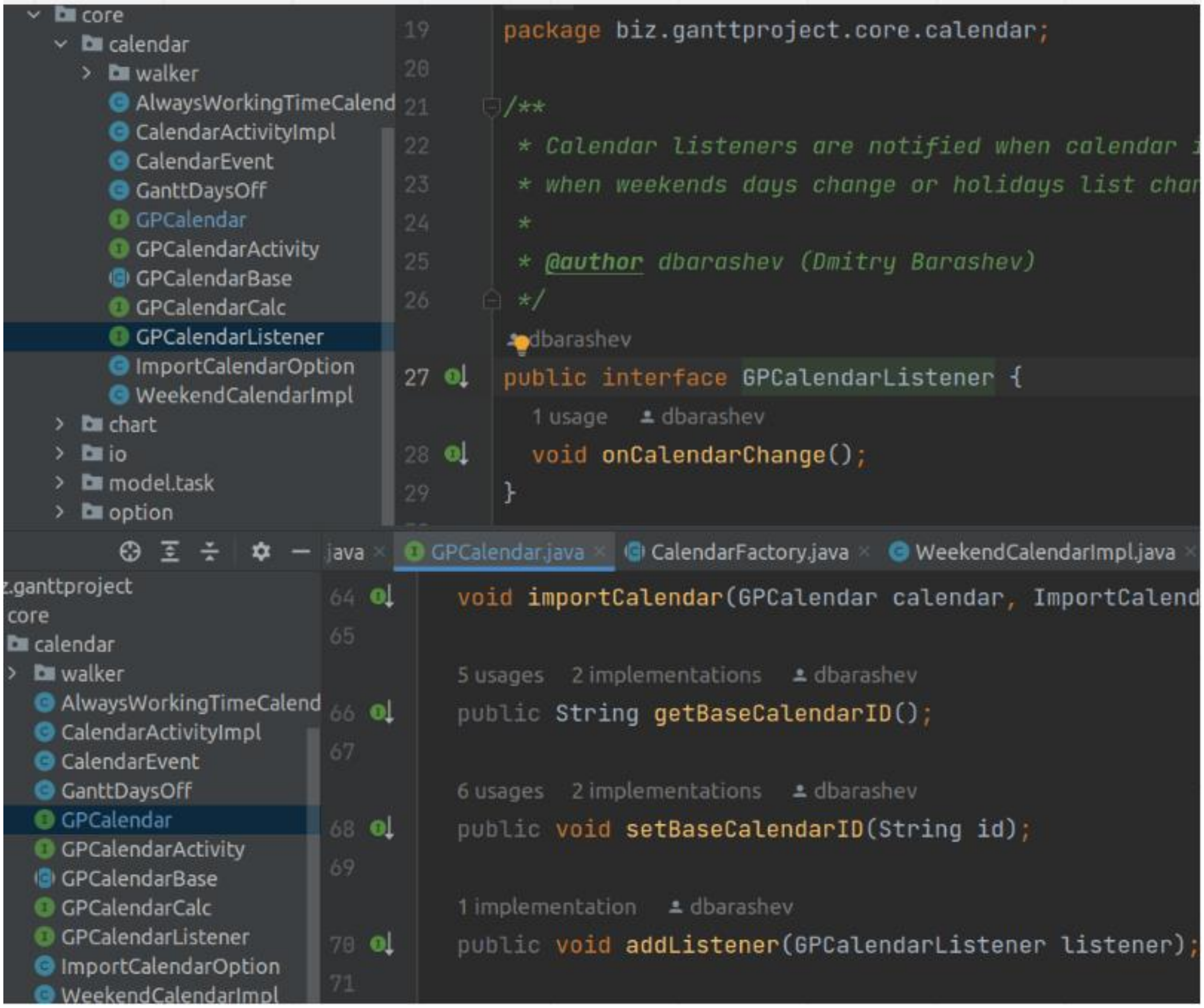
```
3 usages
private final Document myDocumentBefore;

```



```
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);
  }
}
```

File:ganttproject/src/main/java/net/sourceforge/ganttproject/undo/UndoableEditImpl.java
Text: Saves an object with the previous state of the document and has an undo() method that restores the document to it's previous state.
Author: João Oliveira
Review: Correct, it is possible to restore the previous document state with the undo() method by keeping a snapshot of the previous state **by** Ricardo Gonçalo

Observer Pattern

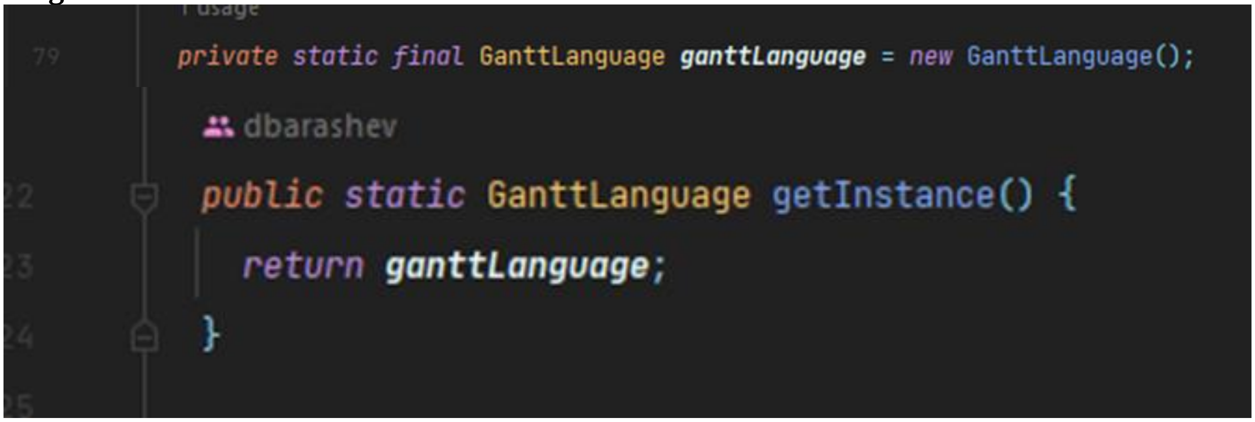


```java
package biz.ganttproject.core.calendar;

/**
 * Calendar listeners are notified when calendar i
 * when weekends days change or holidays list char
 *
 * @author dbarashev (Dmitry Barashev)
 */
public interface GPCalendarListener {
  void onCalendarChange();
}
```

```java
void importCalendar(GPCalendar calendar, ImportCalend

public String getBaseCalendarID();

public void setBaseCalendarID(String id);

public void addListener(GPCalendarListener listener);
```

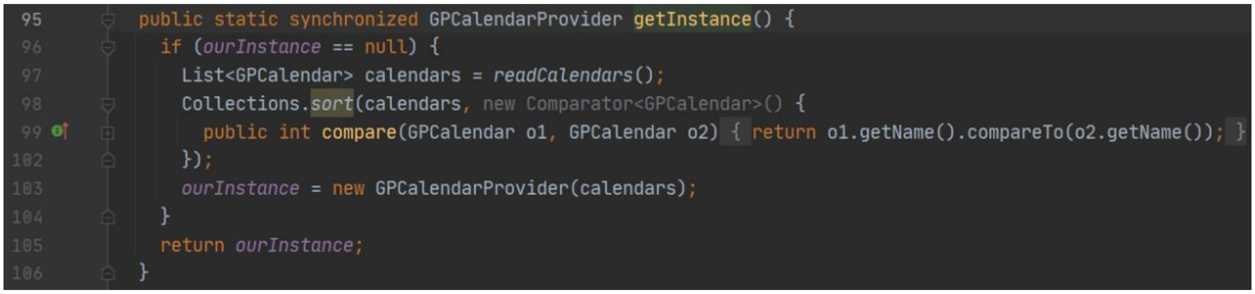
File:biz.ganttproject.core/src/main/java/biz/ganttproject/core/calendar/
Text: In the biz.ganttproject.core.calendar there’s an observer pattern where the Subject interface is GPCalendar and the Observer is the GPCalendarListener.
Author: James Furtado
Review: GPCalendar implementations (the Subject) keep a collection of GPCalendarListener (added through the addListener method) and notify each one when changing the calendar, the pattern seems correct **by** Francisco Vasco

Singleton



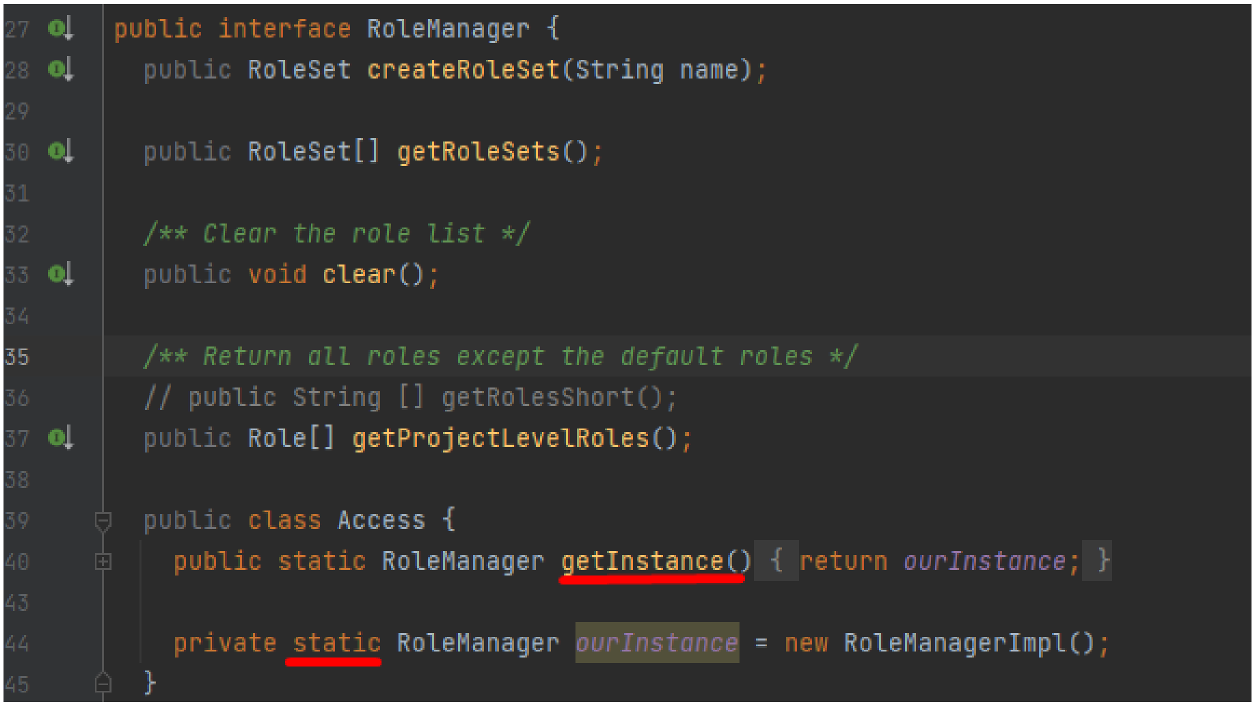
```
    1 usage
79  private static final GanttLanguage ganttLanguage = new GanttLanguage();

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

File:ganttproject/src/main/java/net/sourceforge/ganttproject/parser/AbstractTagHandler.javaganttproject/src/main/java/net/sourceforge/ganttproject/language/GanttLanguage.java
Text: Usage of the getInstance() method that returns a single object instance that is initialized only once through the usage of a static variable (line 79).
Author: João Oliveira
Review: Correct, standard Singleton Pattern implementation **by** Francisco Vasco



```
public static synchronized GPCalendarProvider getInstance() {
  if (ourInstance == null) {
    List<GPCalendar> calendars = readCalendars();
    Collections.sort(calendars, new Comparator<GPCalendar>() {
      public int compare(GPCalendar o1, GPCalendar o2) { return o1.getName().compareTo(o2.getName()); }
    });
    ourInstance = new GPCalendarProvider(calendars);
  }
  return ourInstance;
}
```

File:ganttproject\ganttproject\src\main\java\net\sourceforge\ganttproject\calendar\GPCalendarProvider.java
Text: Usage of the GetInstance method implemented as expected for a Singleton Pattern.
Author: Iago Paulo
Review: This pattern, using the ourInstance variable, does in fact assure only one GPCalendarProvider instance exists **by** Francisco Vasco



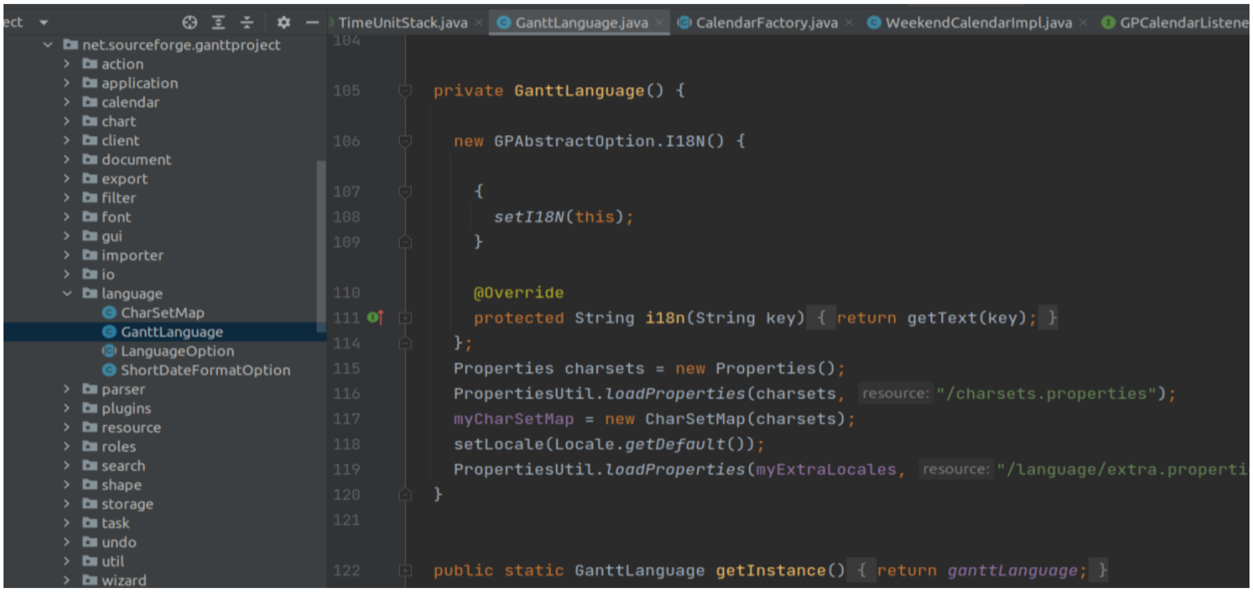
```
27   public interface RoleManager {
28     public RoleSet createRoleSet(String name);
29
30     public RoleSet[] getRoleSets();
31
32     /** Clear the role list */
33     public void clear();
34
35     /** Return all roles except the default roles */
36     // public String [] getRolesShort();
37     public Role[] getProjectLevelRoles();
38
39     public class Access {
40       public static RoleManager getInstance() { return ourInstance; }
43
44       private static RoleManager ourInstance = new RoleManagerImpl();
45     }
```

File:ganttproject/src/main/java/net/sourceforge/ganttproject/roles/RoleManager.java

Text: The Access class on lines 39-45 assures there’s only one instance of RoleManager.

Author: Francisco Vasco

Review: The RoleManager’s Access inner class does use the Singleton Pattern **by** James Furtado

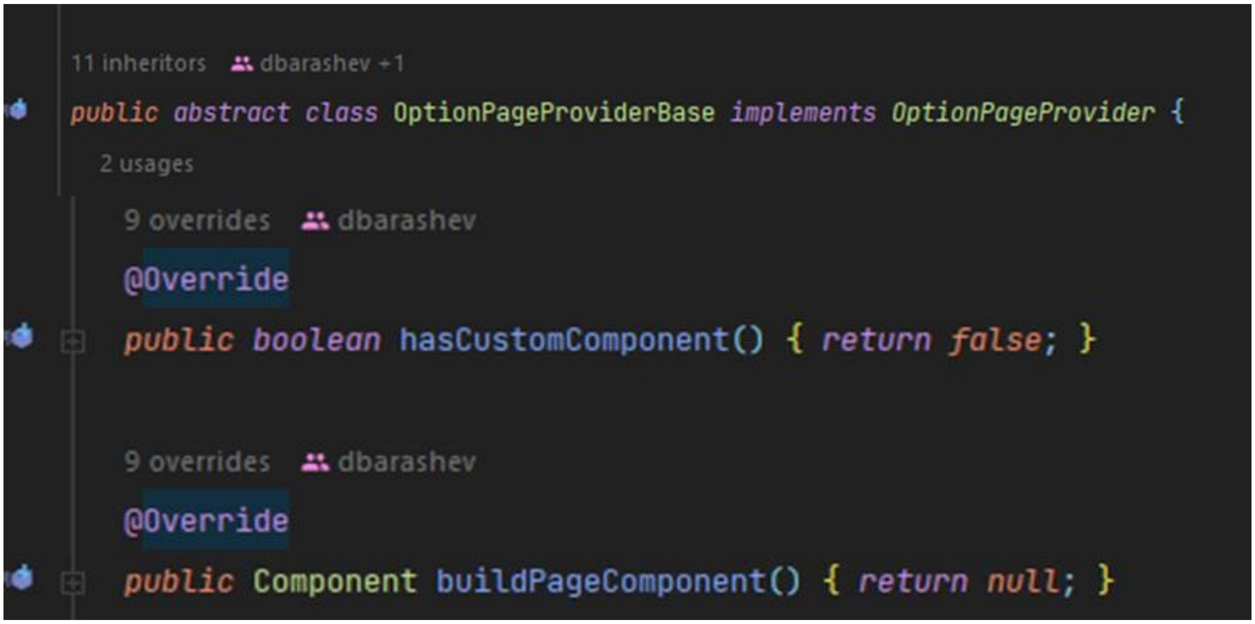


```
  private GanttLanguage() {

    new GPAbstractOption.I18N() {

      {
        setI18N(this);
      }

      @Override
      protected String i18n(String key) { return getText(key); }
    };
    Properties charsets = new Properties();
    PropertiesUtil.loadProperties(charsets,  resource: "/charsets.properties");
    myCharSetMap = new CharSetMap(charsets);
    setLocale(Locale.getDefault());
    PropertiesUtil.loadProperties(myExtraLocales,  resource: "/language/extra.properti
  }


  public static GanttLanguage getInstance() { return ganttLanguage; }
```

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

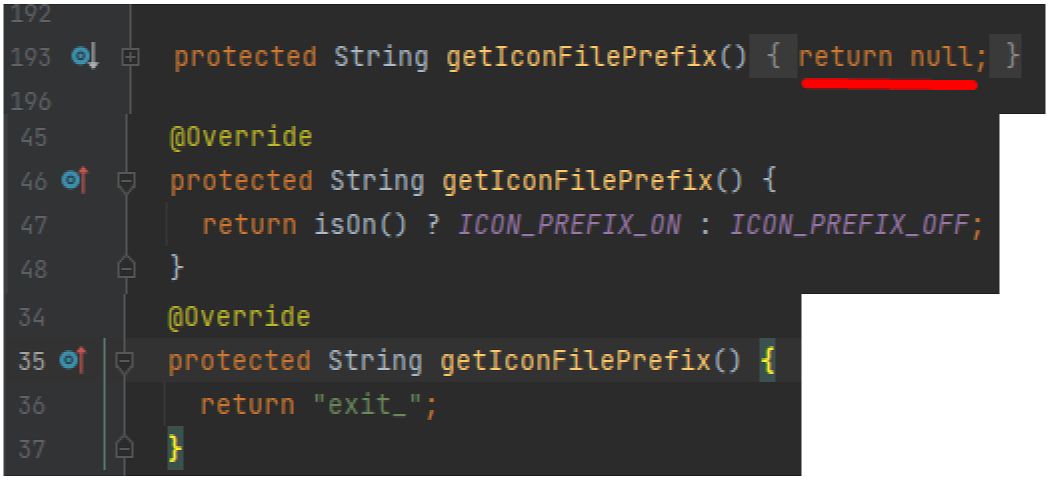
Text: The GanttLanguage class has a private constructor and a static method getInstance() that returns the single class instance.
Author: James Furtado
Review: It is in fact a singleton pattern **by** Ricardo Gonçalo

Template method



```
11 inheritors   dbarashev +1
public abstract class OptionPageProviderBase implements OptionPageProvider {
  2 usages
    9 overrides   dbarashev
    @Override
    public boolean hasCustomComponent() { return false; }

    9 overrides   dbarashev
    @Override
    public Component buildPageComponent() { return null; }
```

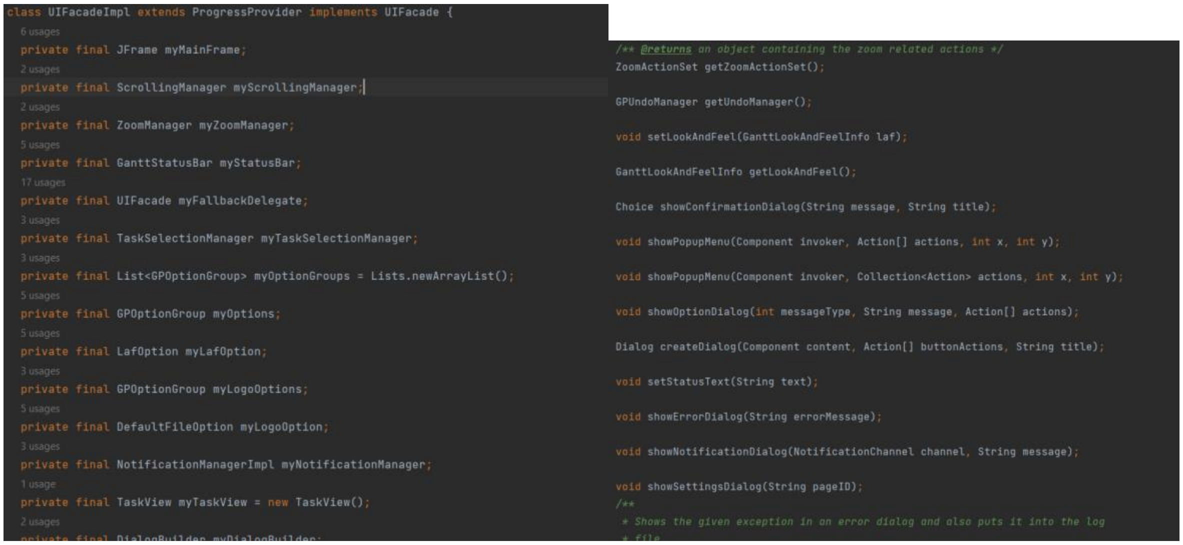
File:ganttproject/src/main/java/net/sourceforge/ganttproject/gui/options/OptionPageProviderBase.java
Text: Abstract class extended by 11 classes. 9 of them override the methods hasCustomComponent() and buildPageComponent()
Author: João Oliveira
Review: This is in fact a template method. The class is abstract and the subclasses override those and other methods **by** James Furtado



```java
protected String getIconFilePrefix() { return null; }

@Override
protected String getIconFilePrefix() {
  return isOn() ? ICON_PREFIX_ON : ICON_PREFIX_OFF;
}
@Override
protected String getIconFilePrefix() {
  return "exit_";
}
```

File:ganttproject/src/main/java/net/sourceforge/ganttproject/action/GPAction.java (linhas 138-151 e 193-195)
Text: The final method createIcon(String iconSize) in the abstract class GPAction calls the getIconFilePrefix() method whose concrete implementation varies in the classes that extend GPAction.
Author: Francisco Vasco
Review: This is a behavioral pattern where in the class there's a getIconFilePrefix() that is then changed by 19 classes that extend the class **by** João Oliveira

Facade pattern



```java
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;
```

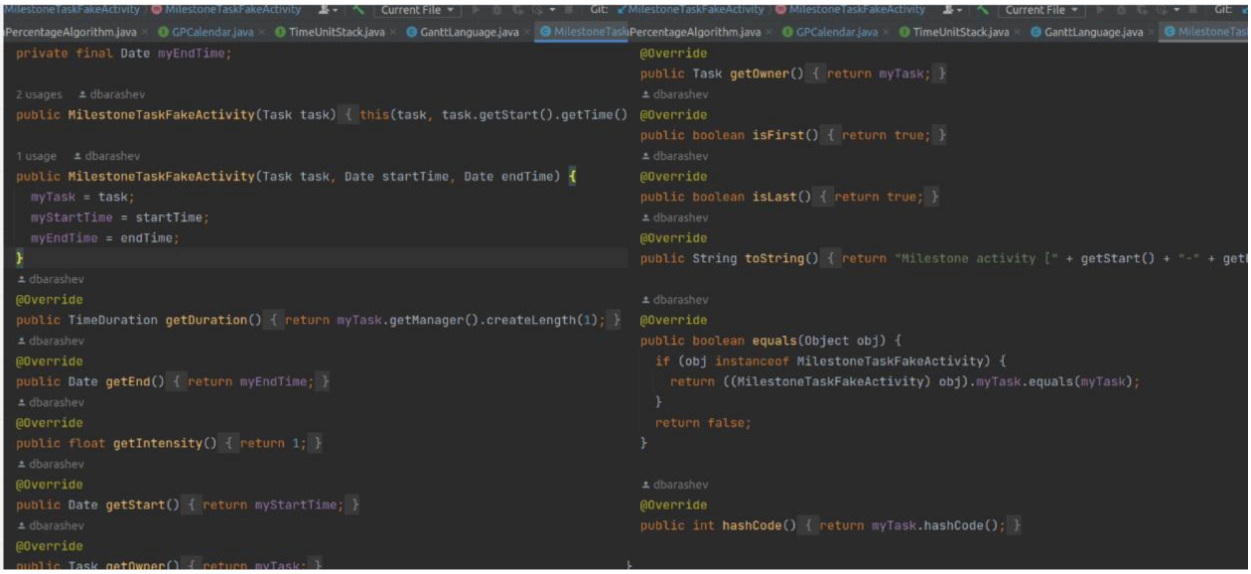
```java
/** @returns an object containing the zoom related actions */
ZoomActionSet getZoomActionSet();

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
```

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

Text: Represents a facade template that tries to hide the UI’s complexity.
Author: Ricardo Gonçalo
Review: The class name is very descriptive, I agree with my colleague **by** James Furtado

Code Smells

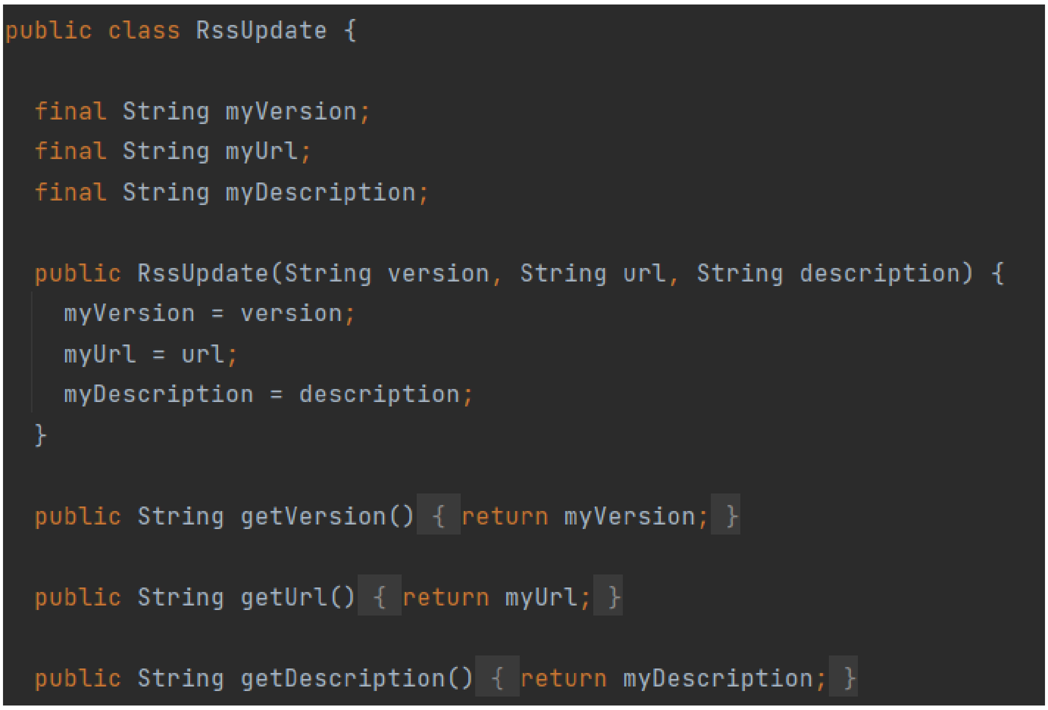
Data Class



```java
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; }
```

```java
@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(); }

}
```

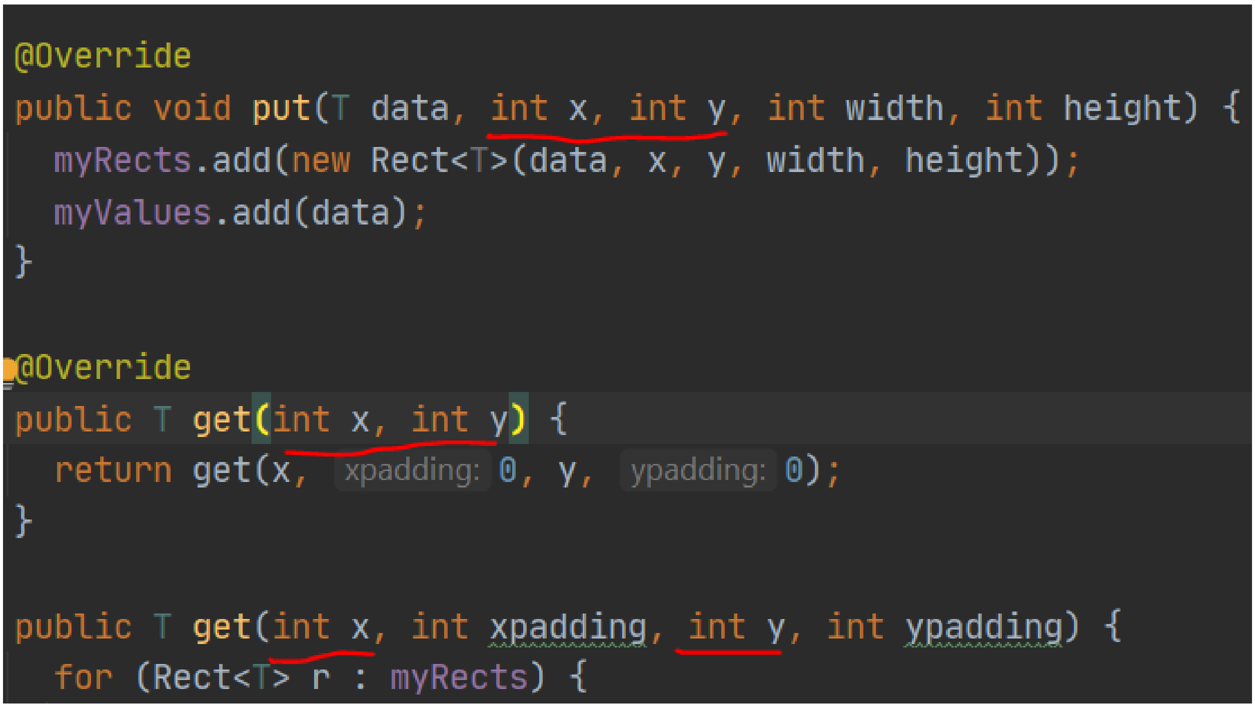
File:ganttproject/src/main/java/net/sourceforge/ganttproject/chart/MilestoneTaskFakeActivity.java
Text: The MilestoneTaskFakeActivity only has getters.
Solution: Add some functionality to the class.
Author: James Furtado
Review: Yes the class only has getters. I agree with the solution, another one would be moving the variables to the classes that actually use them **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; }
```

File: ganttproject/src/main/java/net/sourceforge/ganttproject/client/RssUpdate.java
Text: This is a Data Class, completely defined by the 3 primitives it has, it's isolated existence adds unnecessary complexity.
Solution: This class could be nested in the RssParser, since it's only used there.
Author: Ricardo Gonçalo
Review: Viable solution to reduce complexity **by** Iago Paulo

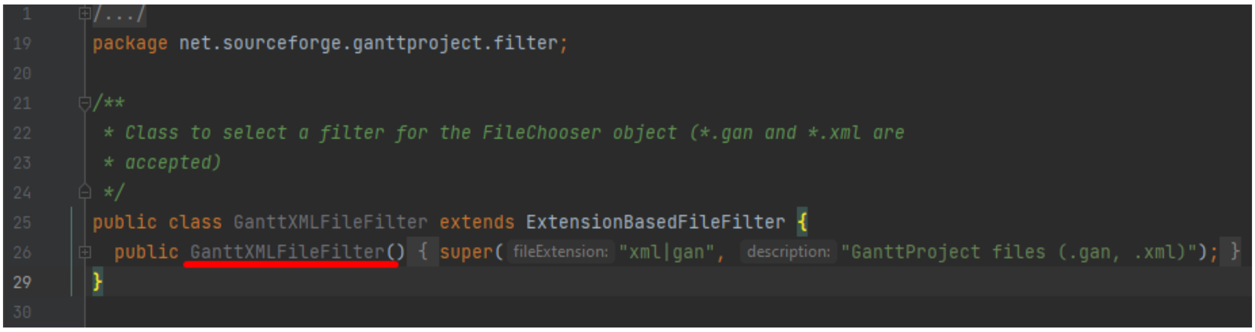
Data Clump



```
@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) {
```

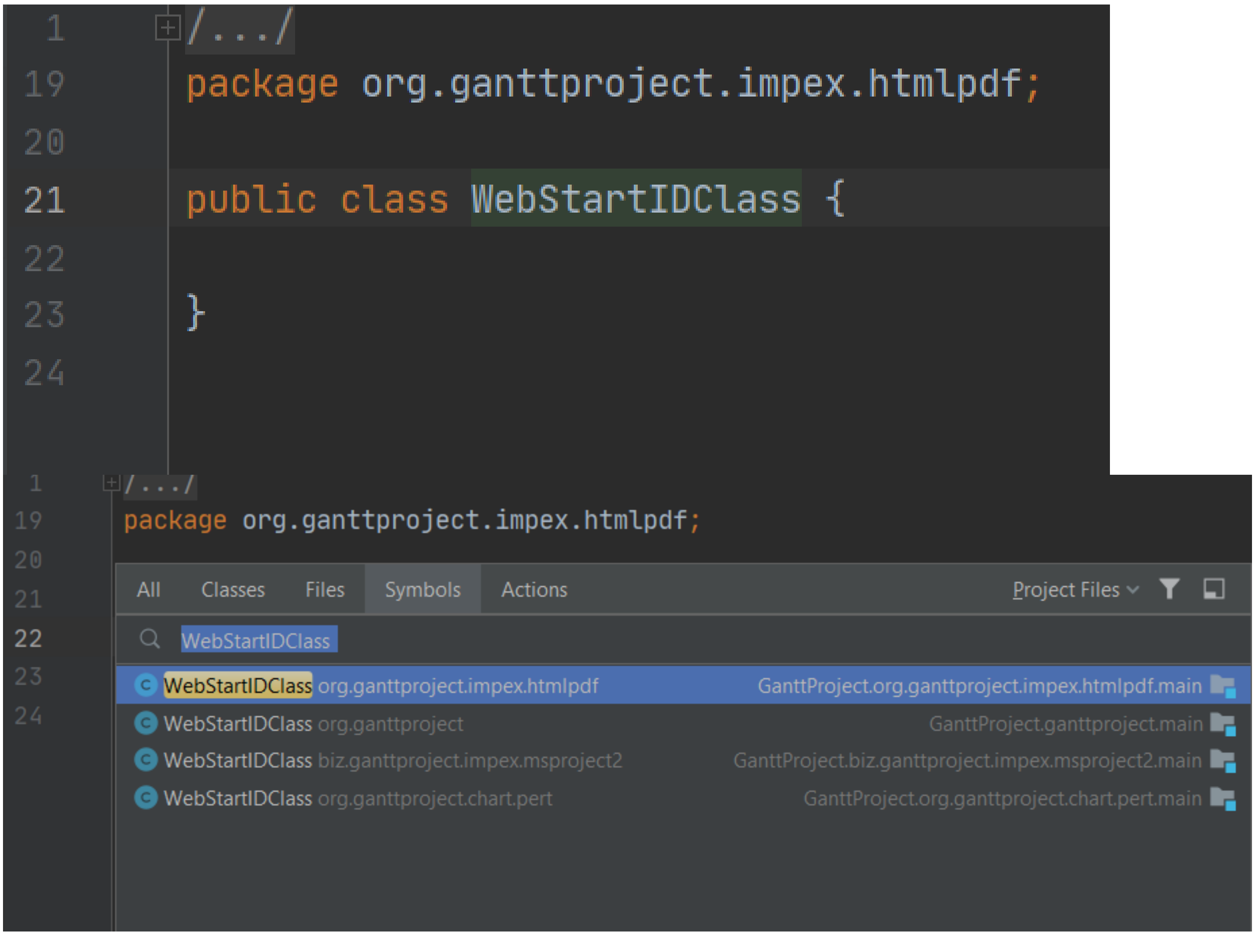
File: ganttproject-master\biz.ganttproject.core\src\main\java\biz\ganttproject\core\chart\canvas\DummySpatialIndex.java
Text: Long parameter list.
Solution: Create to group some of the arguments like x and y.
Author: Iago Paulo
Review: It is in fact a data clump **by** João Oliveira

DeadCode



```
/.../
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)"); }
}
```

File:ganttproject/src/main/java/net/sourceforge/ganttproject/filter/GanttXMLFileFilter.java
Text: The GanttXMLFileFilter class is never used.
Solution: Delete it.
Author: Francisco Vasco
Review: The GanttXMLFileFilter class is not being used so it can be safely deleted **by** João Oliveira



```
/.../
package org.ganttproject.impex.htmlpdf;

public class WebStartIDClass {

}
```

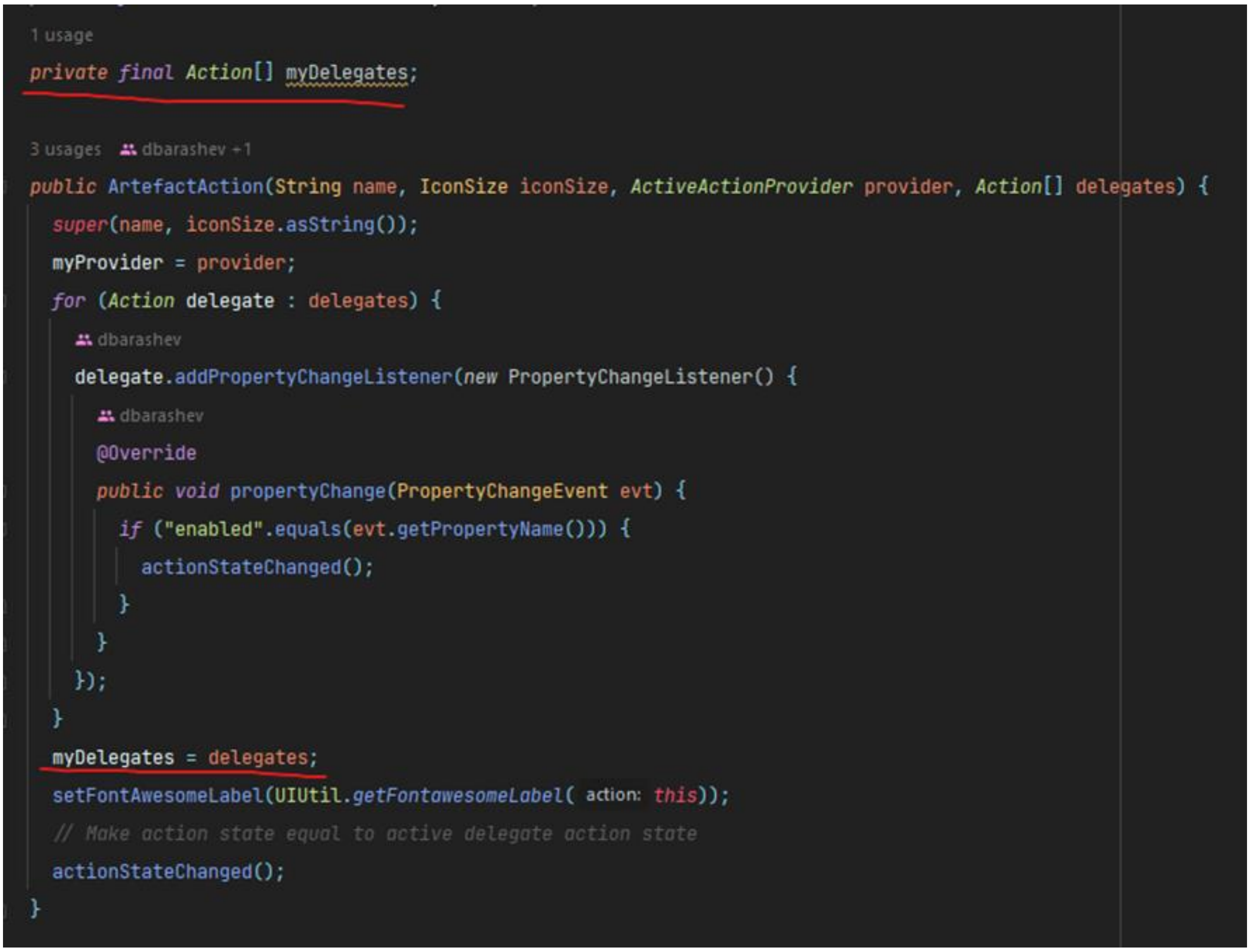
```
/.../
package org.ganttproject.impex.htmlpdf;
```

WebStartIDClass

WebStartIDClass org.ganttproject.impex.htmlpdf GanttProject.org.ganttproject.impex.htmlpdf.main
WebStartIDClass org.ganttproject GanttProject.ganttproject.main
WebStartIDClass biz.ganttproject.impex.msproject2 GanttProject.biz.ganttproject.impex.msproject2.main
WebStartIDClass org.ganttproject.chart.pert GanttProject.org.ganttproject.chart.pert.main

File:gantProj\ganttproject\biz.ganttproject.impex.msproject2\src\main\java\biz\ganttproject\impex\msproject2\WebStartIDClass

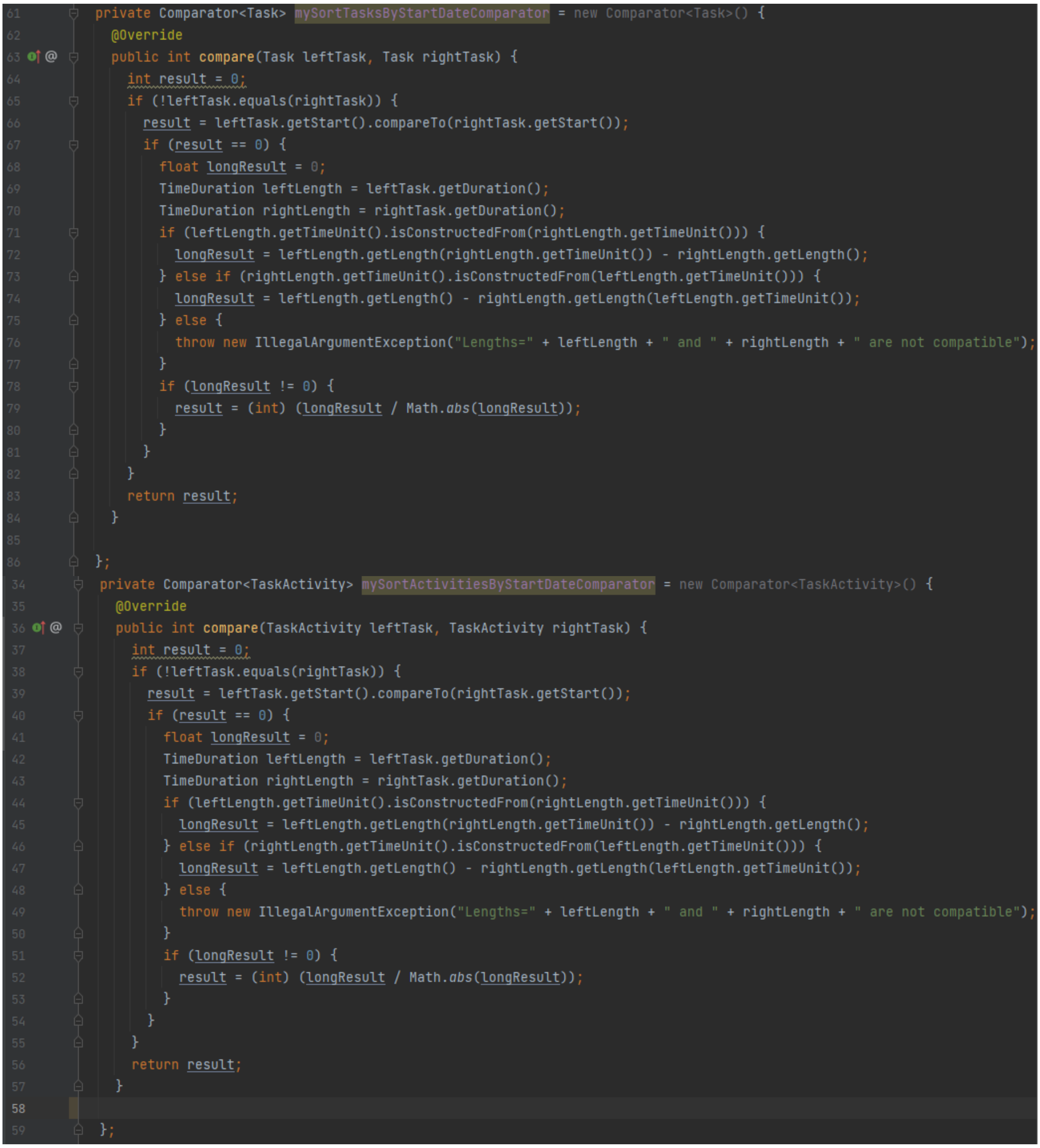
Text: The WebStartIDClass class is empty (all classes with this name are), could also be considered Specular Generality depending on the context.
Solution: Delete the classes or use them.
Author: Iago Paulo
Review: I agree it’s DeadCode, not used and could also be a situation of over-engineering anticipating future functionality **by** Francisco Vasco



```
1 usage
private final Action[] myDelegates;

3 usages  dbarashev +1
public ArtefactAction(String name, IconSize iconSize, ActiveActionProvider provider, Action[] delegates) {
  super(name, iconSize.asString());
  myProvider = provider;
  for (Action delegate : delegates) {
    dbarashev
    delegate.addPropertyChangeListener(new PropertyChangeListener() {
      dbarashev
      @Override
      public void propertyChange(PropertyChangeEvent evt) {
        if ("enabled".equals(evt.getPropertyName())) {
          actionStateChanged();
        }
      }
    });
  }
  myDelegates = delegates;
  setFontAwesomeLabel(UIUtil.getFontawesomeLabel( action: this));
  // Make action state equal to active delegate action state
  actionStateChanged();
}
```

File:ganttproject/src/main/java/net/sourceforge/ganttproject/action/ArtefactAction.java
Text: The myDelegates (private final Action[] myDelegates) is initialized but never used.
Solution: It can be safely deleted.
Author: João Oliveira
Review: Yes, it's true that, being private, it isn't accessible from outside the scope and the only thing it could be doing is preventing some object from being taken by the garbage collector **by** Ricardo Gonçalo

Duplicated Code



```
61  private Comparator<Task> mySortTasksByStartDateComparator = new Comparator<Task>() {
62    @Override
63    public int compare(Task leftTask, Task rightTask) {
64      int result = 0;
65      if (!leftTask.equals(rightTask)) {
66        result = leftTask.getStart().compareTo(rightTask.getStart());
67        if (result == 0) {
68          float longResult = 0;
69          TimeDuration leftLength = leftTask.getDuration();
70          TimeDuration rightLength = rightTask.getDuration();
71          if (leftLength.getTimeUnit().isConstructedFrom(rightLength.getTimeUnit())) {
72            longResult = leftLength.getLength(rightLength.getTimeUnit()) - rightLength.getLength();
73          } else if (rightLength.getTimeUnit().isConstructedFrom(leftLength.getTimeUnit())) {
74            longResult = leftLength.getLength() - rightLength.getLength(leftLength.getTimeUnit());
75          } else {
76            throw new IllegalArgumentException("Lengths=" + leftLength + " and " + rightLength + " are not compatible");
77          }
78          if (longResult != 0) {
79            result = (int) (longResult / Math.abs(longResult));
80          }
81        }
82      }
83      return result;
84    }
85
86  };
34  private Comparator<TaskActivity> mySortActivitiesByStartDateComparator = new Comparator<TaskActivity>() {
35    @Override
36    public int compare(TaskActivity leftTask, TaskActivity rightTask) {
37      int result = 0;
38      if (!leftTask.equals(rightTask)) {
39        result = leftTask.getStart().compareTo(rightTask.getStart());
40        if (result == 0) {
41          float longResult = 0;
42          TimeDuration leftLength = leftTask.getDuration();
43          TimeDuration rightLength = rightTask.getDuration();
44          if (leftLength.getTimeUnit().isConstructedFrom(rightLength.getTimeUnit())) {
45            longResult = leftLength.getLength(rightLength.getTimeUnit()) - rightLength.getLength();
46          } else if (rightLength.getTimeUnit().isConstructedFrom(leftLength.getTimeUnit())) {
47            longResult = leftLength.getLength() - rightLength.getLength(leftLength.getTimeUnit());
48          } else {
49            throw new IllegalArgumentException("Lengths=" + leftLength + " and " + rightLength + " are not compatible");
50          }
51          if (longResult != 0) {
52            result = (int) (longResult / Math.abs(longResult));
53          }
54        }
55      }
56      return result;
57    }
58
59  };
```

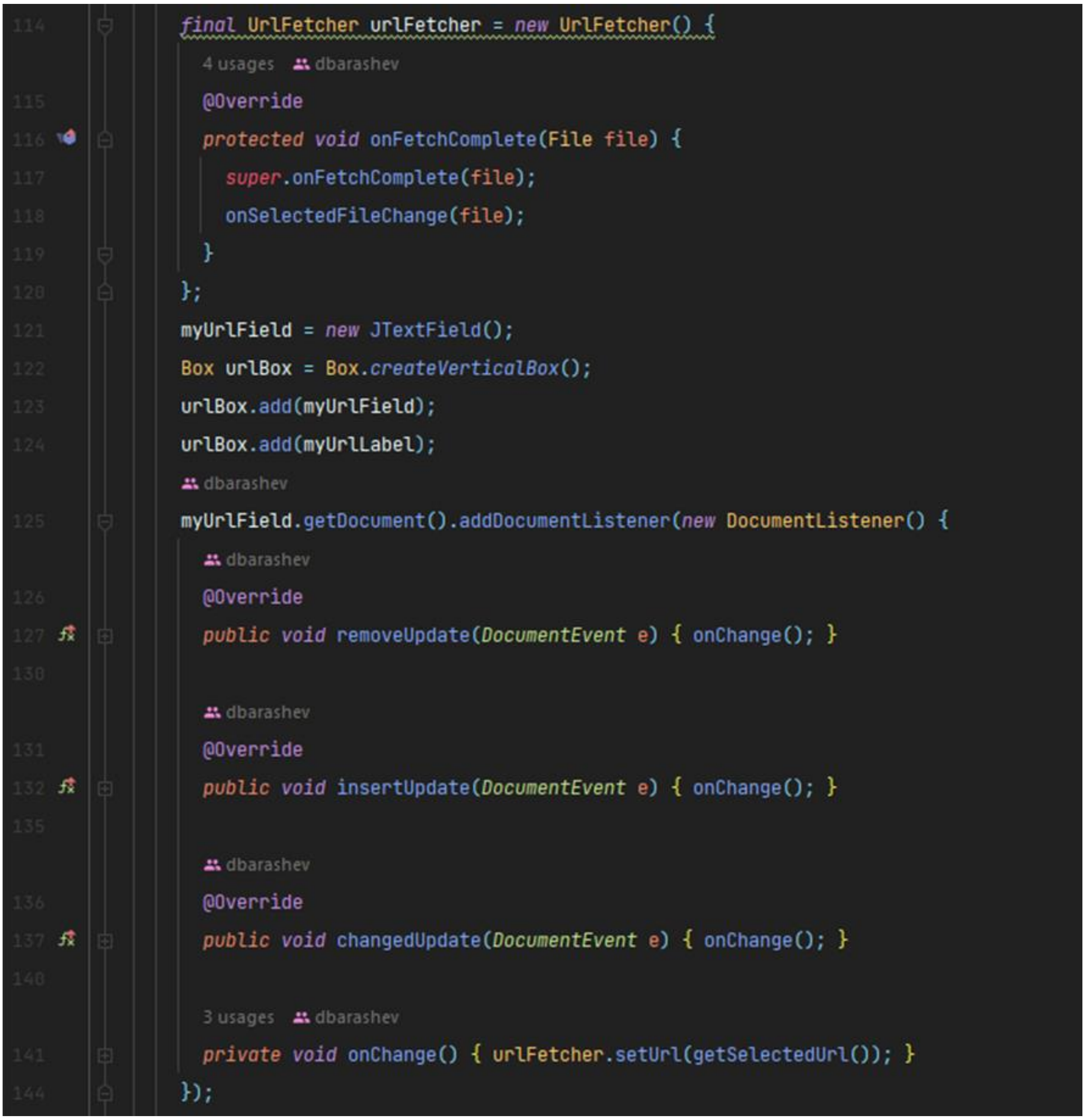
File: ganttproject/src/main/java/net/sourceforge/ganttproject/task/algorithm/SortTasksAlgorithm.java

Text: On the SortTasksAlgorithm class the code on lines 44-53 is identical to the code on lines 71-80.

Solution: Extract part of the duplicated code to an auxiliar method.

Author: Francisco Vasco

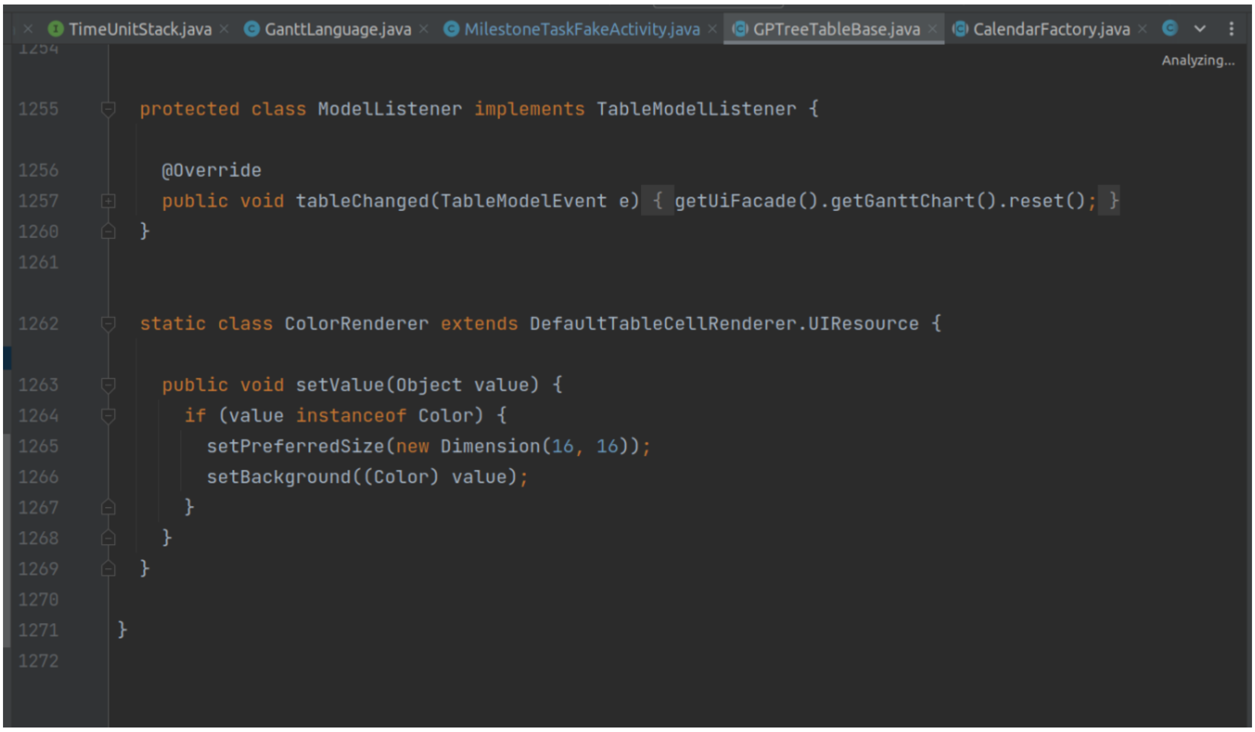
Review: These 2 pieces of code are in fact identical, the solution seems appropriate **by** Iago Paulo



```
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         });
```

File:ganttproject/src/main/java/net/sourceforge/ganttproject/gui/FileChooserPageBase.java e ganttproject/src/main/java/net/sourceforge/ganttproject/wizard/AbstractFileChooserPage.java
Text: Code on line 114 of the first class is duplicated in both classes.
Solution: The code could be extracted to a method or helper class
Author: João Oliveira
Review: The identification seems correct. Although this is just a line, errors could appear after future refactoring, this solution prevents that **by** Francisco Vasco

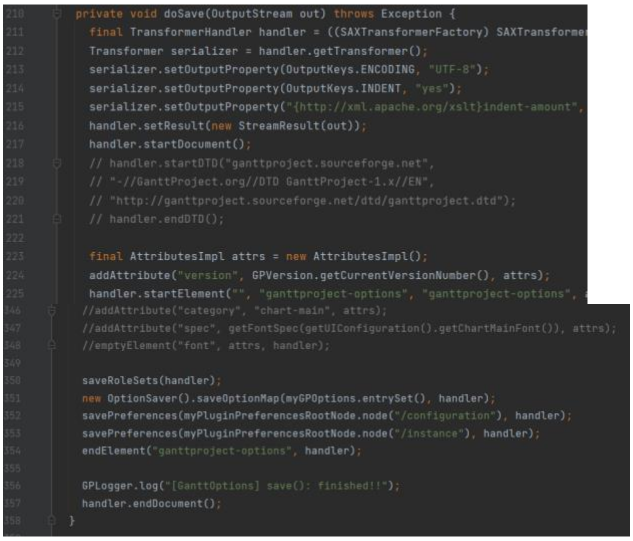
Large class



```
  protected class ModelListener implements TableModelListener {

    @Override
    public void tableChanged(TableModelEvent e) { getUiFacade().getGanttChart().reset(); }
  }


  static class ColorRenderer extends DefaultTableCellRenderer.UIResource {

    public void setValue(Object value) {
      if (value instanceof Color) {
        setPreferredSize(new Dimension(16, 16));
        setBackground((Color) value);
      }
    }
  }

}

```

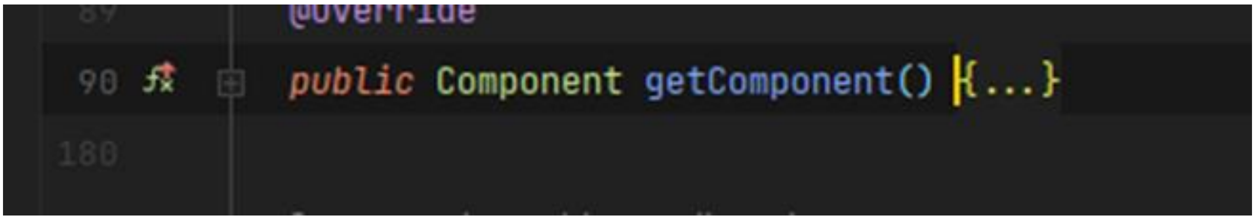
File:ganttproject/src/main/java/net/sourceforge/ganttproject/GPTreeTableBase.java
Text: The GPTreeTableBase class has more than 1000 code lines and has many other classes in it and long methods.
Solution: Dividir a class em outras classes mais pequenas.
Author: James Furtado
Review: Class has over 1200 lines and has 6 classes in it, it is in fact a long class. **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();
  }
```

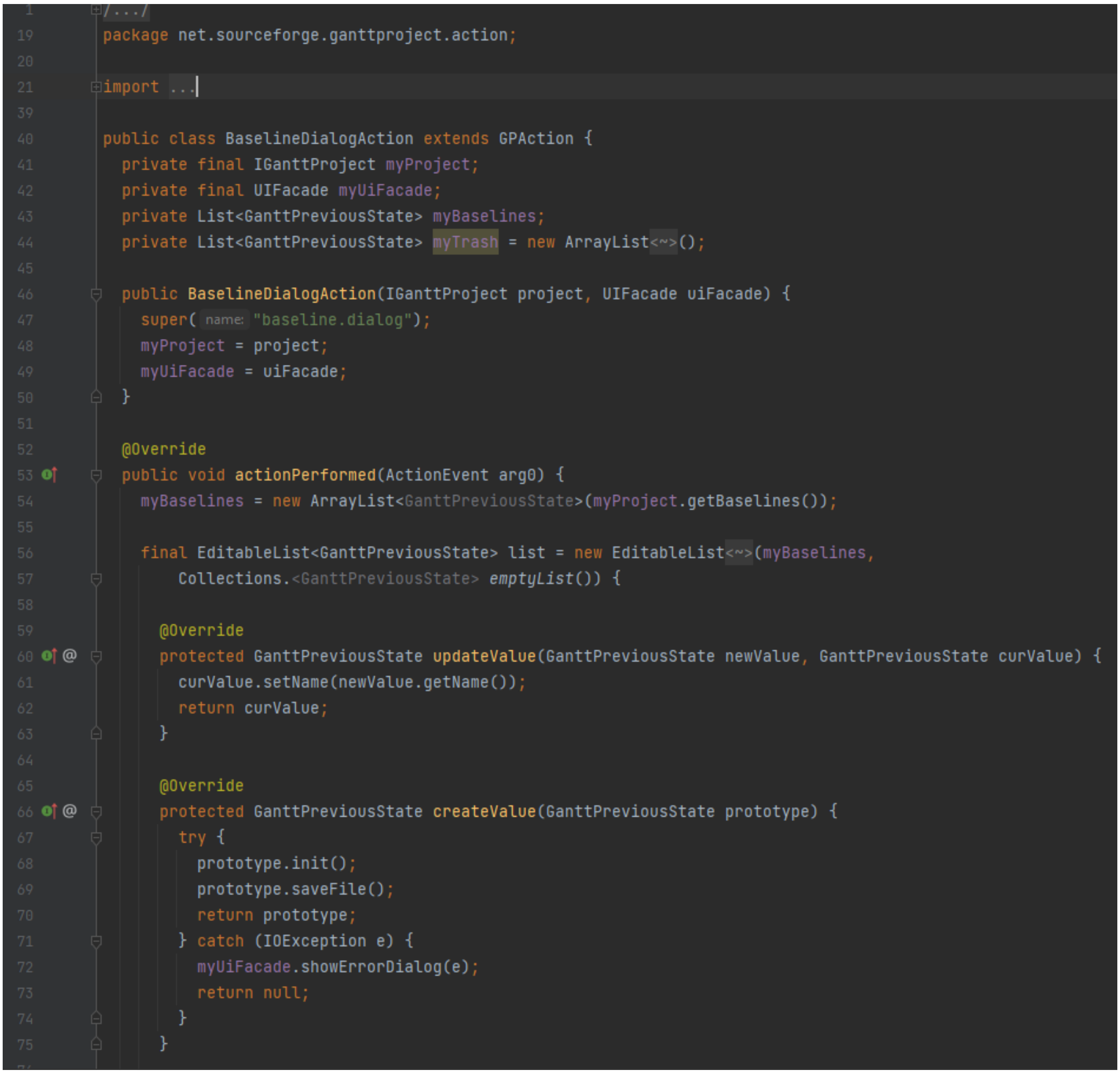
File: ganttproject/src/main/java/net/sourceforge/ganttproject/GanttOptions.java
Text: In the doSave method, lines 210-358, there’s a very large method with commented out code.
Solution: Simplify de method’s tasks in subtasks to make it more readable and understandable.
Author: Ricardo Gonçalo
Review: The method is too large, although it’s purpose is to initialize the UI. Auxiliar functions could have been used to clarify what the method does. The large sections of commented code should be deleted **by** João Oliveira



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

File:ganttproject/src/main/java/net/sourceforge/ganttproject/gui/FileChooserPageBase.java
Text: The getComponent()is too large (90 lines).
Solution: Create auxiliar methods.
Author: João Oliveira
Review: I also think this could be considered a code smell **by** James Hertz

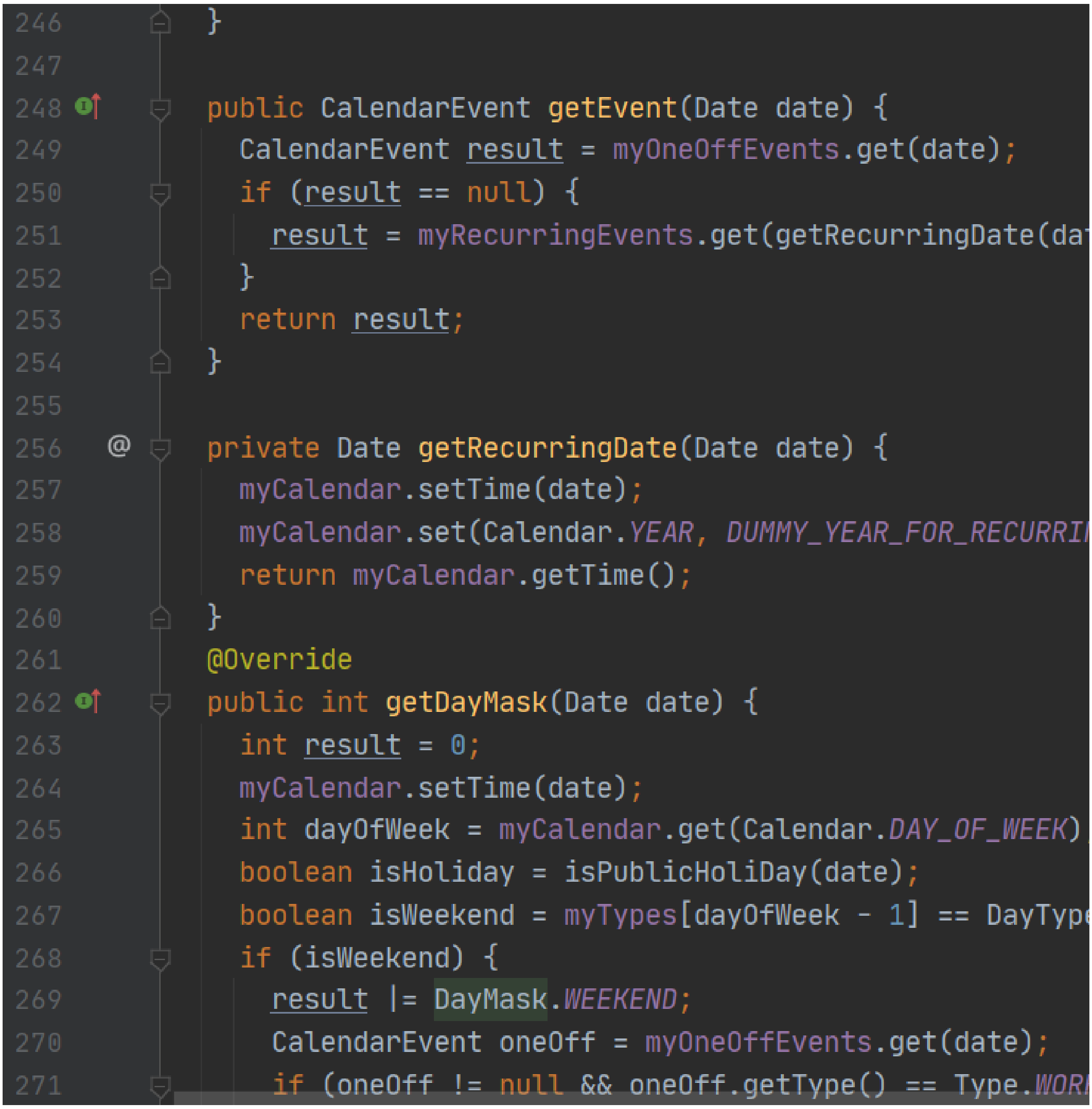
No comment



```
/.../
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)

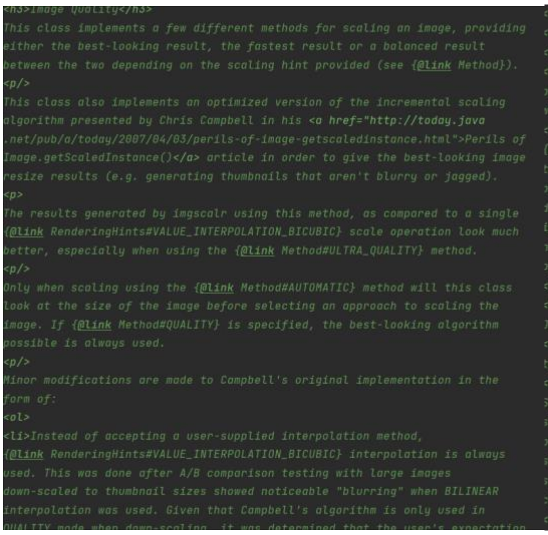
File:ganttproject/src/main/java/net/sourceforge/ganttproject/action/BaselineDialogAction.java
Solution: Add comments.
Text: The BaselineDialogAction has no comments despite it’s complexity.
Author: Francisco Vasco
Review: The class has no comments so I agree it is a code smell **by** James Hertz



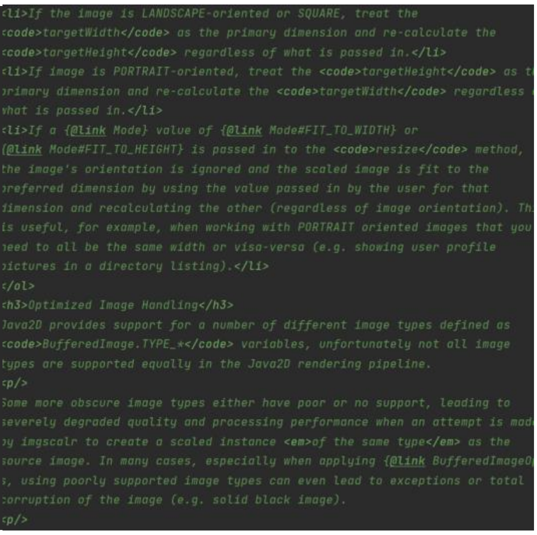
```
}

public CalendarEvent getEvent(Date date) {
  CalendarEvent result = myOneOffEvents.get(date);
  if (result == null) {
    result = myRecurringEvents.get(getRecurringDate(da
  }
  return result;
}

private Date getRecurringDate(Date date) {
  myCalendar.setTime(date);
  myCalendar.set(Calendar.YEAR, DUMMY_YEAR_FOR_RECURRI
  return myCalendar.getTime();
}
@Override
public int getDayMask(Date date) {
  int result = 0;
  myCalendar.setTime(date);
  int dayOfWeek = myCalendar.get(Calendar.DAY_OF_WEEK)
  boolean isHoliday = isPublicHoliDay(date);
  boolean isWeekend = myTypes[dayOfWeek - 1] == DayType
  if (isWeekend) {
    result |= DayMask.WEEKEND;
    CalendarEvent oneOff = myOneOffEvents.get(date);
    if (oneOff != null && oneOff.getType() == Type.WOR
```

File:biz.ganttproject.core/src/main/java/biz/ganttproject/core/calendar/WeekendCalendarImpl.java
Text: This class has no comments.
Solution: Add comments.
Author: Iago Paulo
Review: Yes, for a class with 380 lines it has few comments **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/>
```

File: ganttproject/src/main/java/org/imgscalr/Scalr.java

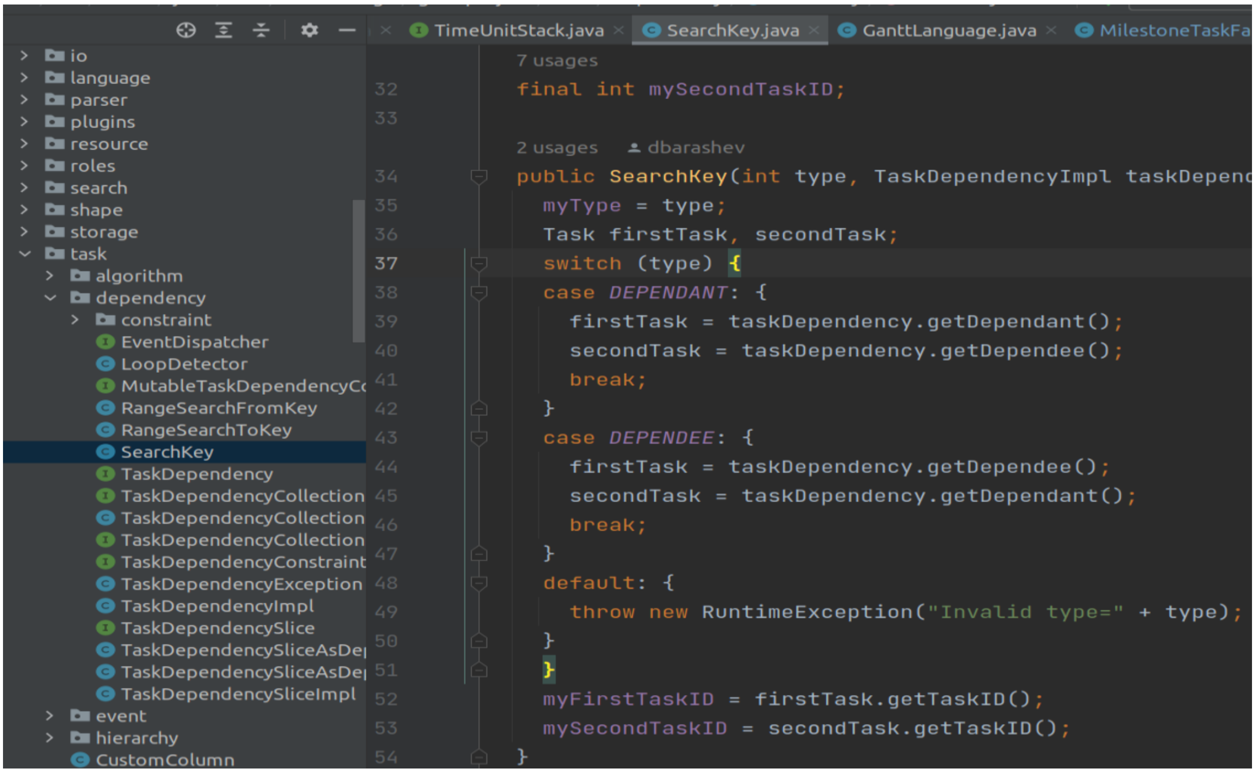
Text: The comments are too long. For example, the one right above the class has over 130 lines.

Solution: Simplify these comments, is there really a need for explanations this extensive for how the class works? Surely not, too much unnecessary information.

Author: Ricardo Gonçalo

Review: Unlike the rest of the code (which has almost no comments) this class has too many, so it’s clearly a code smell **by** James Furtado

Switch statement



```
7 usages
final int mySecondTaskID;

2 usages  dbarashev
public SearchKey(int type, TaskDependencyImpl taskDepend
  myType = type;
  Task firstTask, secondTask;
  switch (type) {
  case DEPENDANT: {
    firstTask = taskDependency.getDependant();
    secondTask = taskDependency.getDependee();
    break;
  }
  case DEPENDEE: {
    firstTask = taskDependency.getDependee();
    secondTask = taskDependency.getDependant();
    break;
  }
  default: {
    throw new RuntimeException("Invalid type=" + type);
  }
  }
  myFirstTaskID = firstTask.getTaskID();
  mySecondTaskID = secondTask.getTaskID();
}
```

File:ganttproject/src/main/java/net/sourceforge/ganttproject/task/dependency/SearchKey.java

Text: The class has an attribute myType which is an integer. The constants DEPENDEE and DEPENDANT are used as types. And it has a switch statement in the constructor for both cases and an exception otherwise.

Solution: Create two subclasses. Not only do we eliminate the myType attribute but we also eliminate the exception which is also good.

Author: James Furtado

Review: Yes, at least replace it with an Enum **by** Ricardo Gonçalo

Phase 2:

Sprint 1

During the sprint the main task was to identify two features that we wanted to add to the ganttproject. The **Scrum master** of this week was Ricardo Gonaçalo.

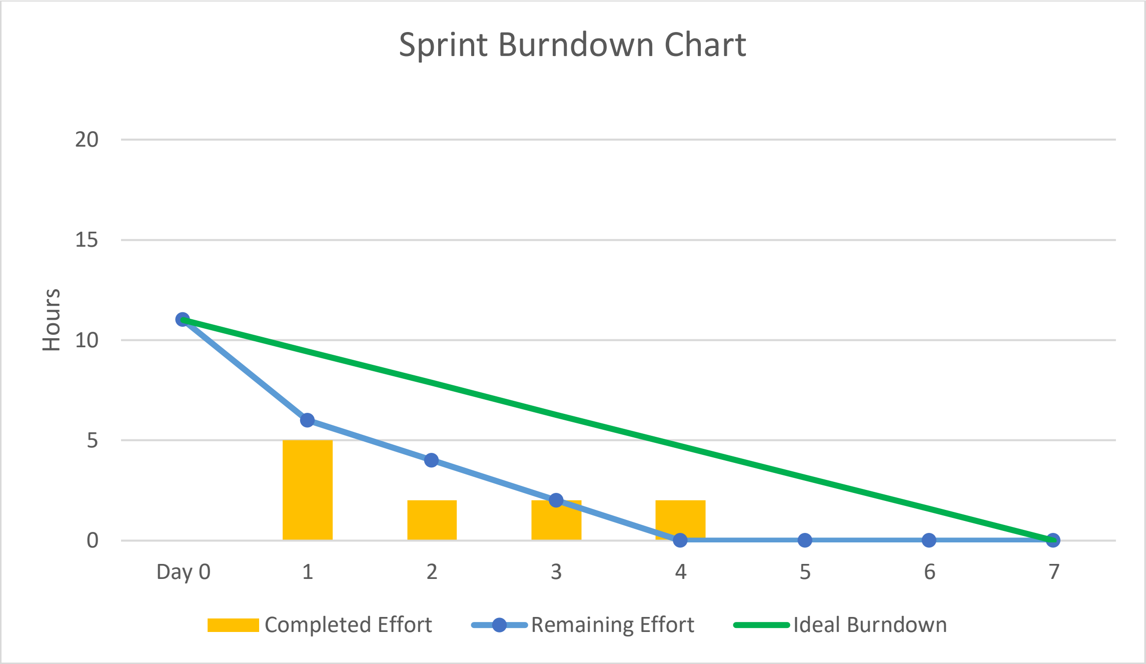
Sprint Backlog

Todo	In Progress	Reviewing	Done
			Sugerir 2 Features (Francisco)
			Sugerir 2 Features (Iago)
			Sugerir 2 Features (James)
			Sugerir 2 Features (Joao)
			Sugerir 2 Features (Ricardo)
			Criar User Story Primeira Feature
	-		Criar User Story Segunda Feature
			Descrever Totalmente Primeita Feature
			Descrever Totalmente Segunda Feature

Burndown chart

Sprint 2 Burndown Chart

Task ID	Task Description	Initial Estimate	7/Nov/22	8/Nov/22	9/Nov/22	10/Nov/22	11/Nov/22	11/Nov/22	13/Nov/22
		Day 0	1	2	3	4	5	6	7
1	Sugerir 2 Features (Francisco)	1	1						
2	Sugerir 2 Features (Iago)	1	1						
3	Sugerir 2 Features (James)	1	1						
4	Sugerir 2 Features (Joao)	1	1						
5	Sugerir 2 Features (Ricardo)	1	1						
6	Criar User Story Primeira Feature	1		1					
7	Criar User Story Segunda Feature	1		1					
8	Descrever Totalmente Primeita Feature	2			1	1			
9	Descrever Totalmente Segunda Feature	2			1	1			
Completed Effort		0	5	2	2	2	0	0	0
Remaining Effort		11	6.0	4.0	2.0	0.0	0.0	0.0	0.0
Ideal Burndown		11	9.4	7.9	6.3	4.7	3.1	1.6	0.0



Produced content

After a bit of discussion in the end we choose these two features. We list below the features and their respective user story.

<u>Feature 1:</u>

Description:

When a file is loaded into the application, there will be an option for the user to extend the duration of tasks that were supposed to end in the past to end during the current day, also rearranging tasks according to their dependencies. Additionally, there will be a button that will also have this feature.

It should also (optionally) automatically make the end date of a task supposed to end in the future to the current one, when it is marked as finished.

User Story: As a project manager, **I want to** see the task duration changing dynamically in case a task takes more or less time than anticipated **so that** I may have a more accurate project duration time.

<u>Feature 2:</u>

Description: Statistics revealing the state of the project regarding how complete it is.

User Story: As a user, **I want to** easily see the project completion state, **so that** I can easily know if the project is evolving as expected.

Sprint 2

In this sprint we produced the use cases diagrams asked for the whole project. The **Scrum master** of this week was Francisco Vasco.

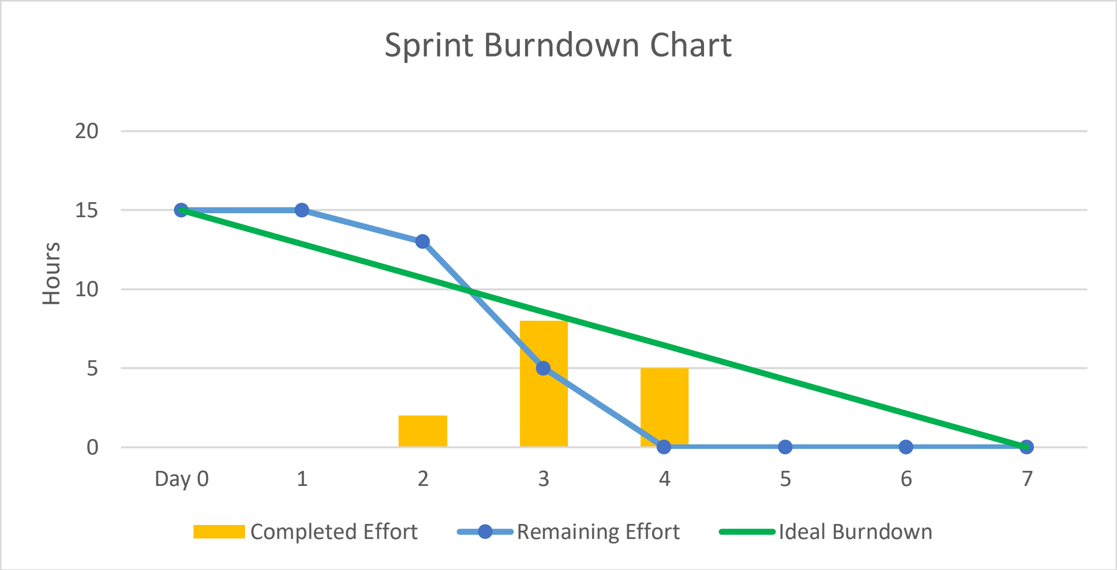
Sprint Backlog

Todo	In Progress	Reviewing	Done
			Find/discribe related use cases (Francisco)
			Find/discribe related use cases (Iago)
			Find/discribe related use cases (James)
			Find/discribe related use cases (Joaoa)
			Find/discribe related use cases (Ricardo)
			Develop an use case sub-diagram (Francisco)
			Develop an use case sub-diagram (Iago)
			Develop an use case sub-diagram (James)
			Develop an use case sub-diagram (Joao)
			Develop an use case sub-diagram (Ricardo)

Burndown chart

Sprint 3 Burndown Chart

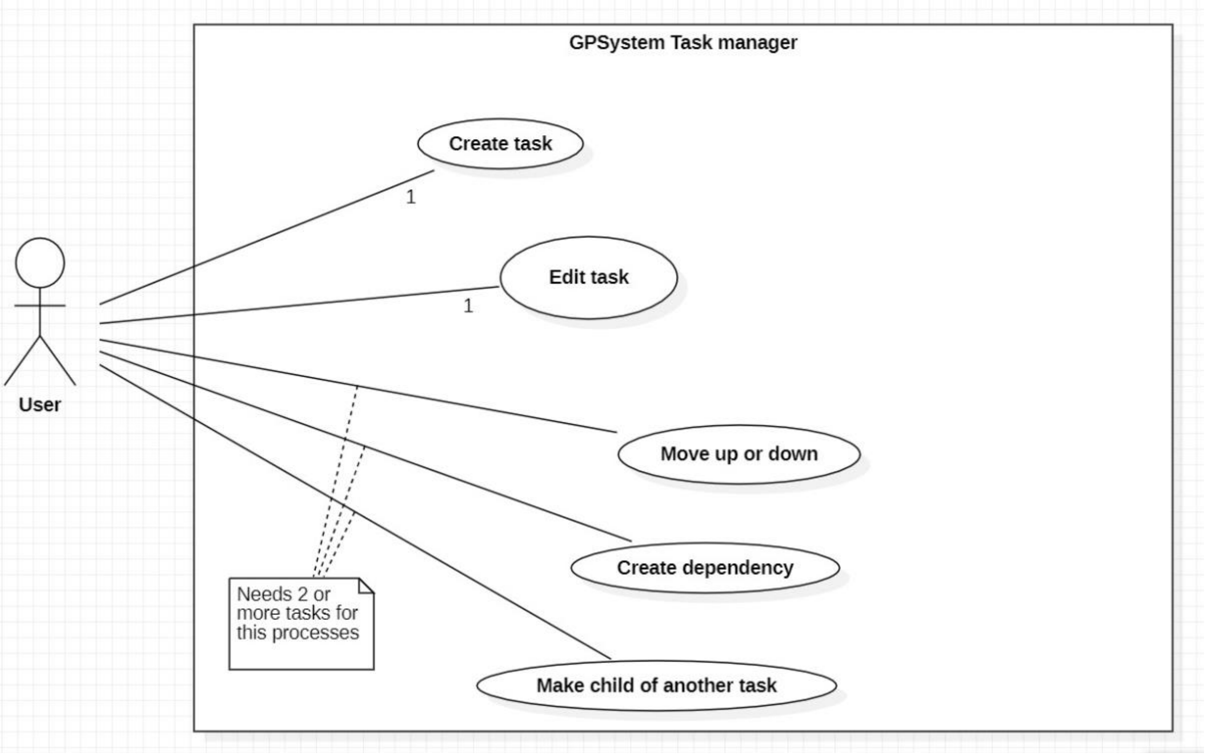
Task ID	Task Description	Initial Estimate	14/Nov/22	15/Nov/22	16/Nov/22	17/Nov/22	18/Nov/22	19/Nov/22	20/Nov/22
		Day 0	1	2	3	4	5	6	7
1	Find/discribe related use cases (Francisco)	2			1	1			
2	Find/discribe related use cases (Iago)	2		1	1				
3	Find/discribe related use cases (James)	2		1	1				
4	Find/discribe related use cases (Joaoa)	2			2				
5	Find/discribe related use cases (Ricardo)	2			1	1			
6	Develop an use case sub-diagram (Francisco)	1				1			
7	Develop an use case sub-diagram (Iago)	1			1				
8	Develop an use case sub-diagram (James)	1			1				
9	Develop an use case sub-diagram (Joao)	1				1			
10	Develop an use case sub-diagram (Ricardo)	1				1			
Completed Effort		0	0	2	8	5	0	0	0
Remaining Effort		15	15.0	13.0	5.0	0.0	0.0	0.0	0.0
Ideal Burndown		15	12.9	10.7	8.6	6.4	4.3	2.1	0.0



Produced content

Here below are the diagram produced along with its author and the reviewer comments.

<u>Task manager</u>



Author: Francisco Vasco

Description:

<u>Name:</u> Create task
<u>Description:</u> Create a new task on the chart
<u>Actors:</u> User

<u>Name:</u> Edit task
<u>Description:</u> Edit an existing task's properties
<u>Actors:</u> User

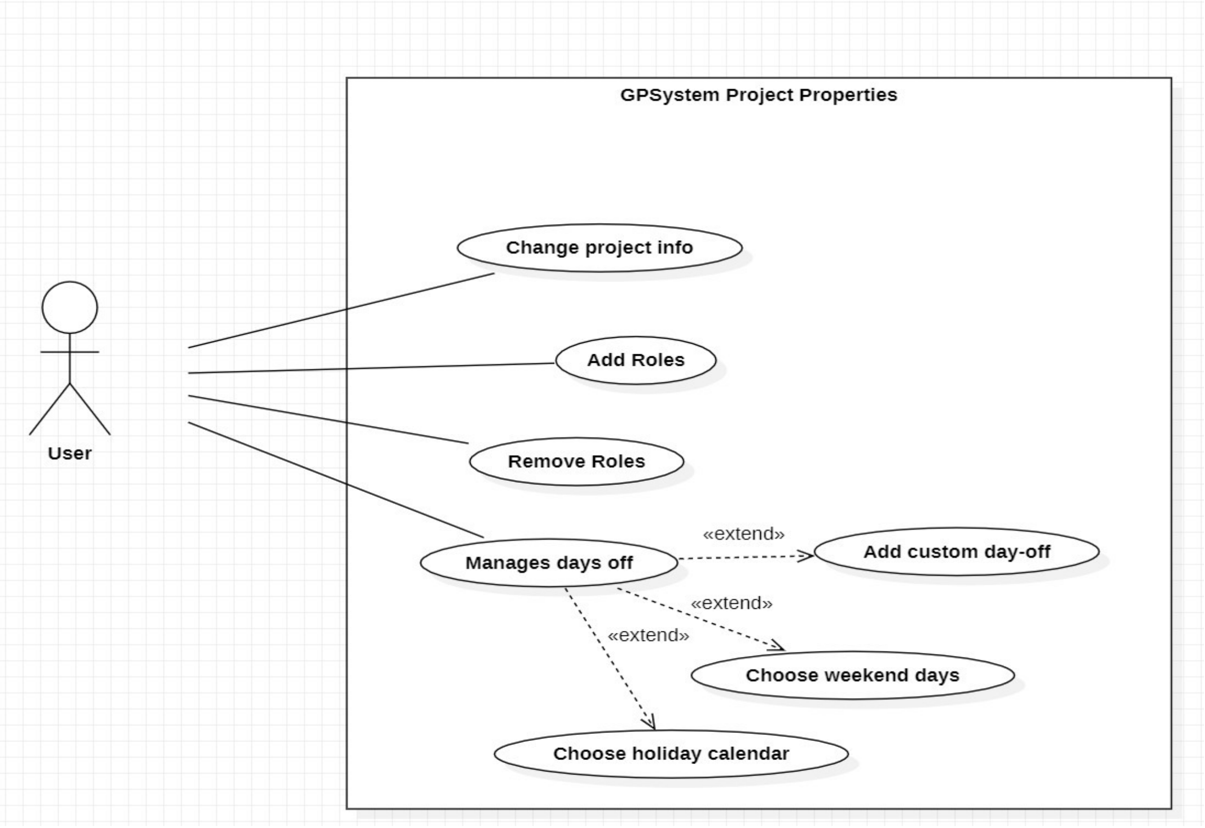
<u>Name:</u> Move up or down
<u>Description:</u> Moves the task on the chart, relative to other tasks
<u>Actors:</u> User

<u>Name:</u> Create dependency
<u>Description:</u> creates a dependency between 2 tasks
<u>Actors:</u> User

<u>Name:</u> Make child of another task
<u>Description:</u> the selected task becomes a requirement for another task
<u>Actors:</u> User

Review: The overall idea and the manipulation of the task was well captured **by** Ricardo Gonçalo.

Properties manager



Author: Iago Paulo
Description:

The only actor is the user, which can do all this use cases.

Name: Change project info
desc: User can change/add things like the name of the project, Organization, Web Link and description.

Name: Add Roles
desc: User can add custom roles that are not on the default roles yet.

Name: Remove Roles
desc: User can remove those custom roles.

Name: Manage days off
desc: User can use this to set the days that he will have a break.

Name: Add custom day-off

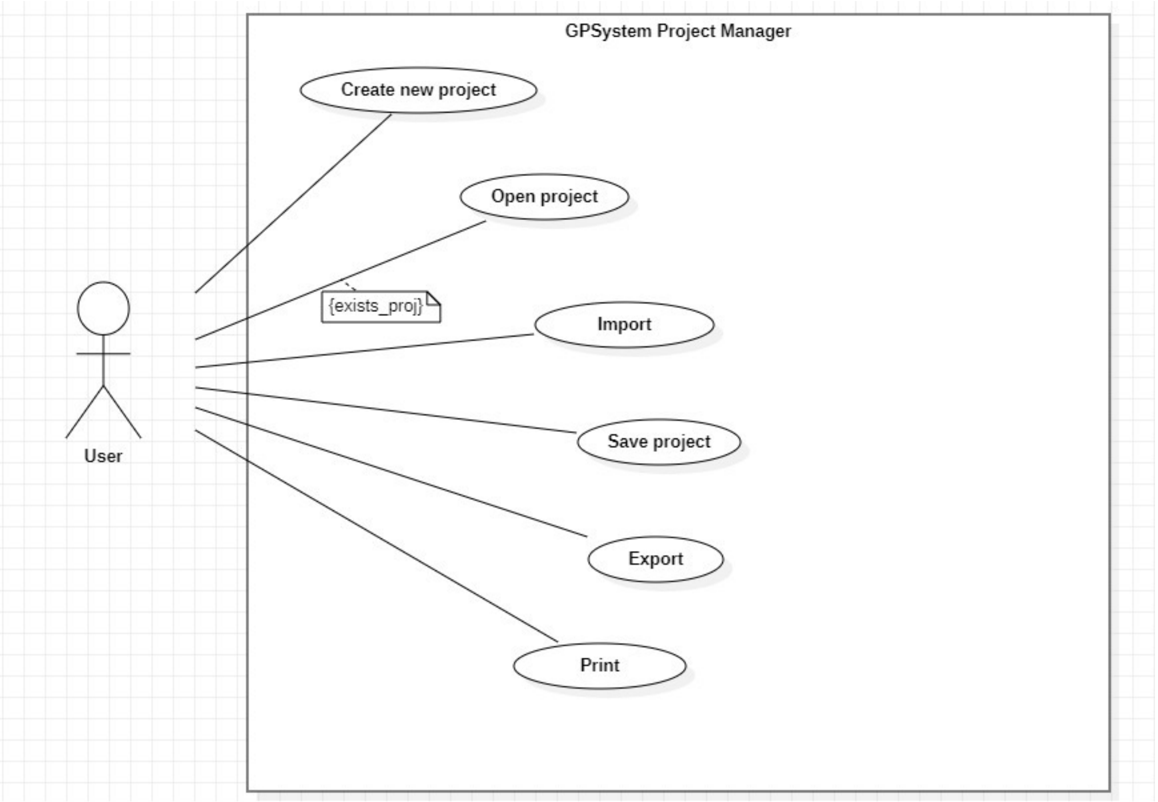
desc: User can add a specific day off, no holiday and no weekend day, just an arbitrary day off.

Name: Choose weekend days
desc: User can set the weekend days instead of the default Saturday and Sunday
(Example: Some companies also work on Saturday, so Sunday is the only Weekend Day)

Name: Choose holiday calendar
desc: User can choose his holidays on this calendar from several countries, making it easy to put the custom days-off from each country.

Review: He diagram seems to encapsulate all the functionalities associated with the project properties. The usage of the «extends» keyword between "Manages days off" and its related use cases seems correct as it is not required to use any of these use cases, they are optional **by** Francisco Vasco.

Project manager



Author: James Furtado

Description:

Name: Create new Project
Description: The actor can create a new project that will hold a set of tasks. The actor will have to give some information in order to do so, such as the project name.
Actors: User

Name: Save Project
Description: The actor can save the project (with all its tasks) in a file that can later be opened.
Actors: User

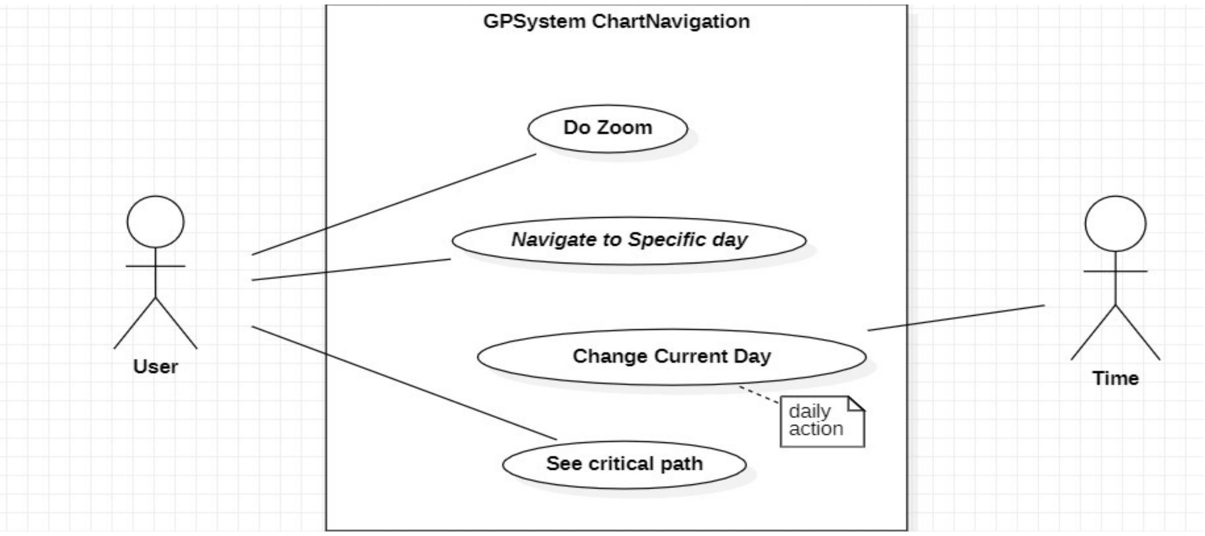
Name: Open Project
Description: The actor selects a project (from a file or an url - webserver). The system closes the current opened project (if any) and loads the selected project's (tasks, resources, etc...) so the project can be edited.
Actors: User

Name: Export
Description: The actor can export (save to a file) the project to a format other the default ganttproject.
Actors: User

Name: Import
Description: The actor chooses a file (or an url) that was previously export as a project and the system adds the task (resources, etc...) of such project to the current opened project.
Actors: User

Review: Everything is correct. Contains the name, the description and the actor **by** João Oliveira.

Timeline manager



Author: João Oliveira

Description:

Use case referring to the timeline manager.
Has two actors, one primary and one secondary.
The primary is the user that interacts with the system
The secondary is the time.

Name: Do Zoom
Description: Action that allows the user to zoom in and out.

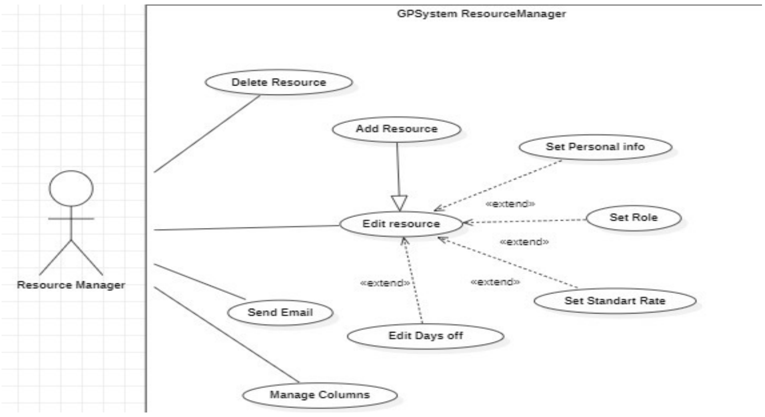
Name: Navigate to specific day
Description: The user can navigate in the timeline to select the date range he want to see. Can be done by dragging the timeline or by selecting some options, like "Past", "Future", "Project Start", "Today", "Project End" or "Selection".

Name: See critical path:
Description: The user can choose an option to see the critical path.

Name: Change current day
Description: The secondary actor, the time, updates the day to the current one.

Review: I agree, but we can also say that the Time updates de current day *every day*, although it is self-explanatory **by** Iago Paulo

Timeline manager



Author: Ricardo Gonçalo

Description:

Name: Resource Manager
desc: The actor responsible for the management of the human resources, **being the principal actor of all the actions.**

Name: Edit Resource:
desc: User can remove those custom roles. Which is inherited by: Add Resource and extended by: Set Personal Info, Set Role, Set Standard Rate and Edit Days Off.

Name: Delete resource:
desc: The user can delete a resource from the system

Name: Add Resource:
desc: The user can add a resource to the system, setting its values (Edit Resource)

Name: Set Personal Info:
desc: The user can set the name, phone number and mail of the Resource

Name: Set Role
desc: The user can set the role of the resource, choosing from available values

Name: Set Standart (standard) Rate:
desc: The user can set the standart rate of the resource

Name: Edit Days Off:
desc: The user can add or remove intervals of days off

Name: Send Email:
desc: The user can send an email to the resource

Name: Manage columns:
desc: The user can show, hide, create columns, delete created columns and set values in the columns of the Resources Chart.

Review:

When looking at the use case diagram I agree with what was done regarding
the use case "Delete Resource" and "Send email".

Regarding "Manage columns" I don't think it's even a use case, I think that the column is a way
to access others, use cases.
My major critic is regarding the use case "Edit resource" which seems to be an empty
use case where everything that makes the use case extends it.

What I think is that probably it should've been inherited by the uses cases that extends it (since these use cases have one thing in common - they change some properties of a resource). I also think that it's a bit weird to make the use case add resource inherit from "Edit resource"

by James Furtado.

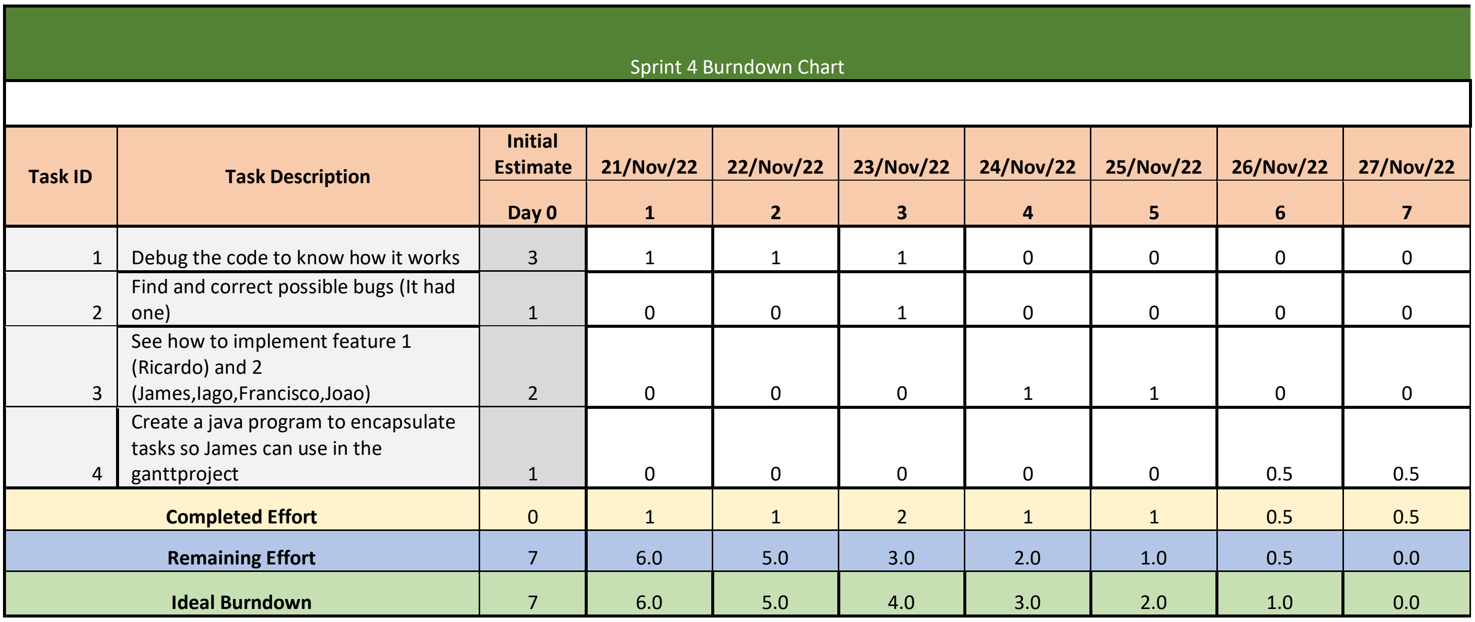
Sprint 3

In this sprint we spent some time exploring the ganttproject and trying to fix some bugs. The **Scrum master** of this week was Iago Paulo.

Sprint Backlog

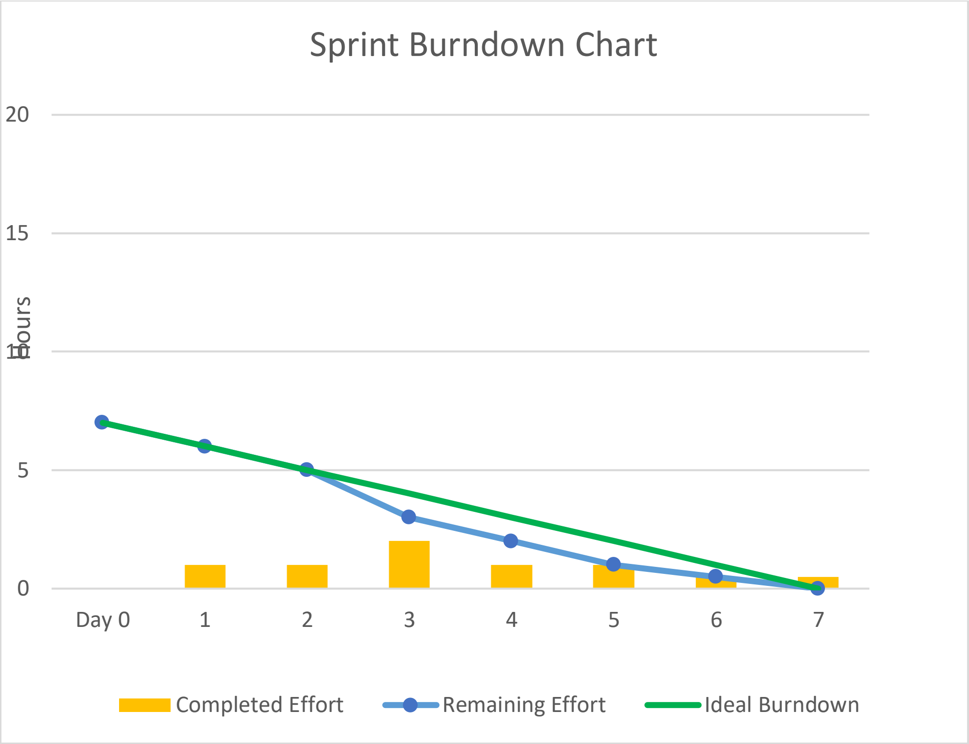
Todo	In Progress	Reviewing	Done
			Debugs (Iago)
			Correct bugs (Iago)
			Analyse how to implement features 1 (Ricardo)
			Analyse how to implement features 2 (Iago,James,Joao,Francisco)
			Make java program to encapsulate tasks (Iago)

Burndown chart



Sprint 4 Burndown Chart

Task ID	Task Description	Initial Estimate	21/Nov/22	22/Nov/22	23/Nov/22	24/Nov/22	25/Nov/22	26/Nov/22	27/Nov/22
		Day 0	1	2	3	4	5	6	7
1	Debug the code to know how it works	3	1	1	1	0	0	0	0
2	Find and correct possible bugs (It had one)	1	0	0	1	0	0	0	0
3	See how to implement feature 1 (Ricardo) and 2 (James,Iago,Francisco,Joao)	2	0	0	0	1	1	0	0
4	Create a java program to encapsulate tasks so James can use in the ganttproject	1	0	0	0	0	0	0.5	0.5
Completed Effort		0	1	1	2	1	1	0.5	0.5
Remaining Effort		7	6.0	5.0	3.0	2.0	1.0	0.5	0.0
Ideal Burndown		7	6.0	5.0	4.0	3.0	2.0	1.0	0.0



Produced content

We didn't do nothing more than exploring the code and think discussing about it. During this our dear college **Iago Paulo** found a bug and fixed it. Whenever we moved a task a count that is supposed to increase only when we create a new task would increase so if we moved a task **x** times the next task would appear with a counter of the + **(x + 1)** of the last task created.

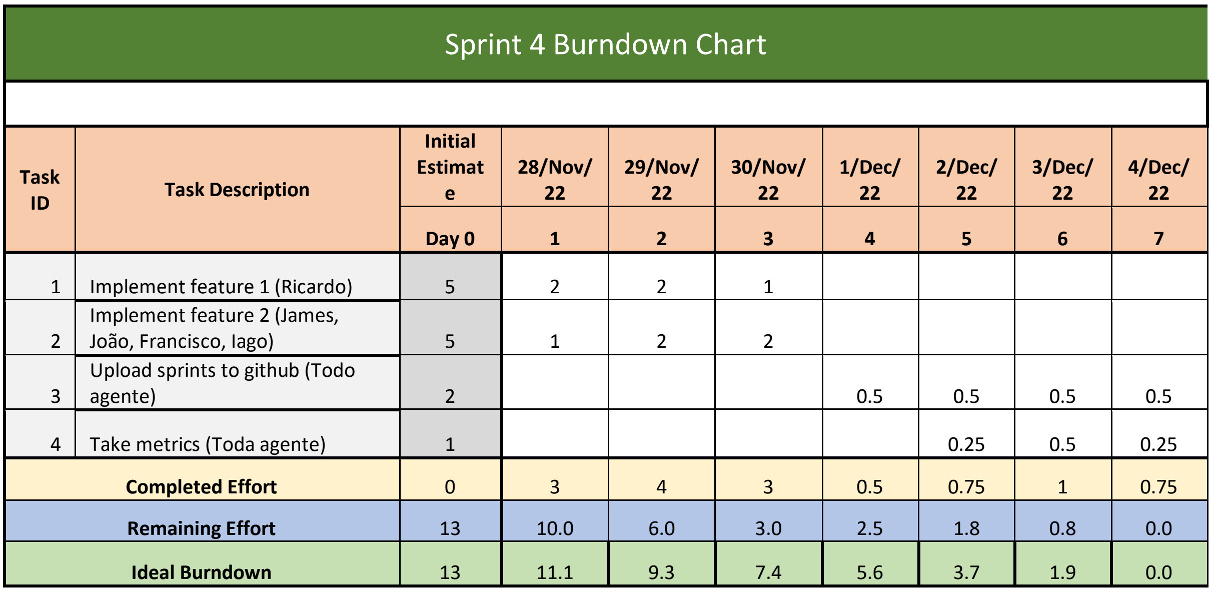
Sprint 4

In this sprint we implemented the features, the unit test for the features and we did the metrics report.

Sprint Backlog

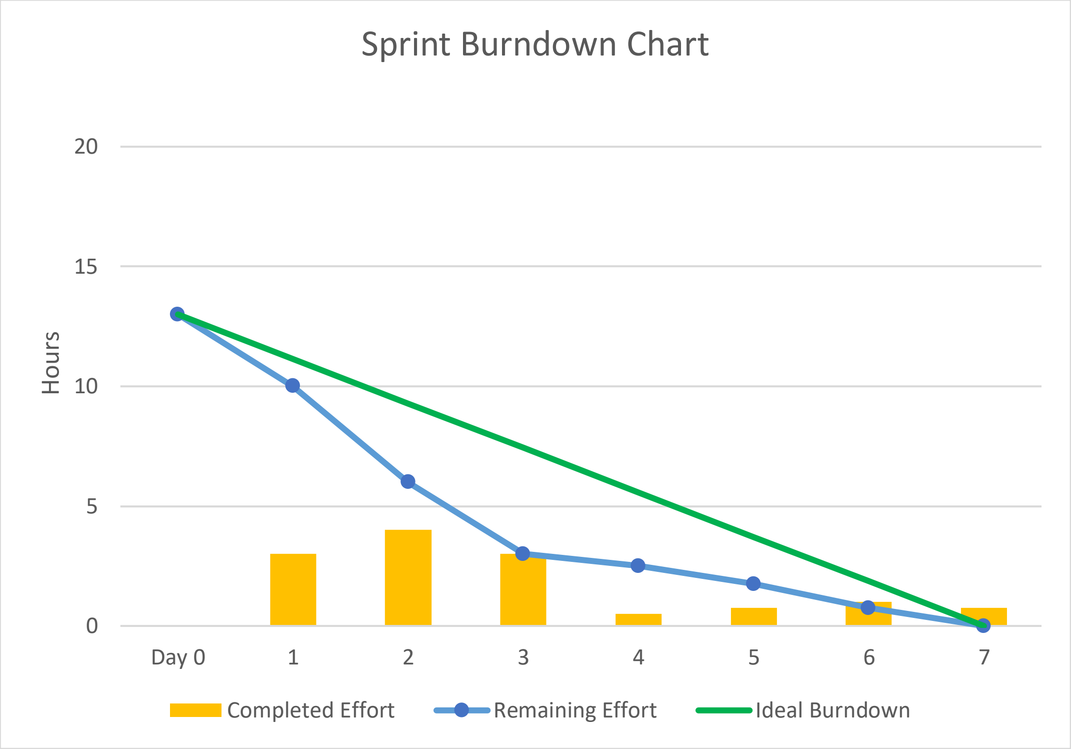
Todo	In Progress	Reviewing	Done
			Implement feature 1 (Ricardo)
			Implement feature 2 (James, João, Francisco, Iago)
			Upload sprints to github (Todo agente)
			Take metrics (Todo agente)

Burndown chart



Sprint 4 Burndown Chart

Task ID	Task Description	Initial Estimate	28/Nov/22	29/Nov/22	30/Nov/22	1/Dec/22	2/Dec/22	3/Dec/22	4/Dec/22
		Day 0	1	2	3	4	5	6	7
1	Implement feature 1 (Ricardo)	5	2	2	1				
2	Implement feature 2 (James, João, Francisco, Iago)	5	1	2	2				
3	Upload sprints to github (Todo agente)	2				0.5	0.5	0.5	0.5
4	Take metrics (Toda agente)	1					0.25	0.5	0.25
Completed Effort		0	3	4	3	0.5	0.75	1	0.75
Remaining Effort		13	10.0	6.0	3.0	2.5	1.8	0.8	0.0
Ideal Burndown		13	11.1	9.3	7.4	5.6	3.7	1.9	0.0



Produced content

Each one of us did a report about the metrics that is in the folder Phase2/Sprint4/<member_name>. Since it's too big and it would mess up with our document format, we decided not to include such reports in this final report. We will only include the reviews the name of the metrics each one of us chose to do.

Metric: Mood Metrics Review
Responsible: Francisco Vasco
Review: This evaluation makes it seem like the project is acceptable in the metrics regarded, being the worst score by far the 75.5% in the AIF.
This doesn't really make the project harder to understand, harder to change or any less performant **by** Ricardo Goncalo.

Metric: Line of code
Responsible: Iago Paulo
Review: The conclusions Iago has reached from the collected metrics agree with the consensus of the team over the development period of the project,
that is the project has a severe lack of comments which makes it more challenging to understand and improve **by** Francisco Vasco.

Metric: Dependency
Responsible: James Furtado
Review: There are classes that are really accoupled and this makes it hard to extend the project with new features. (Which means that we had some troubles trying to implement the new features) **by** Iago Paulo.

Metric: Chidamber and Kemerer
Responsible: João Oliveira
Review: I could see that he did some research and I agree with him that there is some values that are a bit high **by** James Furtado.

Metric: Complexity
Responsible: Ricardo Gonçalo
Review: The report is complete. Contains the average, sum, maximum and minimum.
Each metric is well explained **by** João Oliveira.