Software Engineering
2022/23

Gantt Project
Mergedoc deliverable

Students:

- Diogo Dias, 59731
- João Ribeiro, 60127
- Pedro Gasparinho, 60590
- Tiago Meirim, 60811
- Afonso Nunes, 64862

Chapter 1 - Patterns

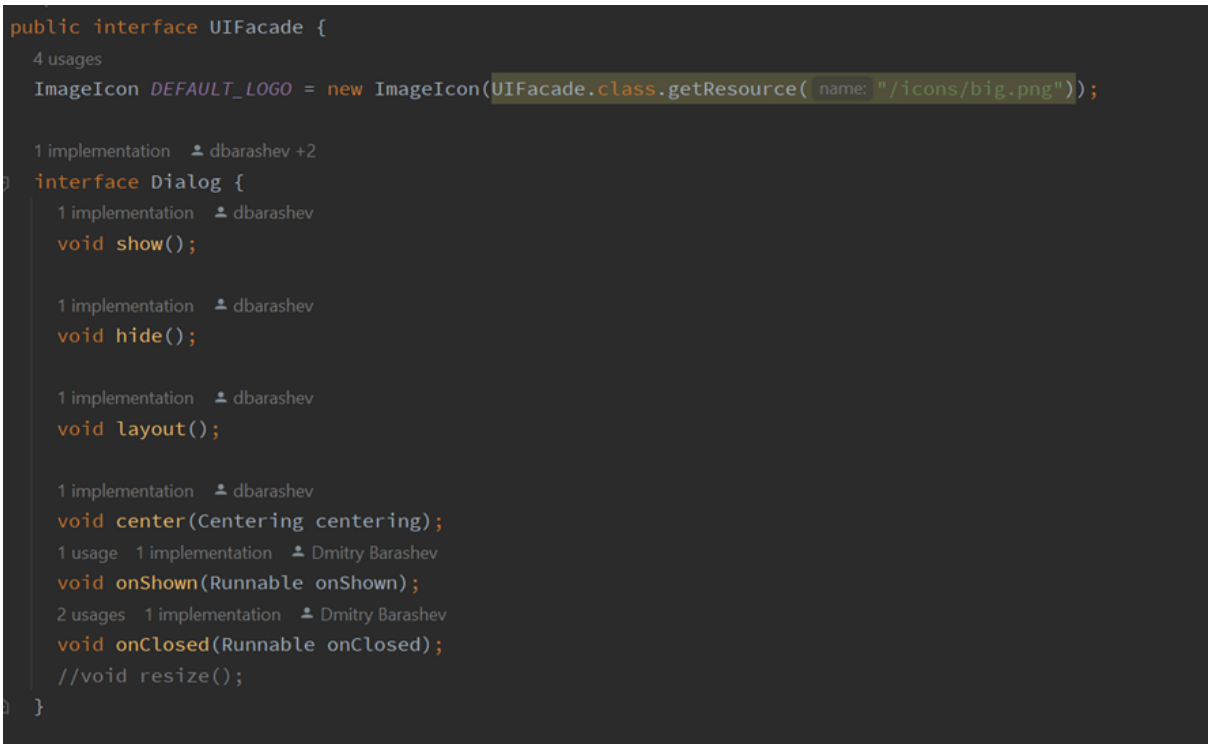
Diogo Dias' Patterns

Pattern 1 - Façade

Location - ganttproject\ganttproject\src\main\java\net\sourceforge\ganttproject\gui\UIfacade

Motive - In this case, UIFacade is an interface that combines features that share a common goal, particularly, to show a message on screen. This interface can offer an abstraction of these features that can be used in the system, without knowing the complexity of the implementation.

Code snippet -

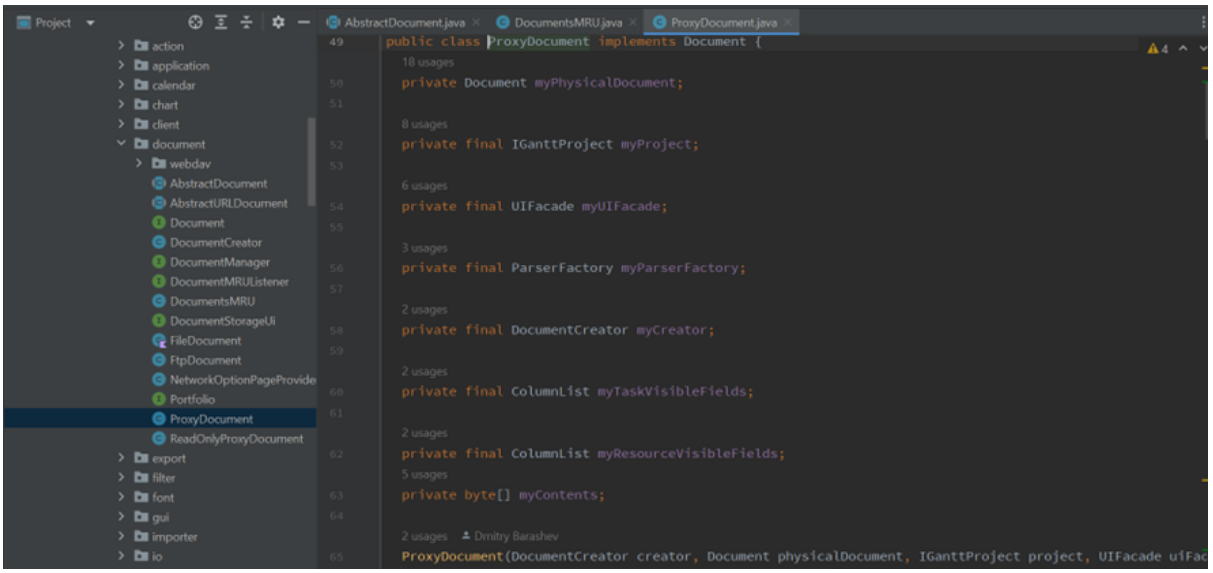


Pattern 2 - Proxy

Location - ganttproject\src\main\java\net\sourceforge\ganttproject\document\ProxyDocument.java

Motive - The proxyDocument class is a simplified implementation of the Document interface. It delegates some requests to other non-proxy Document objects that can fulfill them, for example the getFileName() method, which returns the document's name using another class that extends Document. The proxyDocument class can also perform some tasks by itself, such as the write() method, which writes bytes in a file.

Code snippet -

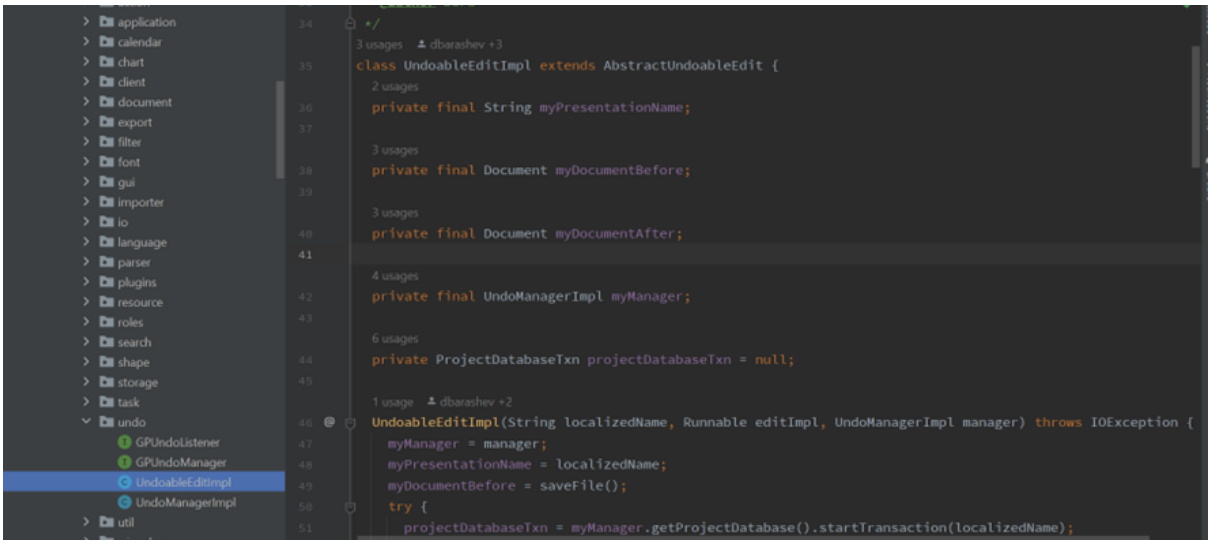


Pattern 3 - Memento Pattern

Location - ganttproject\src\main\java\net\sourceforge\ganttproject\undo\UndoableEditImpl.java

Motive - This class allows the user to save his document or require an old version of his document.

Code snippet -



João Ribeiro's Patterns

Pattern 1 - Builder

Location - ganttproject/src/main/java/net/sourceforge/ganttproject/gui/options/OptionsPageBuilder.java

Motive - OptionsPageBuilder is a builder class that allows the construction of complex and different objects step by step using the same construction code.

Code snippet -

```java
public class OptionsPageBuilder {
  22 usages
  I18N myi18n = new I18N();
  1 usage
  private Component myParentComponent;
  2 usages
  private final LayoutApi myLayoutApi;
  3 usages
  private UIFacade myUiFacade;
  private DecimalFormat myFormat;

  6 usages  2 implementations  dbarashev
  public static interface LayoutApi {
    2 implementations  dbarashev
    void layout(JPanel panel, int componentsCount);
  }

  4 usages  dbarashev
  public static LayoutApi TWO_COLUMN_LAYOUT = new LayoutApi() {
    dbarashev
    @Override
    public void layout(JPanel panel, int componentsCount) {
      panel.setLayout(new SpringLayout());
      SpringUtilities.makeCompactGrid(panel, componentsCount, cols: 2, initialX: 0, initialY: 0, xPad: 5, yPad: 3);
    }
  };

  1 usage  dbarashev
  public static LayoutApi ONE_COLUMN_LAYOUT = new LayoutApi() {
```

Pattern 2 - Prototype

Location - ganttproject/src/main/java/net/sourceforge/ganttproject/gui/EditableList.java

Motive - This method, located in the EditableList class, allows for the creation of copies of existing objects without making the code dependent on their original classes.

Code snippet -

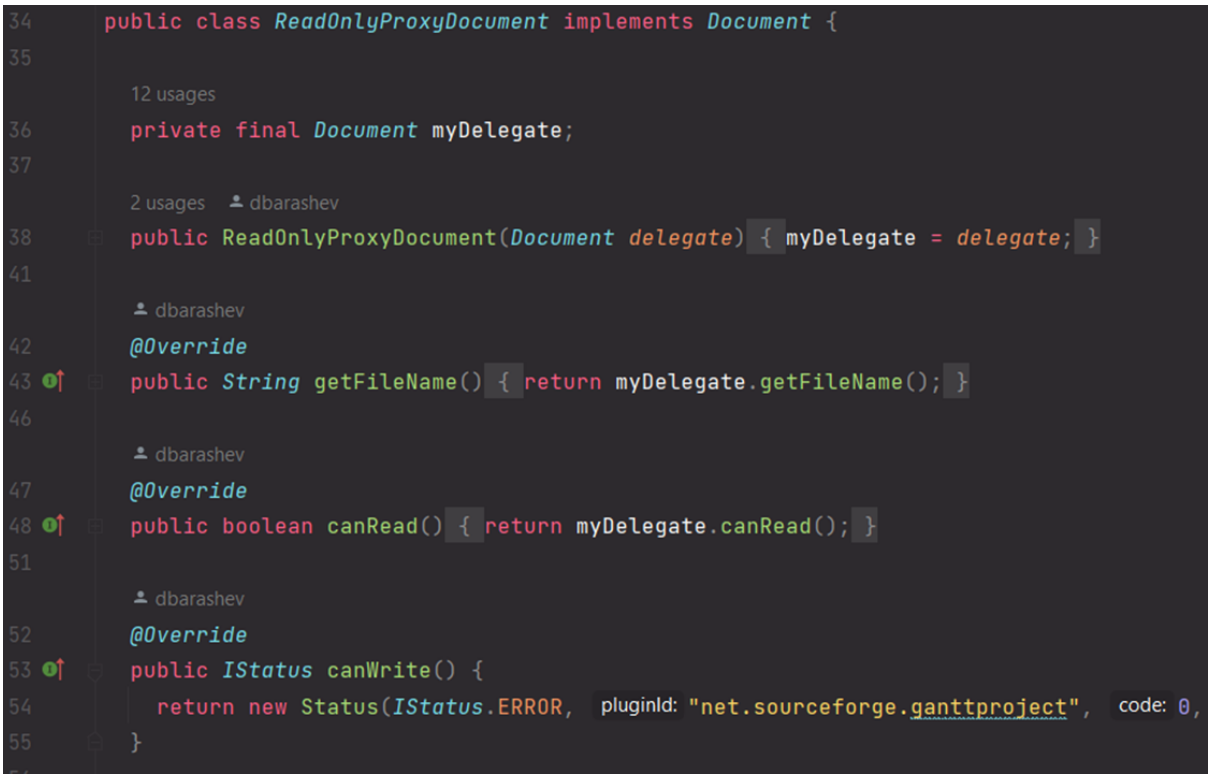
```
protected T createPrototype(Object editValue) {
  ComboItem setItem = null;
  if (ComboItem.class.equals(editValue.getClass())) {
    setItem = (ComboItem) editValue;
  } else {
    for (int i = 0; i < myComboBox.getModel().getSize(); i++) {
      if (((ComboItem) myComboBox.getModel().getElementAt(i)).myText.equals(editValue)) {
        setItem = (ComboItem) myComboBox.getModel().getElementAt(i);
        break;
      }
    }
  }
  return setItem == null ? null : setItem.myObject;
}
```

Pattern 3 - Proxy

Location - ganttproject/src/main/java/net/sourceforge/ganttproject/document/ReadOnlyProxyDocument.java

Motive - ReadOnlyProxyDocument is a proxy class that permits the usage of itself in order to allow other classes of the Document type to be used in codes that normally wouldn't be accepted. For example, this proxy is used to surround a Document in the usage of the openProject() method inside the recover method in the HelpMenu class.

Code snippet -



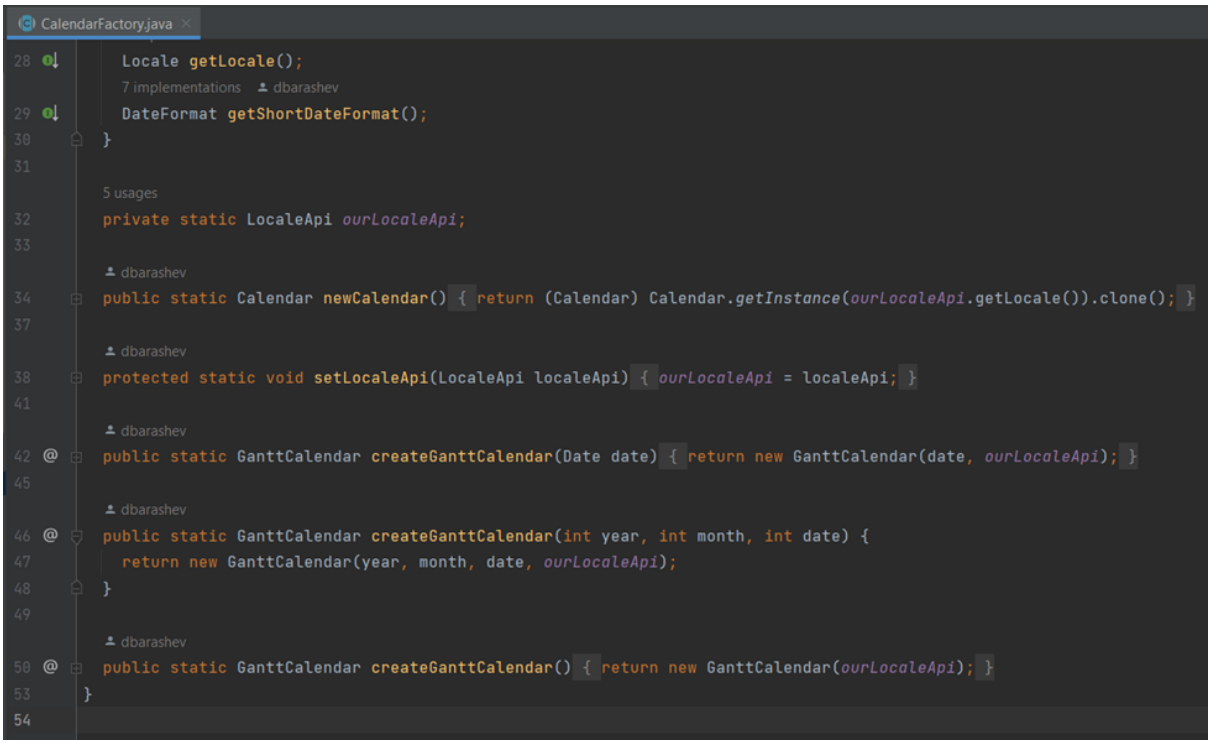
Pedro Gasparinho's Patterns

Pattern 1 - Factory

Location - ganttproject\biz.ganttproject.core\src\main\java\biz.ganttproject\core\time\CalenderFactory.

Motive - This abstract class is used to create GanttCalenders given different arguments. This helps to hide the complete process behind the creation of a GanttCalender to the user, starting off with a predefined localeApi, but also giving the possibility to change localeApi's for users with more knowledge about the system.

Code snippet -



Pattern 2 - Façade

Location - ganttproject\ganttproject\src\main\java\net.sourceforge.ganttproject\export\ConsoleUIFacade

Motive - ConsoleUIFacade is a façade class used to hide complex details about the system while also formatting the outputs and other functionalities in a more user-friendly way to be used in the console UI.

Code snippet -

ConsoleUIFacade.java

```
3 usages  dbarashev +1
51  public ConsoleUIFacade(UIFacade realFacade) { myRealFacade = realFacade; }
54

 Dmitry Barashev
55  @Override
56  public IntegerOption getDpiOption() { return null; }
59

4 usages  Dmitry Barashev
60  @Override
61  public GPOption<String> getLafOption() { return null; }
64

5 usages  dbarashev
65  @Override
66  public ScrollingManager getScrollingManager() {
67    // TODO Auto-generated method stub
68    return null;
69  }
70

 dbarashev +1
71  @Override
72  public ZoomManager getZoomManager() {
73    return myRealFacade != null ? myRealFacade.getZoomManager() : new ZoomManager(new GPTimeUnitStack());
74  }
75

4 usages  dbarashev
76  @Override
77  public ZoomActionSet getZoomActionSet() { return myRealFacade.getZoomActionSet(); }
80

3 usages  dbarashev
81  @Override
82  public Choice showConfirmationDialog(String message, String title) {
```

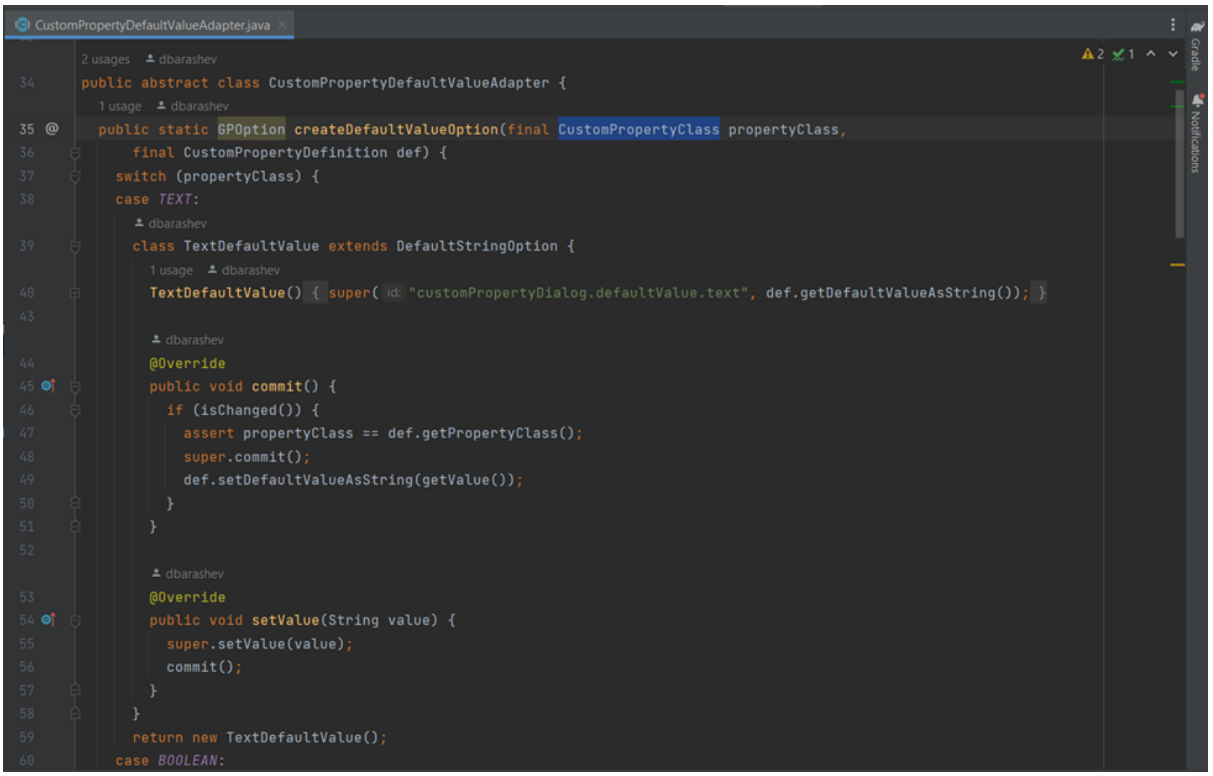
```
5 usages  dbarashev
104 @Override
105 public void showOptionDialog(int messageType, String message, Action[] actions) {
106   System.err.println("[ConsoleUIFacade]: " + message);
107 }
108

 dbarashev
109 @Override
110 public void showErrorDialog(String errorMessage) { System.err.println("[ConsoleUIFacade] ERROR: " + errorMessage); }
113

4 usages  dbarashev
114 @Override
115 public void showNotificationDialog(NotificationChannel channel, String message) {
116   System.err.println("[ConsoleUIFacade] " + channel.toString() + ": " + message);
117 }
118

 dbarashev
119 @Override
120 public void showErrorDialog(Throwable e) {
121   System.err.println("[ConsoleUIFacade] ERROR: " + e.getMessage());
122   e.printStackTrace();
123 }
124
```

Pattern 3 - Adapter

Location - ganttproject\ganttproject\src\main\java\net.sourceforge.ganttproject\gui\options\model\CustomPropertyDefaultValueAdapter

Motive - Given a CustomPropertyClass object and a CustomPropertyDefinition object this class implements the ability to adapt the default value from the CustomPropertyDefinition object to a GPOption implemented class base on a primitive type given by the CustomPropertyClass object. While also providing a commit and a setValue method.

Code snippet -



Tiago Meirim's Patterns

Pattern 1 - Builder

Location - biz.ganttproject.core/src/main/java/biz/ganttproject/core/chart/grid/OffsetBuilderImpl.java

Motive - Although this pattern is not yet explored and explained by the teacher, with a little googling I found that this class fit perfectly in the builder pattern. The class lets you build the different types of objects offsets step by step, like the method constructOffsets, by exporting the object construction code, in this case from the offset, and using it in the object builders/constructors.

Code snippet -

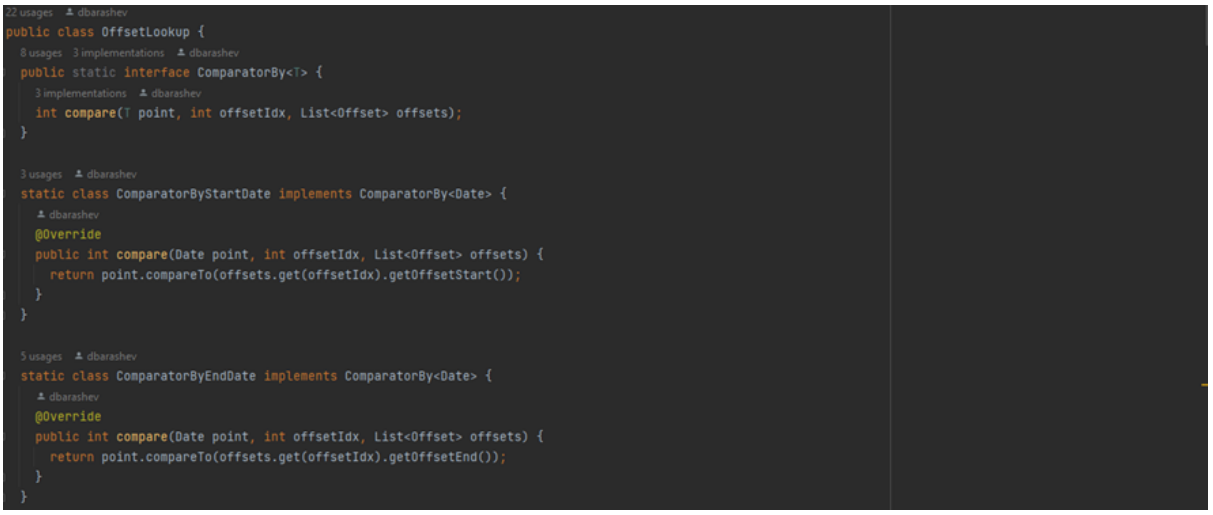
```
//  protected RegularFrameOffsetBuilder(GPCalendar calendar, TimeUnit topUnit, TimeUnit bottomUnit, Date startDate,
//      Date viewportStartDate, int defaultUnitWidth, int chartWidth, float weekendDecreaseFactor, Date endDate,
//      int rightMarginTimeUnits) {
  1 usage   dbarashev
protected OffsetBuilderImpl(OffsetBuilder.Factory factory) {
  myCalendar = factory.myCalendar;
  myStartDate = factory.myStartDate;
  myViewportStartDate = factory.myViewportStartDate;
  myTopUnit = factory.myTopUnit;
  myBottomUnit = factory.myBottomUnit;
  myDefaultUnitWidth = factory.myAtomicUnitWidth;
  myChartWidth = factory.myEndOffset;
  myWeekendDecreaseFactor = factory.myWeekendDecreaseFactor;
  myEndDate = factory.myEndDate;
  baseUnit = factory.myBaseUnit;
  myRightMarginBottomUnitCount = factory.myRightMarginTimeUnits;
  myOffsetStepFn = factory.myOffsetStepFn;
}
```

Pattern 2 - Façade

Location - biz.ganttproject.core/src/main/java/biz/ganttproject/core/chart/grid/OffsetLookup.java

Motive - The class creates other comparator classes that implement the same interface but for different types that are used as arguments when creating the class. Because both classes use the same name to compare the only difference between them is the type.

Code snippet -



```
22 usages  dbarashev
public class OffsetLookup {
  8 usages  3 implementations  dbarashev
  public static interface ComparatorBy<T> {
    3 implementations  dbarashev
    int compare(T point, int offsetIdx, List<Offset> offsets);
  }

  3 usages  dbarashev
  static class ComparatorByStartDate implements ComparatorBy<Date> {
    dbarashev
    @Override
    public int compare(Date point, int offsetIdx, List<Offset> offsets) {
      return point.compareTo(offsets.get(offsetIdx).getOffsetStart());
    }
  }

  5 usages  dbarashev
  static class ComparatorByEndDate implements ComparatorBy<Date> {
    dbarashev
    @Override
    public int compare(Date point, int offsetIdx, List<Offset> offsets) {
      return point.compareTo(offsets.get(offsetIdx).getOffsetEnd());
    }
  }
```

Pattern 3 - Factory

Location - biz.ganttproject.core/src/main/java/biz/ganttproject/core/chart/grid/OffsetBuilderImpl.java

Motive - This class is hiding the creation of instances of the given type behind an interface, so the client doesn't depend on concrete classes and one class can be easily replaceable or added. In this specific implementation the class builds grid offsets for timelines where top cells are always constructed from the integer number of bottom cells.

Code snippet -

```
5 usages  dbarashev
public static class FactoryImpl extends OffsetBuilder.Factory {
  dbarashev
  @Override
  public OffsetBuilder build() {
    preBuild();
    return new OffsetBuilderImpl( factory: this);
  }
}
}
```

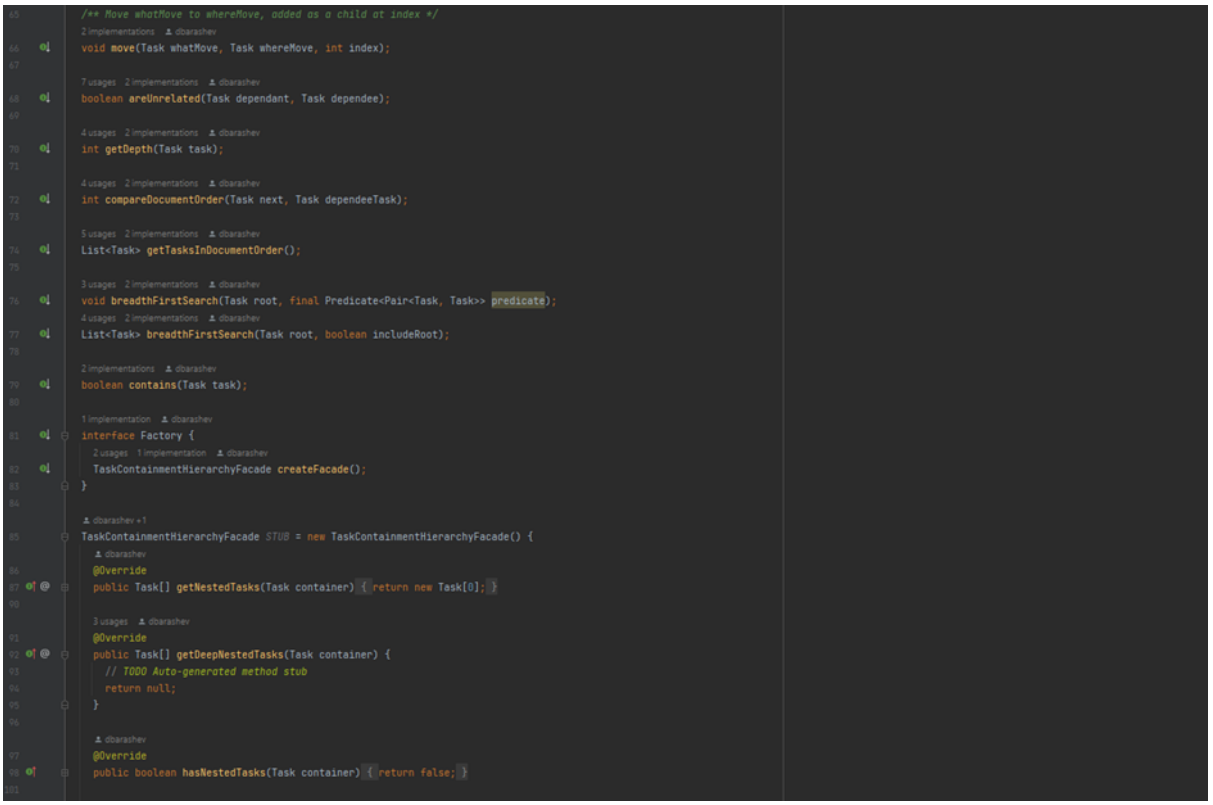
Afonso Nunes's Patterns

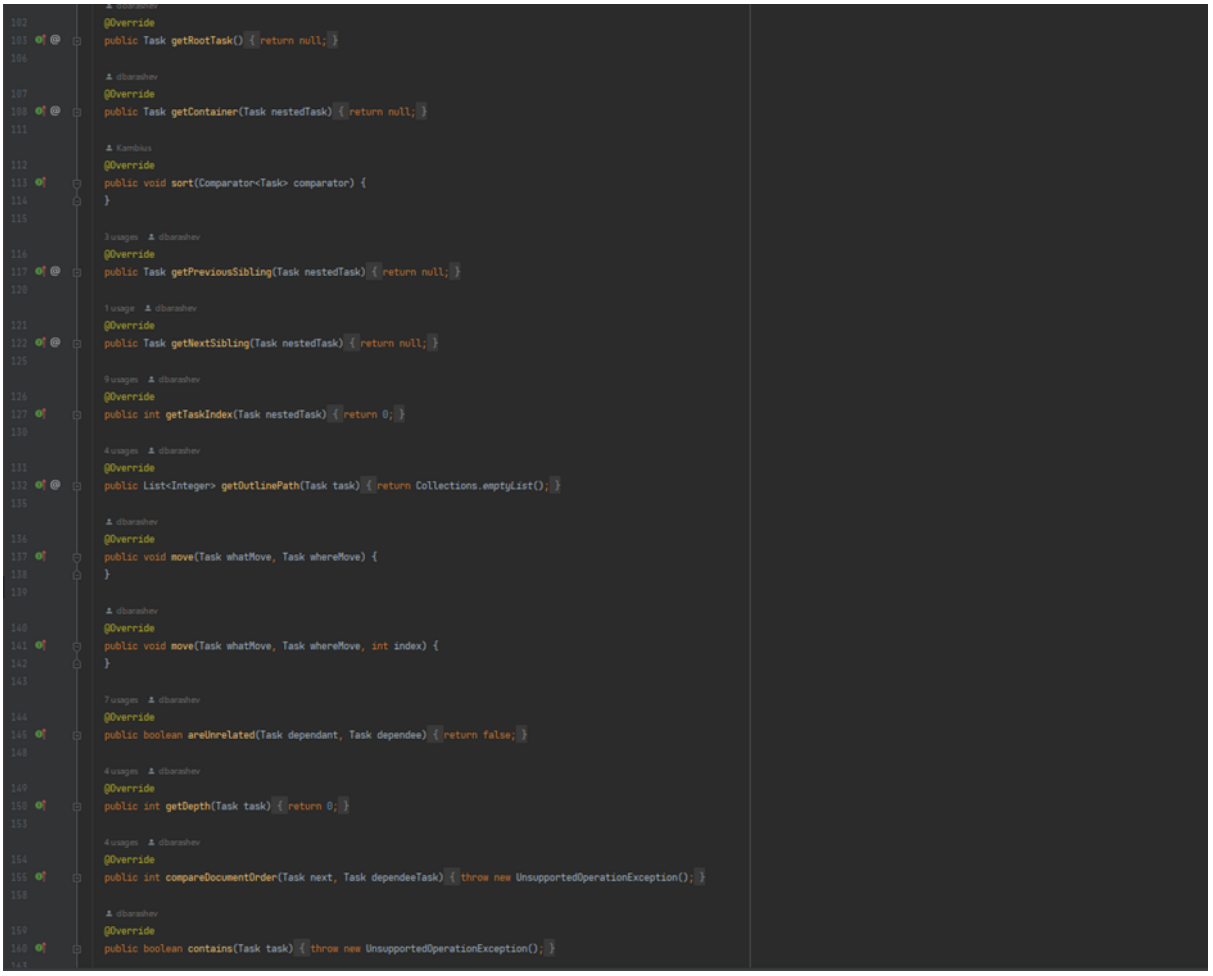
Pattern 1 - Façade

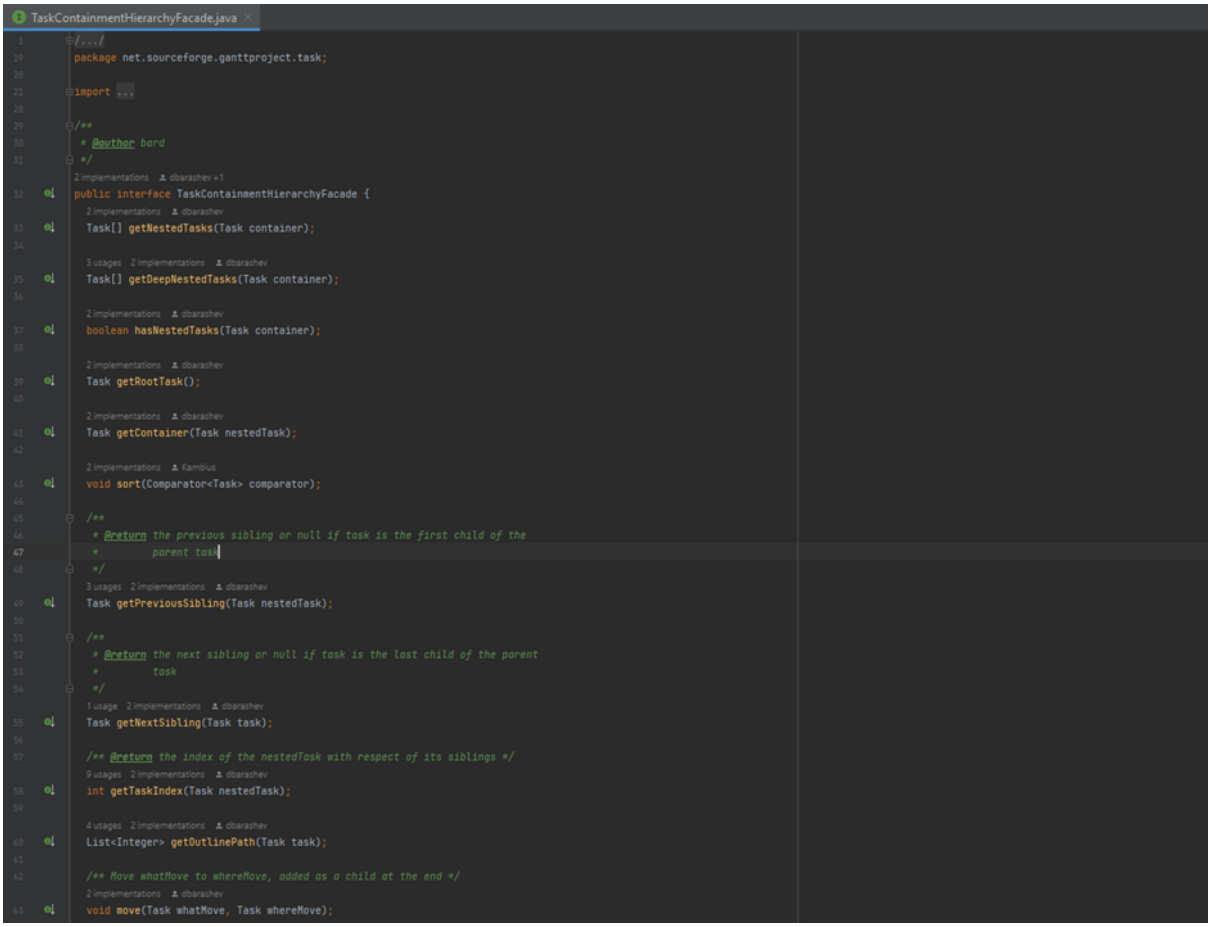
Location - ganttproject/src/main/java/net/sourceforge/ganttproject/task/TaskContainmentHierarchyFacade.java

Motive - After a discussion with my colleague and reviewing the code a little bit more I came to terms with his review and can also agree that this interface is indeed used to encapsulate Tasks whose implementation is done in a tree and not any type of Task.

Code snippet -




@Override
public Task getRootTask() { return null; }
dbarashev
@Override
public Task getContainer(Task nestedTask) { return null; }
Kambius
@Override
public void sort(Comparator<Task> comparator) {
}
3 usages dbarashev
@Override
public Task getPreviousSibling(Task nestedTask) { return null; }
1 usage dbarashev
@Override
public Task getNextSibling(Task nestedTask) { return null; }
9 usages dbarashev
@Override
public int getTaskIndex(Task nestedTask) { return 0; }
4 usages dbarashev
@Override
public List<Integer> getOutlinePath(Task task) { return Collections.emptyList(); }
dbarashev
@Override
public void move(Task whatMove, Task whereMove) {
}
dbarashev
@Override
public void move(Task whatMove, Task whereMove, int index) {
}
7 usages dbarashev
@Override
public boolean areUnrelated(Task dependant, Task dependee) { return false; }
4 usages dbarashev
@Override
public int getDepth(Task task) { return 0; }
4 usages dbarashev
@Override
public int compareDocumentOrder(Task next, Task dependeeTask) { throw new UnsupportedOperationException(); }
dbarashev
@Override
public boolean contains(Task task) { throw new UnsupportedOperationException(); }

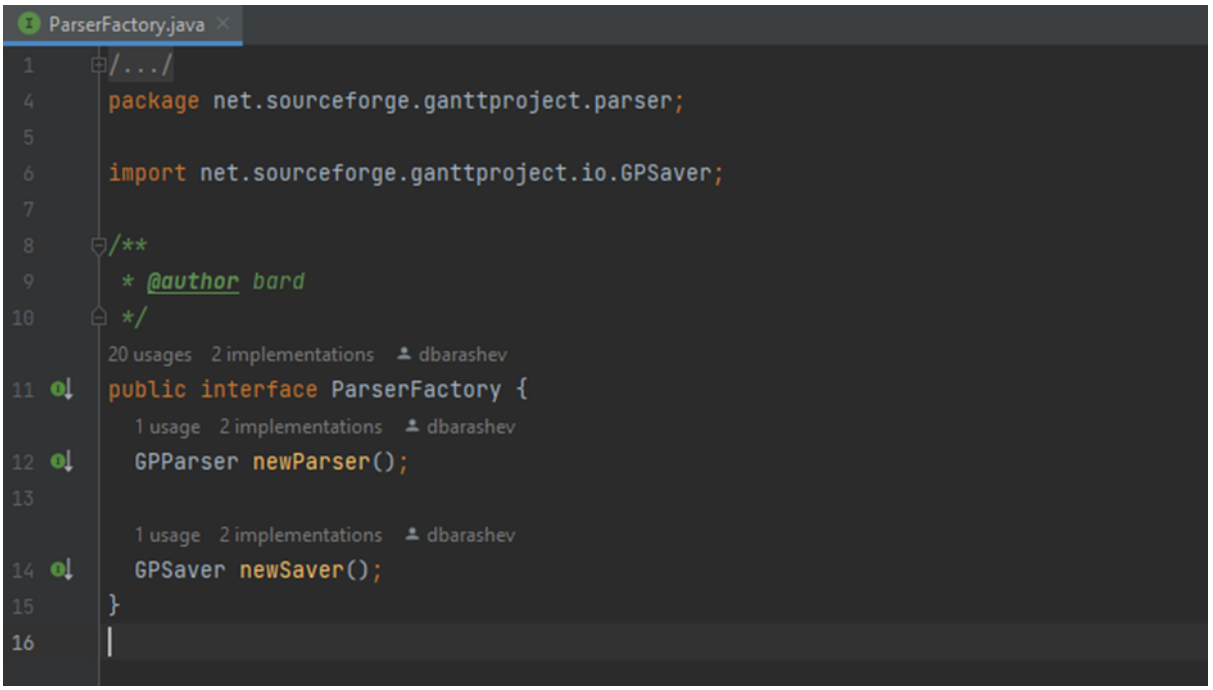


Pattern 2 - Factory

Location - ganttproject/src/main/java/net/sourceforge/ganttproject/parser/ParserFactory.java

Motive - This interface is used to hide the creation of instances of type Parser behind it making it so we can replace the concrete parser classes with others if we need to and it has no impact on the client code.

Code snippet -



Pattern 3 - Command

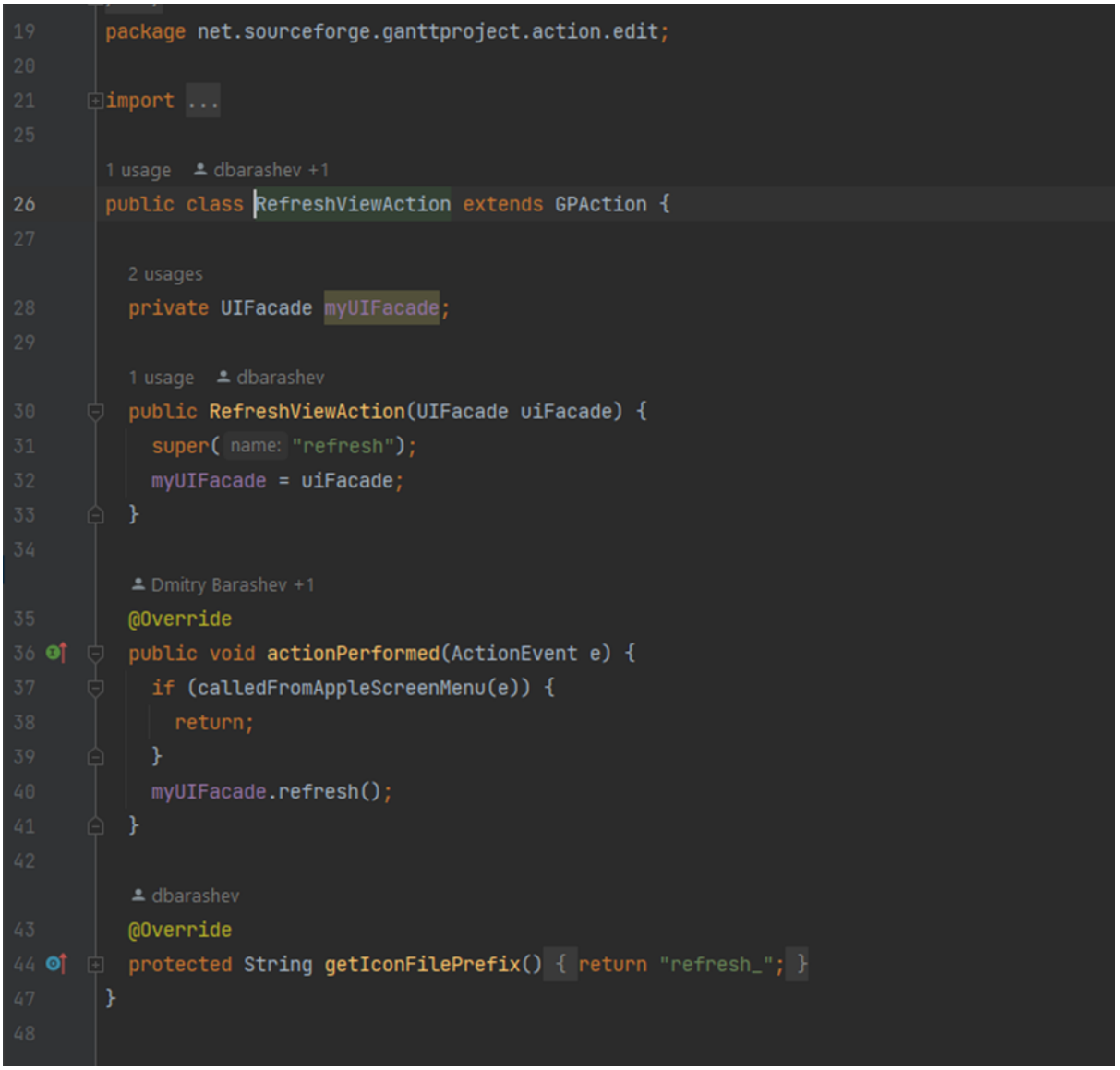
Location - ganttproject/src/main/java/net/sourceforge/ganttproject/action/edit

Motive - Since I didn't specify any of the operations, I can understand my colleague's review. As it can be **seen from the screenshots** there are many operations supported in this implementation of the command pattern and that are useful for the user.

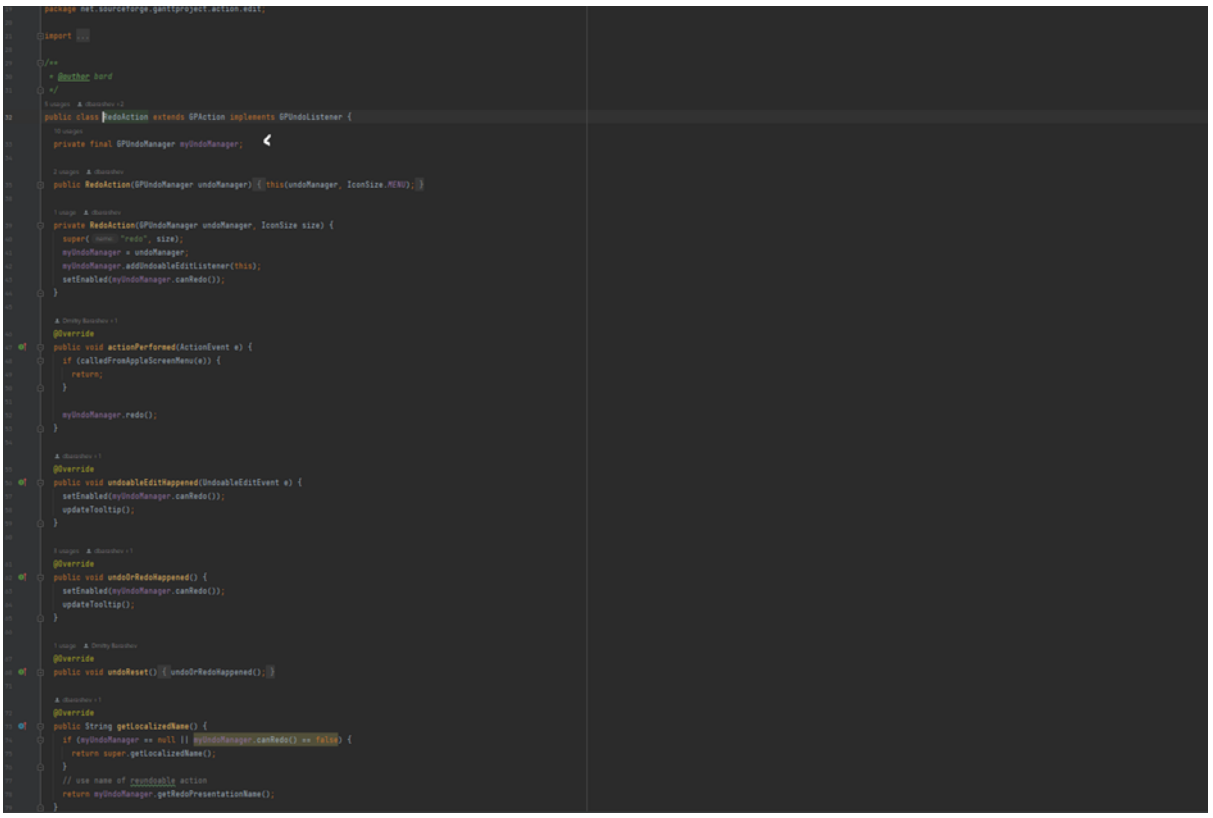
Code snippet -

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

import ...

/**
 * Action to show the options dialog for the application. It will seach and show
 * all available OptionPageProvider classes
 */
3 usages  dbarashev +2
public class SettingsDialogAction extends GPAction {
  2 usages
  private final UIFacade myUiFacade;
  2 usages
  private final IGanttProject myProject;

  2 usages  dbarashev
  public SettingsDialogAction(IGanttProject project, UIFacade uiFacade) {
    super( name: "settings.app");
    myUiFacade = uiFacade;
    myProject = project;
  }

  dbarashev +2
  @Override
  public void actionPerformed(ActionEvent e) {
    if (calledFromAppleScreenMenu(e)) {
      return;
    }
    SettingsDialog2 dialog = new SettingsDialog2(myProject, myUiFacade, pageOrderKey: "settings.app.pageOrder", titleKey: "settings.app");
    dialog.show();
  }
}
```

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

import ...

2 usages  dbarashev +1
public class SearchDialogAction extends GPAction {
  2 usages
  private final Runnable mySearchUi;

  1 usage  dbarashev +1
  public SearchDialogAction(Runnable searchUi) {
    super( name: "search.dialog.open");
    mySearchUi = searchUi;
  }

  dbarashev +1
  @Override
  public void actionPerformed(ActionEvent e) { mySearchUi.run(); }
}
```

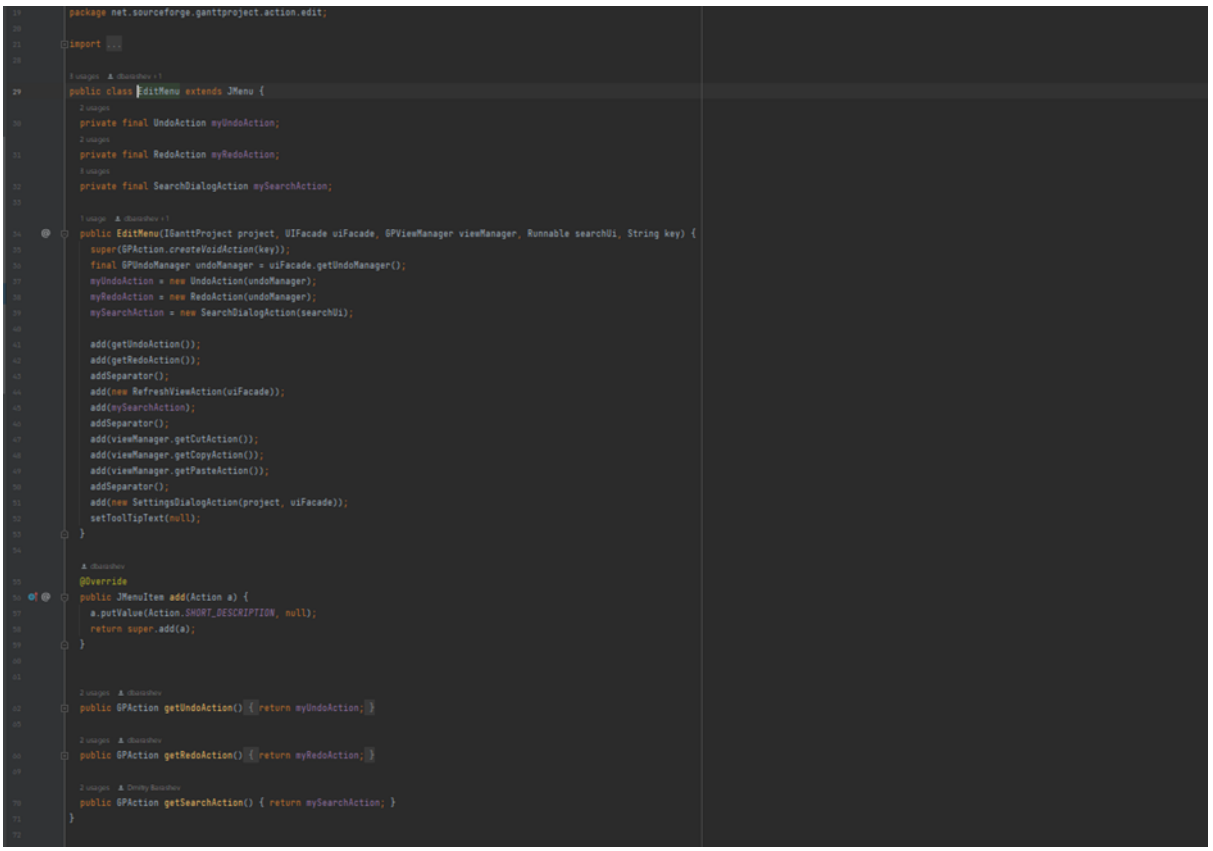


```java
package net.sourceforge.ganttproject.action.edit;

import ...

1 usage  dbarashev +1
public class RefreshViewAction extends GPAction {

  2 usages
  private UIFacade myUIFacade;

  1 usage  dbarashev
  public RefreshViewAction(UIFacade uiFacade) {
    super( name: "refresh");
    myUIFacade = uiFacade;
  }

  Dmitry Barashev +1
  @Override
  public void actionPerformed(ActionEvent e) {
    if (calledFromAppleScreenMenu(e)) {
      return;
    }
    myUIFacade.refresh();
  }

  dbarashev
  @Override
  protected String getIconFilePrefix() { return "refresh_"; }
}
```



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

import ...

/**
 * @author bard
 */
5 usages  dbarashev +2
public class RedoAction extends GPAction implements GPUndoListener {
  10 usages
  private final GPUndoManager myUndoManager;

  2 usages  dbarashev
  public RedoAction(GPUndoManager undoManager) { this(undoManager, IconSize.MENU); }

  1 usage  dbarashev
  private RedoAction(GPUndoManager undoManager, IconSize size) {
    super( name: "redo", size);
    myUndoManager = undoManager;
    myUndoManager.addUndoableEditListener(this);
    setEnabled(myUndoManager.canRedo());
  }

  Dmitry Barashev +1
  @Override
  public void actionPerformed(ActionEvent e) {
    if (calledFromAppleScreenMenu(e)) {
      return;
    }

    myUndoManager.redo();
  }

  dbarashev +1
  @Override
  public void undoableEditHappened(UndoableEditEvent e) {
    setEnabled(myUndoManager.canRedo());
    updateTooltip();
  }

  3 usages  dbarashev +1
  @Override
  public void undoOrRedoHappened() {
    setEnabled(myUndoManager.canRedo());
    updateTooltip();
  }

  1 usage  Dmitry Barashev
  @Override
  public void undoReset() { undoOrRedoHappened(); }

  dbarashev +1
  @Override
  public String getLocalizedName() {
    if (myUndoManager == null || myUndoManager.canRedo() == false) {
      return super.getLocalizedName();
    }
    // use name of redoable action
    return myUndoManager.getRedoPresentationName();
  }
}
```

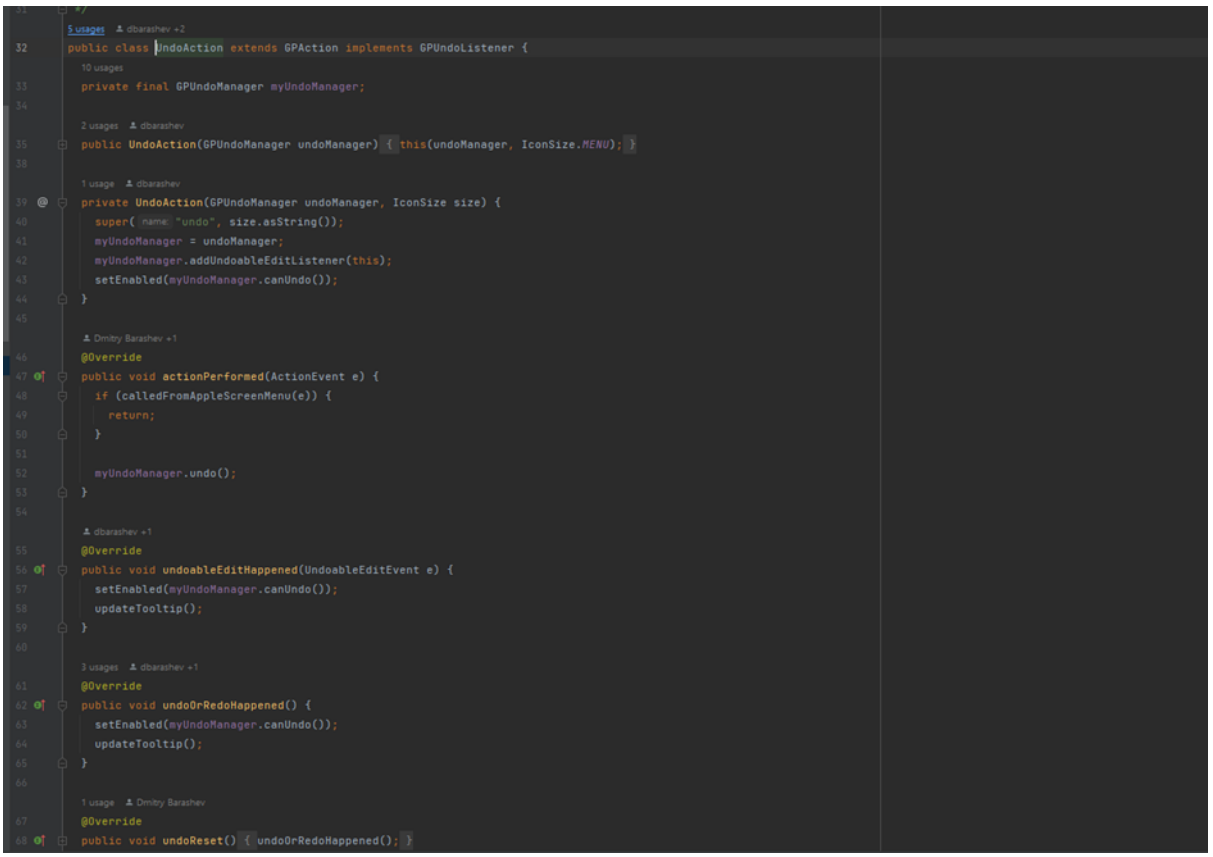
```java
package net.sourceforge.ganttproject.action.edit;

import ...

//TODO Enable/Disable action depending on clipboard contents
6 usages  Your Name +4
public class PasteAction extends GPAction {
  5 usages
  private final GPViewManager myViewmanager;
  5 usages
  private final GPUndoManager myUndoManager;
  5 usages
  private final IGanttProject myProject;
  5 usages
  private final UIFacade myUiFacade;

  2 usages  dbarashev +1
  public PasteAction(IGanttProject project, UIFacade uiFacade, GPViewManager viewManager, GPUndoManager undoManager) {
    super( name: "paste");
    myViewmanager = viewManager;
    myUndoManager = undoManager;
    myProject = project;
    myUiFacade = uiFacade;
  }

  dbarashev +1
  private PasteAction(IGanttProject project, UIFacade uiFacade, GPViewManager viewmanager, IconSize size, GPUndoManager undoManager) {
    super( name: "paste", size);
    myViewmanager = viewmanager;
    myUndoManager = undoManager;
    myProject = project;
    myUiFacade = uiFacade;
  }

  Your Name +1
  @Override
  public void actionPerformed(ActionEvent evt) {
    if (calledFromAppleScreenMenu(evt)) {
      return;
    }
    ChartSelection selection = myViewmanager.getSelectedArtefacts();
    if (!selection.isEmpty()) {
      pasteInternalFlavor(selection);
      return;
    }
    var clipboardProject  = ClipboardContentsKt.getProjectFromClipboard(new BufferProject(myProject, myUiFacade));
    if (clipboardProject != null) {
      try {
        myUndoManager.undoableEdit(getLocalizedName(), () -> pasteExternalDocument(clipboardProject));
      } catch (Exception e) {
        e.printStackTrace();
      }
    }
  }
```



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

import ...

3 usages  dbarashev + 1
public class EditMenu extends JMenu {
  2 usages
  private final UndoAction myUndoAction;
  2 usages
  private final RedoAction myRedoAction;
  3 usages
  private final SearchDialogAction mySearchAction;

  1 usage  dbarashev + 1
  public EditMenu(IGanttProject project, UIFacade uiFacade, GPViewManager viewManager, Runnable searchUi, String key) {
    super(GPAction.createVoidAction(key));
    final GPUndoManager undoManager = uiFacade.getUndoManager();
    myUndoAction = new UndoAction(undoManager);
    myRedoAction = new RedoAction(undoManager);
    mySearchAction = new SearchDialogAction(searchUi);

    add(getUndoAction());
    add(getRedoAction());
    addSeparator();
    add(new RefreshViewAction(uiFacade));
    add(mySearchAction);
    addSeparator();
    add(viewManager.getCutAction());
    add(viewManager.getCopyAction());
    add(viewManager.getPasteAction());
    addSeparator();
    add(new SettingsDialogAction(project, uiFacade));
    setToolTipText(null);
  }

  dbarashev
  @Override
  public JMenuItem add(Action a) {
    a.putValue(Action.SHORT_DESCRIPTION, null);
    return super.add(a);
  }


  2 usages  dbarashev
  public GPAction getUndoAction() { return myUndoAction; }

  2 usages  dbarashev
  public GPAction getRedoAction() { return myRedoAction; }

  2 usages  Dmitry Barashev
  public GPAction getSearchAction() { return mySearchAction; }
}
```

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

import ...

//TODO Enable/Disable action on selection changes
6 usages  Dmitry Barashev +2
public class CopyAction extends GPAction {
  3 usages
  private final GPViewManager myViewmanager;

  2 usages  dbarashev
  public CopyAction(GPViewManager viewManager) {
    super( name: "copy");
    myViewmanager = viewManager;
  }

  dbarashev +1
  @Override
  public void actionPerformed(ActionEvent e) {
    if (calledFromAppleScreenMenu(e)) {
      return;
    }
    myViewmanager.getSelectedArtefacts().startCopyClipboardTransaction();
  }

  Dmitry Barashev
  @Override
  public CopyAction asToolbarAction() {
    final CopyAction result = new CopyAction(myViewmanager);
    result.setFontAwesomeLabel(UIUtil.getFontawesomeLabel(result));
    Dmitry Barashev
    this.addPropertyChangeListener(new PropertyChangeListener() {
      Dmitry Barashev
      @Override
      public void propertyChange(PropertyChangeEvent evt) {
        if ("enabled".equals(evt.getPropertyName())) {
          result.setEnabled((Boolean)evt.getNewValue());
        }
      }
    });
    result.setEnabled(this.isEnabled());
    return result;
  }
}
```

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

import ...

//TODO Enable/Disable action on selection changes
6 usages  Dmitry Barashev +3
public class CutAction extends GPAction {
  3 usages
  private final GPViewManager myViewmanager;
  2 usages
  private final GPUndoManager myUndoManager;

  2 usages  dbarashev
  public CutAction(GPViewManager viewManager, GPUndoManager undoManager) {
    super( name: "cut");
    myViewmanager = viewManager;
    myUndoManager = undoManager;
  }

  Dmitry Barashev +2
  @Override
  public void actionPerformed(ActionEvent e) {
    if (calledFromAppleScreenMenu(e)) {
      return;
    }
//    myUndoManager.undoableEdit(getLocalizedName(), new Runnable() {
//      @Override
//      public void run() {
        myViewmanager.getSelectedArtefacts().startMoveClipboardTransaction();
//      }
    //});
  }

  Dmitry Barashev
  @Override
  public CutAction asToolbarAction() {
    final CutAction result = new CutAction(myViewmanager, myUndoManager);
    result.setFontAwesomeLabel(UIUtil.getFontawesomeLabel(result));
    Dmitry Barashev
    this.addPropertyChangeListener(new PropertyChangeListener() {
      Dmitry Barashev
      @Override
      public void propertyChange(PropertyChangeEvent evt) {
        if ("enabled".equals(evt.getPropertyName())) {
          result.setEnabled((Boolean)evt.getNewValue());
        }
      }
    });
    result.setEnabled(this.isEnabled());
    return result;
  }
}
```



```
5 usages  dbarashev +2
public class UndoAction extends GPAction implements GPUndoListener {
  10 usages
  private final GPUndoManager myUndoManager;

  2 usages  dbarashev
  public UndoAction(GPUndoManager undoManager) { this(undoManager, IconSize.MENU); }

  1 usage  dbarashev
  private UndoAction(GPUndoManager undoManager, IconSize size) {
    super( name: "undo", size.asString());
    myUndoManager = undoManager;
    myUndoManager.addUndoableEditListener(this);
    setEnabled(myUndoManager.canUndo());
  }

  Dmitry Barashev +1
  @Override
  public void actionPerformed(ActionEvent e) {
    if (calledFromAppleScreenMenu(e)) {
      return;
    }

    myUndoManager.undo();
  }

  dbarashev +1
  @Override
  public void undoableEditHappened(UndoableEditEvent e) {
    setEnabled(myUndoManager.canUndo());
    updateTooltip();
  }

  3 usages  dbarashev +1
  @Override
  public void undoOrRedoHappened() {
    setEnabled(myUndoManager.canUndo());
    updateTooltip();
  }

  1 usage  Dmitry Barashev
  @Override
  public void undoReset() { undoOrRedoHappened(); }
```

Chapter 2 - Code Smells

Diogo Dias' Code Smells

Code Smell 1 - Duplicated Code

Location:

- biz.ganttproject.core\src\main\java\biz\ganttproject\core\chart\renderer\RectangleRenderer.java
- biz.ganttproject.core\src\main\java\biz\ganttproject\core\chart\renderer\PolygonRenderer.java
- biz.ganttproject.core\src\main\java\biz\ganttproject\core\chart\renderer\LineRenderer.java
- biz.ganttproject.core\src\main\java\biz\ganttproject\core\chart\renderer\SummaryTaskRenderer.java

Motive - All these classes have the same variables, one method equal and one method that is different in each class but has the same name and same purpose.

Solution - It’s possible to eliminate this repetition using an abstract class, AbstractRenderer. This class should have the variables myProperties and myGraphics. In terms of methods, this class implements the method setGraphics and may have an abstract method, render, that allows each class’ son to implement this method.

Code snippet -

```java
public class SummaryTaskRenderer {
  2 usages
  private final Properties myProperties;
  2 usages
  private Graphics2D myGraphics;

  1 usage  Dmitry Kazakov
  public SummaryTaskRenderer(Properties props) { myProperties = props; }

  Dmitry Kazakov
  public void setGraphics(Graphics2D graphics) { myGraphics = graphics; }

  Dmitry Kazakov
  public void render(Canvas.Rectangle rect) {
    Graphics2D g = (Graphics2D) myGraphics.create();
    Style style = Style.getStyle(myProperties, rect.getStyle());

    Style.Color background = style.getBackgroundColor(rect);
    if (background != null) {
      g.setColor(background.get());
    }
    Style.Padding padding = style.getPadding();
    g.fillRect( x: rect.getLeftX() + padding.getLeft(),  y: rect.getTopY() + padding.getTop(),
```

```java
public class RectangleRenderer {
  2 usages
  private final Properties myProperties;
  2 usages
  private Graphics2D myGraphics;

  1 usage  dbarashev
  public RectangleRenderer(Properties props) { myProperties = props; }

  dbarashev
  public void setGraphics(Graphics2D graphics) { myGraphics = graphics; }

  dbarashev
  public boolean render(Canvas.Rectangle rect) {
    Graphics2D g = (Graphics2D) myGraphics.create();
    Style style = Style.getStyle(myProperties, rect.getStyle());

    if (style.getVisibility(rect) == Style.Visibility.HIDDEN) {
      return false;
    }
    Style.Color background = style.getBackgroundColor(rect);
    if (background != null) {
      g.setColor(background.get());
    }
    Paint paint = style.getBackgroundPaint(rect);
    if (paint != null) {
```

```java
public class PolygonRenderer {
  2 usages
  private final Properties myProperties;
  2 usages
  private Graphics2D myGraphics;

  1 usage  dbarashev
  public PolygonRenderer(Properties props) { myProperties = props; }

  dbarashev
  public void setGraphics(Graphics2D graphics) { myGraphics = graphics; }

  dbarashev
  public void render(Canvas.Polygon p) {
    Graphics2D g = (Graphics2D) myGraphics.create();
    Style style = Style.getStyle(myProperties, p.getStyle());

    if (style.getVisibility(p) == Style.Visibility.HIDDEN) {
      return;
    }
    Style.Color background = style.getBackgroundColor(p);
    if (background != null) {
      g.setColor(background.get());
    }
    g.fillPolygon(p.getPointsX(), p.getPointsY(), p.getPointCount());
  }
```

```java
public class LineRenderer {
  2 usages
  private final Properties myProperties;
  2 usages
  private Graphics2D myGraphics;

  1 usage  dbarashev
  public LineRenderer(Properties props) { myProperties = props; }

  dbarashev
  public void setGraphics(Graphics2D graphics) { myGraphics = graphics; }

  1 usage  dbarashev +1
  public void renderLine(Canvas.Line line) {
    Graphics2D g = (Graphics2D) myGraphics.create();
    Style style = Style.getStyle(myProperties, line.getStyle());

    Style.Borders border = style.getBorder(line);
    g.setColor(border == null ? java.awt.Color.BLACK : border.getTop().getColor());
    g.setStroke(border == null ? Style.DEFAULT_STROKE : border.getTop().getStroke());
    g.drawLine(line.getStartX(), line.getStartY(), line.getFinishX(), line.getFinishY());
    if (line.getArrow() != Canvas.Arrow.NONE) {
      Canvas.Arrow arrow = line.getArrow();
      int xsign = Integer.signum( i: line.getFinishX() - line.getStartX());
      int ysign = Integer.signum( i: line.getFinishY() - line.getStartY());
```

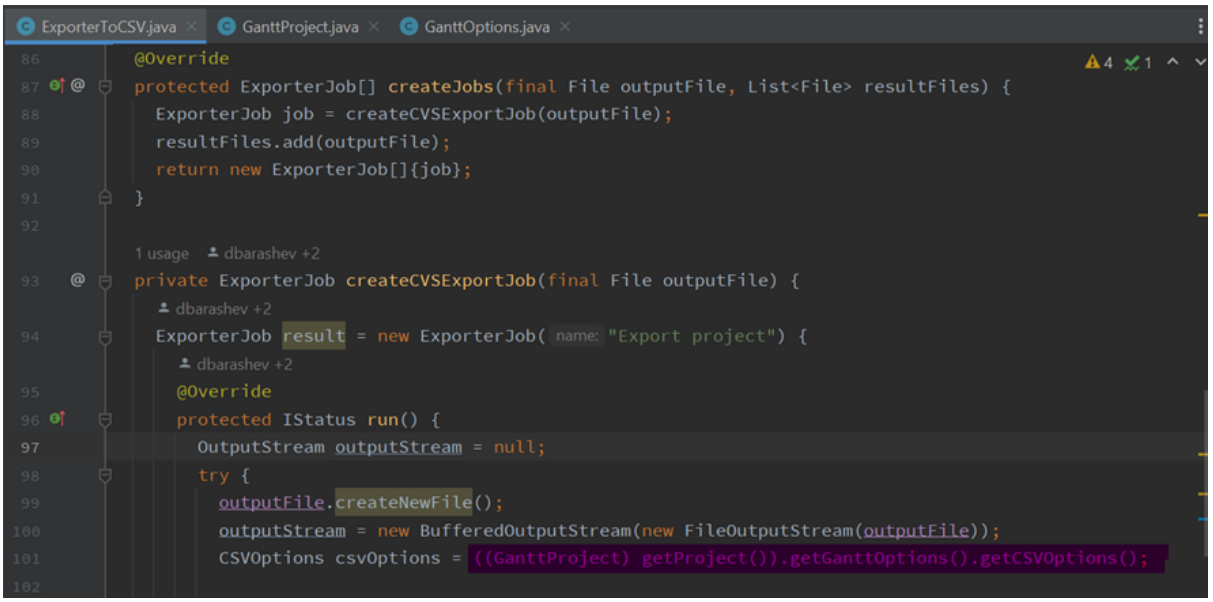
Code Smell 2 - Message Chain

Location - ganttproject\src\main\java\net\sourceforge\ganttproject\export\ExporterToCSV.java

Motive - This method is calling in chains methods that return other objects. The method getProject returns an IGanttProject. After that is called the method getGanttOptions that is implemented in IGanttProject, this method returns a GanttOptions. Finally, the object GanttOptions calls the method getCSVOptions which returns a CSVOptions object.

Solution - The way to resolve this problem the class IGanttProject may have a method that returns a CSVOptions object. After that, the class ExporterToCSV should use the method created in IGanttProject and create a new method that also returns CSVOptions. So we can use the class that we pretend to without knowing the complexity of the system.

Code snippet -



Code Smell 3 - Speculative Generality

Location - ganttproject\src\main\java\net\sourceforge\ganttproject\PreferenceServiceImpl.java

Motive - This class is not implemented.

Solution - This class should be eliminated because at this moment it is not necessary for this app.

Code snippet -

```
public class PreferenceServiceImpl implements IPreferencesService {

    dbarashev
    @Override
    public IStatus applyPreferences(IExportedPreferences preferences) throws CoreException {
        // TODO Auto-generated method stub
        return null;
    }

    dbarashev
    @Override
    public void applyPreferences(IEclipsePreferences node, IPreferenceFilter[] filters) throws CoreException
        // TODO Auto-generated method stub

    }
```

```
    dbarashev
    @Override
    public IStatus exportPreferences(IEclipsePreferences node, OutputStream output, String[] excludesList)
            throws CoreException {
        // TODO Auto-generated method stub
        return null;
    }

    dbarashev
    @Override
    public void exportPreferences(IEclipsePreferences node, IPreferenceFilter[] filters, OutputStream outp
            throws CoreException {
        // TODO Auto-generated method stub

    }

    dbarashev
    @Override
    public String get(String key, String defaultValue, Preferences[] nodes) {
        // TODO Auto-generated method stub
        return null;
    }
```

João Ribeiro's Code Smells

Code Smell 1 - Dead Code

Location - ganttproject/src/main/java/net/sourceforge/ganttproject/chart/ChartModelImpl.java

Motive - Commented code from previous versions that isn't being run and still is occupying space.

Solution - Delete the commented code.

Code snippet -

```
// public java.awt.Rectangle getBoundingRectangle(Task task) {
// java.awt.Rectangle result = null;
// TaskActivity[] activities = task.getActivities();
// for (int i = 0; i < activities.length; i++) {
// GraphicPrimitiveContainer.Rectangle nextRectangle = myTaskRendererImpl
// .getPrimitive(activities[i]);
// if (nextRectangle != null) {
// java.awt.Rectangle nextAwtRectangle = new java.awt.Rectangle(
// nextRectangle.myLeftX, nextRectangle.myTopY,
// nextRectangle.myWidth, nextRectangle.myHeight);
// if (result == null) {
// result = nextAwtRectangle;
// } else {
// result = result.union(nextAwtRectangle);
// }
// }
// }
// return result;
// }

// GraphicPrimitiveContainer.Rectangle[] getTaskActivityRectangles(Task task)
// {
// List<Rectangle> result = new ArrayList<Rectangle>();
// TaskActivity[] activities = task.getActivities();
// for (int i = 0; i < activities.length; i++) {
// GraphicPrimitiveContainer.Rectangle nextRectangle = myTaskRendererImpl
// .getPrimitive(activities[i]);
// if (nextRectangle!=null) {
// result.add(nextRectangle);
// }
// }
```

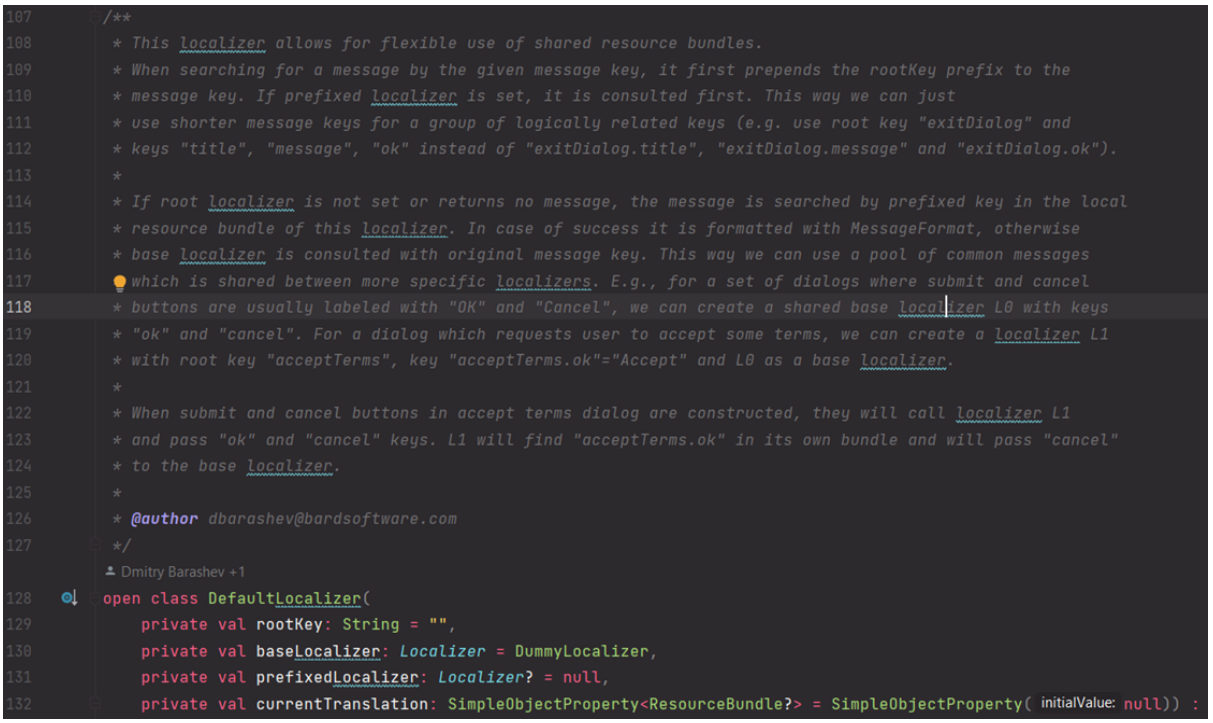
Code Smell 2 - Too many comments

Location - /ganttproject/src/main/java/biz.ganttproject/app/Internationalization.kt

Motive - Way too big of a comment for the size of the class.

Solution - Sum up some of the comments into a simpler and easy to read comment.

Code snippet -



```
/**
 * This localizer allows for flexible use of shared resource bundles.
 * When searching for a message by the given message key, it first prepends the rootKey prefix to the
 * message key. If prefixed localizer is set, it is consulted first. This way we can just
 * use shorter message keys for a group of logically related keys (e.g. use root key "exitDialog" and
 * keys "title", "message", "ok" instead of "exitDialog.title", "exitDialog.message" and "exitDialog.ok").
 *
 * If root localizer is not set or returns no message, the message is searched by prefixed key in the local
 * resource bundle of this localizer. In case of success it is formatted with MessageFormat, otherwise
 * base localizer is consulted with original message key. This way we can use a pool of common messages
 * which is shared between more specific localizers. E.g., for a set of dialogs where submit and cancel
 * buttons are usually labeled with "OK" and "Cancel", we can create a shared base localizer L0 with keys
 * "ok" and "cancel". For a dialog which requests user to accept some terms, we can create a localizer L1
 * with root key "acceptTerms", key "acceptTerms.ok"="Accept" and L0 as a base localizer.
 *
 * When submit and cancel buttons in accept terms dialog are constructed, they will call localizer L1
 * and pass "ok" and "cancel" keys. L1 will find "acceptTerms.ok" in its own bundle and will pass "cancel"
 * to the base localizer.
 *
 * @author dbarashev@bardsoftware.com
 */
Dmitry Barashev +1
open class DefaultLocalizer(
    private val rootKey: String = "",
    private val baseLocalizer: Localizer = DummyLocalizer,
    private val prefixedLocalizer: Localizer? = null,
    private val currentTranslation: SimpleObjectProperty<ResourceBundle?> = SimpleObjectProperty( initialValue: null)) :
```

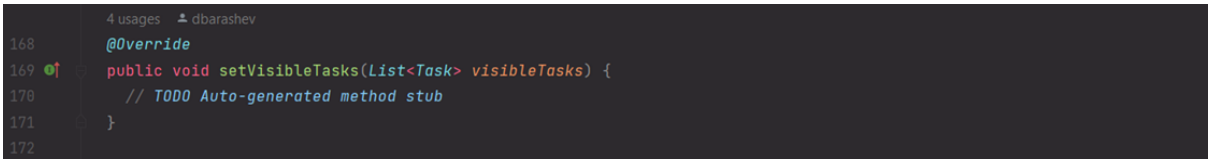
Code Smell 3 - Speculative Generality

Location - ganttproject/src/main/java/net/sourceforge/ganttproject/chart/ChartModelResource.java

Motive - An auto generated method without implementation inside it. Could lead to errors and, for now, it’s not doing anything.

Solution - Delete the method or fully implement it.

Code snippet -



```
4 usages  dbarashev
@Override
public void setVisibleTasks(List<Task> visibleTasks) {
  // TODO Auto-generated method stub
}
```

Pedro Gasparinho's Code Smells

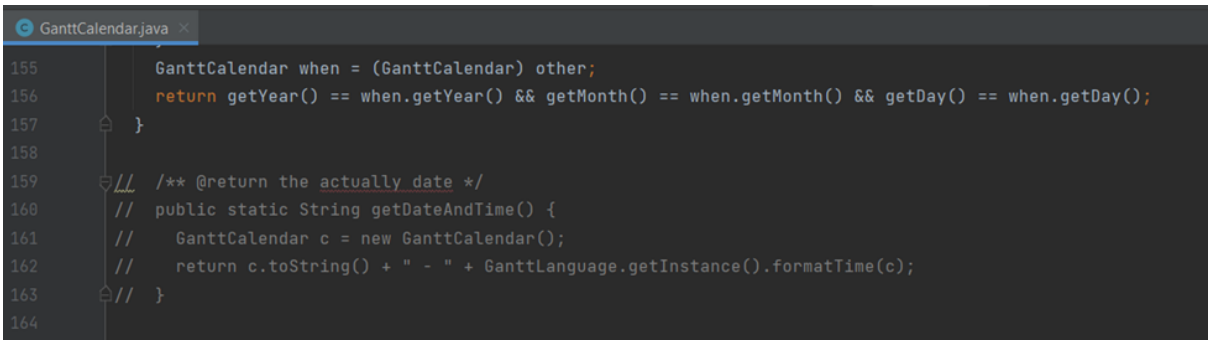
Code Smell 1 - Dead Code

Location - ganttproject\biz.ganttproject.core\src\main\java\biz.ganttproject\core\time\GanttCalender

Motive - When a method, such as this one, is commented it becomes obsolete since the functionality is either not supported anymore or implemented somewhere else, because of that the comments clutter the file where it's written.

Solution - Given the fact that this piece of code is commented and there is no indication of a reminder that it is still being worked on, I suggest that we delete those lines to improve file size and that class's readability.

Code snippet -



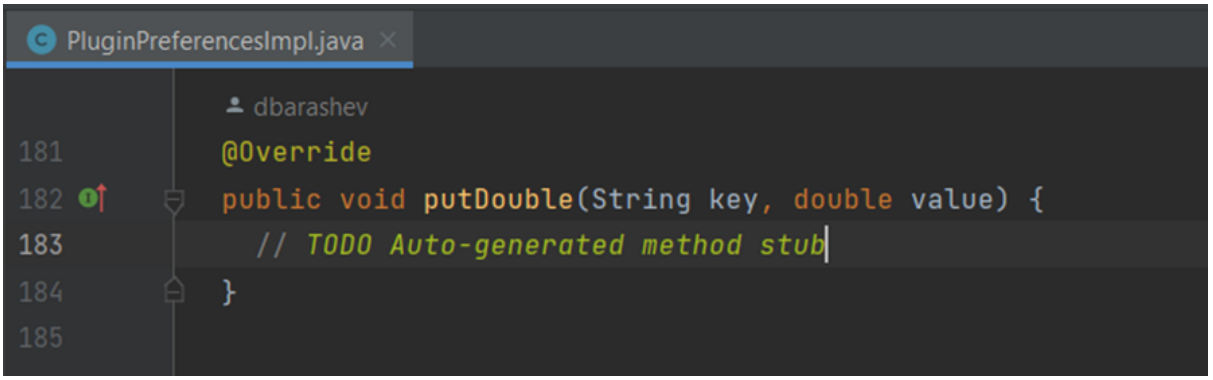
Code Smell 2 - Speculative Generality + Reminder Code

Location - ganttproject\ganttproject\src\main\java\net.sourceforge.ganttproject\PluginPreferencesImpl

Motive - The comment shown below on the codebase snippet indicates that the method is still being worked on, besides its auto generated, meaning that it must have been created by necessity of an Interface that the class implements.

Solution - In the previous version I suggested implementing this method, however it isn't used as of now, meaning it's an instance of a speculative generality code smell, so it's not a very good solution. As suggested by my colleague Afonso we should delete this method, however we first have to delete it from the interface since it's the only instance.

Code snippet -



Code Smell 3 - Long Method

Location - ganttproject\biz.ganttproject.core\src\main\java\biz.ganttproject\core\time\impl\GPTimeUnitStack

Motive - When looking at this method, it's easy to lose track of what it accomplishes and how so, not only because of the length by itself but also the accumulated complexity.

Solution - To refactor this piece of code what I suggest is to create a few auxiliary private methods to reduce the size of the original method and make it more digestible.

Code snippet -

GPTimeUnitStack.java

```
160     @Override
161     public TimeDuration parseDuration(String lengthAsString) throws ParseException {
162         int state = 0;
163         StringBuffer valueBuffer = new StringBuffer();
164         Integer currentValue = null;
165         TimeDuration currentLength = null;
166         lengthAsString += " ";
167         for (int i = 0; i < lengthAsString.length(); i++) {
168           char nextChar = lengthAsString.charAt(i);
169           if (Character.isDigit(nextChar)) {
170             switch (state) {
171             case 0:
172               if (currentValue != null) {
173                 throw new ParseException(lengthAsString, i);
174               }
175               state = 1;
176               valueBuffer.setLength(0);
177             case 1:
178               valueBuffer.append(nextChar);
179               break;
180             case 2:
181               TimeUnit timeUnit = findTimeUnit(valueBuffer.toString());
182               if (timeUnit == null) {
183                 throw new ParseException(lengthAsString, i);
184               }
185               assert currentValue != null;
186               TimeDuration localResult = createLength(timeUnit, currentValue.floatValue());
187               if (currentLength == null) {
188                 currentLength = localResult;
189               } else {
190                 if (currentLength.getTimeUnit().isConstructedFrom(timeUnit)) {
191                   float recalculatedLength = currentLength.getLength(timeUnit);
192                   currentLength = createLength(timeUnit, length: localResult.getValue() + recalculatedLength);
193                 } else {
194                   throw new ParseException(lengthAsString, i);
195                 }
196               }
197               state = 1;
198               currentValue = null;
199               valueBuffer.setLength(0);
200               valueBuffer.append(nextChar);
201               break;
202             }
203           } else if (Character.isWhitespace(nextChar)) {
204             switch (state) {
205             case 0:
206               break;
207             case 1:
208               currentValue = Integer.valueOf(valueBuffer.toString());
209               state = 0;
210               break;
211             case 2:
212               TimeUnit timeUnit = findTimeUnit(valueBuffer.toString());
213               if (timeUnit == null) {
214                 throw new ParseException(lengthAsString, i);
215               }
```

```
216             assert currentValue != null;
217             TimeDuration localResult = createLength(timeUnit, currentValue.floatValue());
218             if (currentLength == null) {
219               currentLength = localResult;
220             } else {
221               if (currentLength.getTimeUnit().isConstructedFrom(timeUnit)) {
222                 float recalculatedLength = currentLength.getLength(timeUnit);
223                 currentLength = createLength(timeUnit,  length: localResult.getValue() + recalculatedLength);
224               } else {
225                 throw new ParseException(lengthAsString, i);
226               }
227             }
228             state = 0;
229             currentValue = null;
230             break;
231           }
232         } else {
233           switch (state) {
234           case 1:
235             currentValue = Integer.valueOf(valueBuffer.toString());
236           case 0:
237             if (currentValue == null) {
238               throw new ParseException(lengthAsString, i);
239             }
240             state = 2;
241             valueBuffer.setLength(0);
```

```
242           case 2:
243             valueBuffer.append(nextChar);
244             break;
245           }
246         }
247       }
248       if (currentValue != null) {
249         currentValue = Integer.valueOf(valueBuffer.toString());
250         TimeUnit dayUnit = findTimeUnit( code: "d");
251         currentLength = createLength(dayUnit, currentValue.floatValue());
252       }
253       return currentLength;
254     }
255
```

Tiago Meirim' Code Smells

Code Smell 1 - Dead Code

Location - biz.ganttproject.core/src/main/java/biz/ganttproject/core/chart/grid/OffsetBuilderImpl.java

Motive - The method is commented so it is probably an idea for a code that was not finished or didn't have purpose to the code at the moment.

Solution - Since the code is not being used in any part of the project, it should be erased to reduce code size and to make the class simpler.

Code snippet -

```
//  public void setRightMarginBottomUnitCount(int value) {
//    myRightMarginBottomUnitCount = value;
//  }
```

Code Smell 2 - Long Method

Location - biz.ganttproject.core/src/main/java/biz/ganttproject/core/chart/grid/OffsetBuilder.java

Motive - The constructor is not used to simplify the class giving it methods for every attribute that is saved in the class, it gives the code this unnecessary 53 lines.

Solution - Using the constructor as it should be we can initialize every attribute with only 12 lines saving us a total of 41 lines of code.

Code snippet -

```
dbarashev
protected Factory() {
}
```

Code Smell 3 - Data Clumps

Location - biz.ganttproject.core/src/main/java/biz/ganttproject/core/chart/grid/OffsetBuilderImpl.java

Motive - There are a lot of parameters that appear numerous times in the methods.

Solution - Some of the parameters should be firstly stored in the class with the right conditions to be used and modified if needed like initialEnd.

Code snippet -

```
private void constructTopOffsets(TimeUnit timeUnit, List<Offset> topOffsets, List<Offset> bottomOffsets,
    int initialEnd, int baseUnitWidth) {
  int lastBottomOffset = bottomOffsets.get(bottomOffsets.size() - 1).getOffsetPixels();
  OffsetLookup offsetLookup = new OffsetLookup();
  Date currentDate = myStartDate;
  int prevEnd = initialEnd;
  int offsetEnd;
  do {
    TimeUnit concreteTimeUnit = getConcreteUnit(timeUnit, currentDate);
    Date endDate = concreteTimeUnit.adjustRight(currentDate);
    int bottomOffsetLowerBound = offsetLookup.lookupOffsetByEndDate(endDate, bottomOffsets);
    if (bottomOffsetLowerBound >= 0) {
      offsetEnd = bottomOffsets.get(bottomOffsetLowerBound).getOffsetPixels();
    } else {
      if (-bottomOffsetLowerBound > bottomOffsets.size()) {
        offsetEnd = lastBottomOffset + 1;
      } else {
        Offset ubOffset = bottomOffsetLowerBound <= -2 ? bottomOffsets.get(-bottomOffsetLowerBound - 2) : null;
        Date ubEndDate = ubOffset == null ? myStartDate : ubOffset.getOffsetEnd();
        int ubEndPixel = ubOffset == null ? 0 : ubOffset.getOffsetPixels();
        WorkingUnitCounter counter = new WorkingUnitCounter(GPCalendarCalc.PLAIN, baseUnit);
        offsetEnd = ubEndPixel + counter.run(ubEndDate, endDate).getLength() * baseUnitWidth;
      }
    }
    topOffsets.add(Offset.createFullyClosed(concreteTimeUnit, myStartDate, currentDate, endDate, prevEnd,  endPixels: initialEnd
        + offsetEnd, DayMask.WORKING));
    prevEnd = initialEnd + offsetEnd;
    currentDate = endDate;

  } while (offsetEnd <= lastBottomOffset && (myEndDate == null || currentDate.before(myEndDate)));
}
```

```
void constructBottomOffsets(List<Offset> offsets, int initialEnd) {
  int marginUnitCount = myRightMarginBottomUnitCount;
  Date currentDate = myStartDate;
  int shift = 0;
  OffsetStep step = new OffsetStep();
  int prevEnd = initialEnd;
  do {
    TimeUnit concreteTimeUnit = getConcreteUnit(getBottomUnit(), currentDate);
    calculateNextStep(step, concreteTimeUnit, currentDate);
    Date endDate = concreteTimeUnit.adjustRight(currentDate);
    if (endDate.compareTo(myViewportStartDate) <= 0) {
      shift = (int) (step.parrots * getDefaultUnitWidth());
    }
    int offsetEnd = (int) (step.parrots * getDefaultUnitWidth()) - shift;
    Offset offset = Offset.createFullyClosed(concreteTimeUnit, myStartDate, currentDate, endDate,
        prevEnd,  endPixels: initialEnd + offsetEnd, step.dayMask);
    prevEnd = initialEnd + offsetEnd;
    offsets.add(offset);
    currentDate = endDate;

    boolean hasNext = true;
    if (offsetEnd > getChartWidth()) {
      hasNext &= marginUnitCount-- > 0;
    }
    if (hasNext && myEndDate != null) {
      hasNext &= currentDate.before(myEndDate);
    }
    if (!hasNext) {
      return;
    }
  } while (true);
}
```

Afonso Nunes' Code Smells

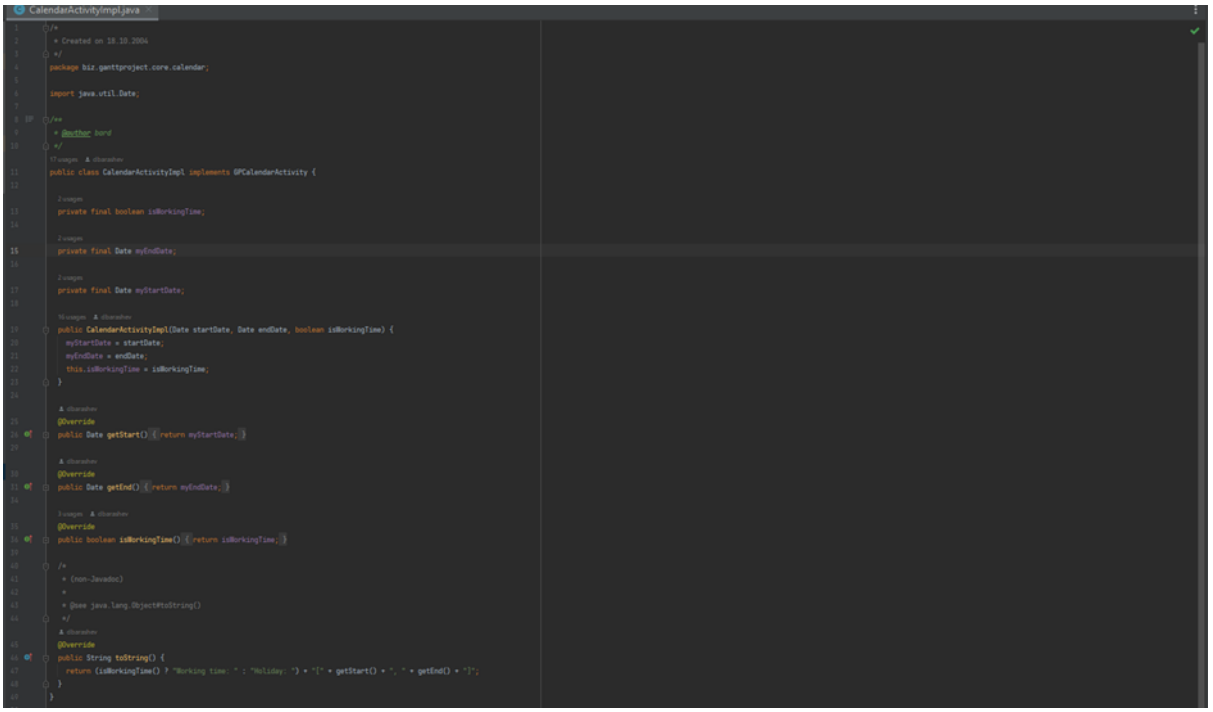
Code Smell 1 - No comments

Location - biz.ganttproject.core/src/main/java/biz/ganttproject/core/calendar/CalendarActivityImpl.java

Motive - When a class on a big project like in this case (or maybe even in a small one) has no comments, it becomes difficult to understand what it is supposed to do.

Solution - Add a simple class comment on what the class should actually be doing.

Code snippet -



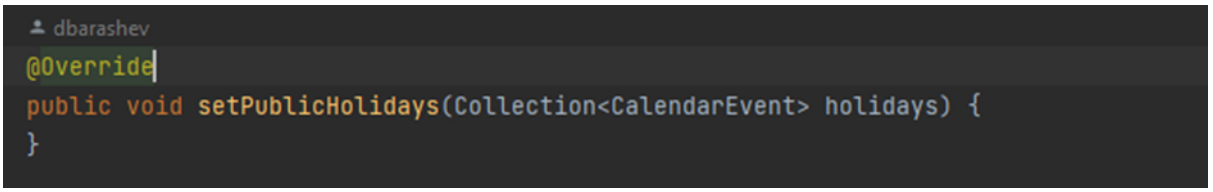
Code Smell 2 - Speculative Generality

Location - biz.ganttproject.core/src/main/java/biz/ganttproject/core/calendar/AlwaysWorkingTimeCalendarImpl.java

Motive - This piece of code indicates that the original writer implements an interface that has those methods declared, but he does not need to implement them in this specific class.

Solution - After a discussion with my colleague, he made me realize that I was making a foolish mistake and that I didn't pay enough attention. Since we are already extending one abstract class we can just implement those methods there and delete them from the subclass by extending that existing abstract class.

Code snippet -



Code Smell 3 - Data Class

Location - biz.ganttproject.core/src/main/java/biz/ganttproject/core/calendar/CalendarActivityImpl.java

Motive - This is a very small class consisting of only data with no real functionality besides getter methods and a simple toString() method.

Solution - Defining these methods in another class where all this data is easily accessible from and where it makes more sense.

Code snippet -

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

import java.util.Date;

/**
 * @author bard
 */
17 usages  dbarashev
public class CalendarActivityImpl implements GPCalendarActivity {

  2 usages
  private final boolean isWorkingTime;

  2 usages
  private final Date myEndDate;

  2 usages
  private final Date myStartDate;

  16 usages  dbarashev
  public CalendarActivityImpl(Date startDate, Date endDate, boolean isWorkingTime) {
    myStartDate = startDate;
    myEndDate = endDate;
    this.isWorkingTime = isWorkingTime;
  }

  dbarashev
  @Override
  public Date getStart() { return myStartDate; }

  dbarashev
  @Override
  public Date getEnd() { return myEndDate; }

  3 usages  dbarashev
  @Override
  public boolean isWorkingTime() { return isWorkingTime; }

  /*
   * (non-Javadoc)
   *
   * @see java.lang.Object#toString()
   */
  dbarashev
  @Override
  public String toString() {
    return (isWorkingTime() ? "Working time: " : "Holiday: ") + "[" + getStart() + ", " + getEnd() + "]";
  }
}
```