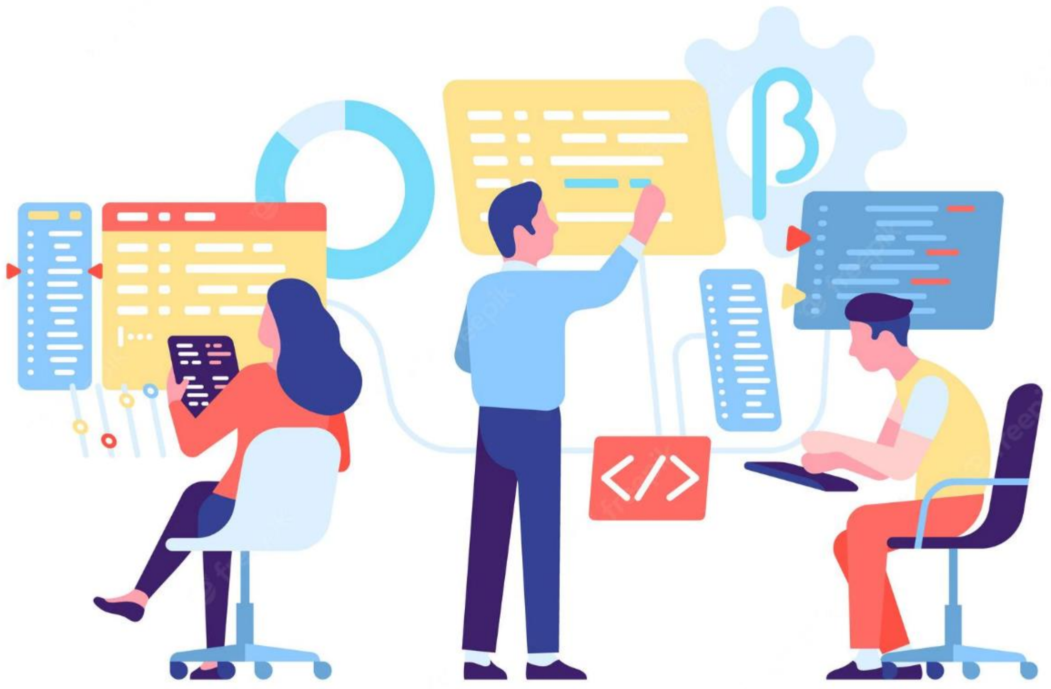
NOVA SCHOOL OF SCIENCE & TECHNOLOGY

THE S TEAM

This document presents the report of the 2nd phase of the project for the Software Engineering course of the 1st semester of 2022/2023.



Carolina Simonet, 59748, Lab 3

Filipe Santo, 64859, Lab 4

Jaime Russo, 60062, Lab 6

Margarida Carvalho, 60437, Lab 4

Rui Capareira, 57046, Lab 4

Team's Git project link:
https://github.com/CarolinaSimonet/ganttproject.git

Team's Demo Video:
https://youtu.be/AfAmVI91RDQ

Content

New Functionalities Description

The two functionalities that we decide to implement are based on things we thought are useful and missing in the program that was given to us.

The first one is the possibility to add a constraint to a task and the associated date. The goal is to choose between a list of already defined constraint types and choose one of them for the task. The defined types are: *must start on*, *must finish on*, *start no earlier than*, *start no later than*, *finish no earlier than, finish no later than*.

The second functionality is the ability to press a button that displays the daily information. This information is about the planned tasks for the current day, so the app user doesn't have to search in the whole Gantt panel to know this information.

User Stories

- As a project manager I want to be able to press a button with the daily information so I can easily check what tasks need to be done that day.
- As a project manager I want to add a constraint type and a constraint date to a task so I can prioritize better my tasks.

Use Cases and Diagrams

Use Case a)

Name: ShowsDailyInformation

Id: 1

Description: The system displays information about the tasks of the current day

Actors:

Main: Project manager

Secondary: None

Pre-conditions: The system has tasks

Main flow:

1. The use case begins when the project manager selects "Show daily information".

2. The system displays a text box with the active subtasks for the current day.

2.1. The information displayed by the summary is the total duration, the start day and the finish day of each subtask.

Alternative flows: ADayWithNoTasks

Post-conditions: A text box with the daily information was showed.

Use Case b)

Name: CloseDailyInformation

Id: 2

Description: The system closes the text box with the daily information

Actors:

Main: Project manager

Secondary: None

Pre-conditions: The system is showing the text box with the daily information

Main flow:

1. The use case begins when the project manager selects "Show daily information"

2. The system displays hides/ closes a text box with the active subtasks for the current day

Alternative flows: None.

Post-conditions: A text box with the daily information was hided.

Use Case c)

Name: ADayWithNoTasks

Id: 3

Description: Daily button dialog without tasks

Actors:

Main: Project Manager

Secondary: None

Pre-conditions: Zero tasks to be done that day

Main flow:

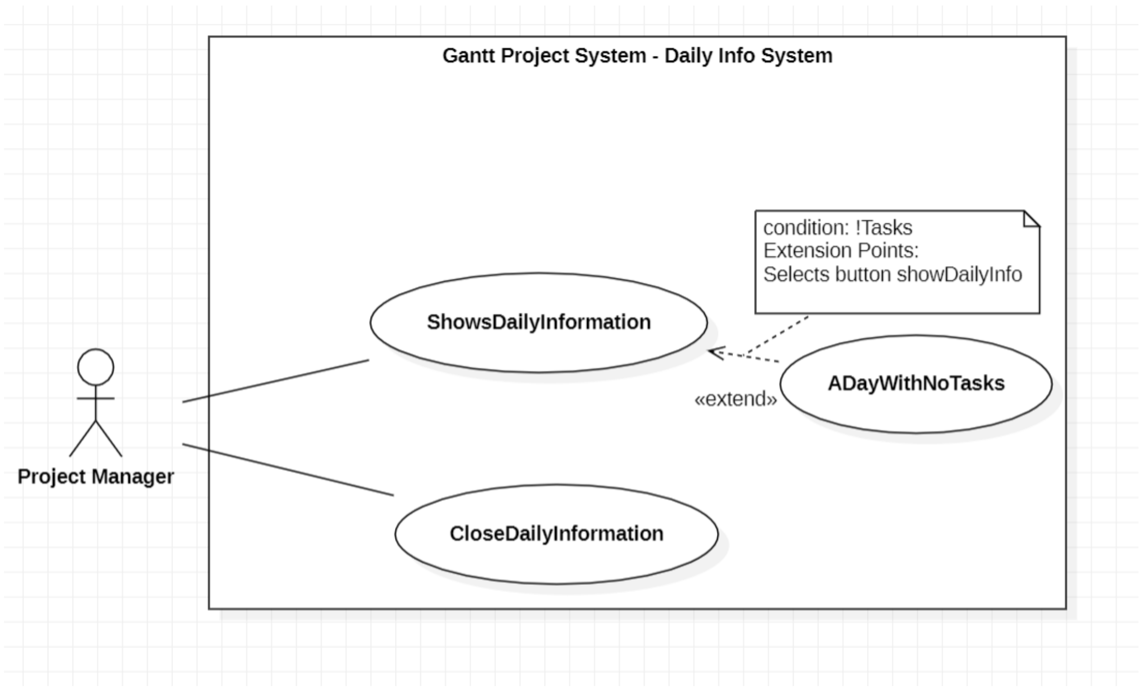
1. The use case starts when the project manager selects the “Show daily information button”.

2. The system displays a text box with a message that says “No tasks for today”.

Alternative flows: None

Post-conditions: A text box with a message is displayed.

a), b), c) Diagram



Done by: Carolina Simonet

Reviewed by: Margarida Carvalho

Use Case d)

Name: DisplayVerticalLine

Id: 4

Description: The system displays a vertical line of the current day

Actors:

Main: Project Manager

Secondary: None

Pre-conditions: None

Main flow:

1. The use case starts when the project manager selects the “Edit” button.

2. The project manager selects the daily “Settings” button.

3. The project manager selects the “Gantt Chart” button.

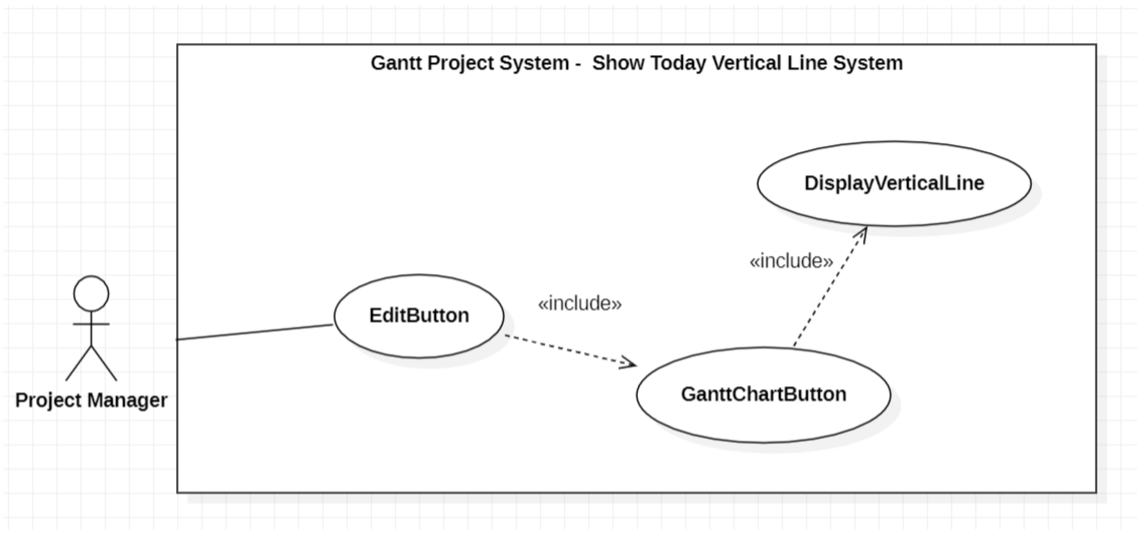
4. The project manager selects the “Yes” option in “Show today as a red line”.

5. The system displays a vertical line at the current day.

Alternative flows: None

Post-conditions: A vertical line displayed.

d) Diagram



Done by: Filipe Santo

Reviewed by: Rui Capareira

Use Case e)

Name: AddToMainTask

Id: 5

Description: Adds an existing task to an existing main task as a subtask

Actors:

Main: Project manager

Secondary: None

Pre-conditions: The task and the main task exist

Main flow:

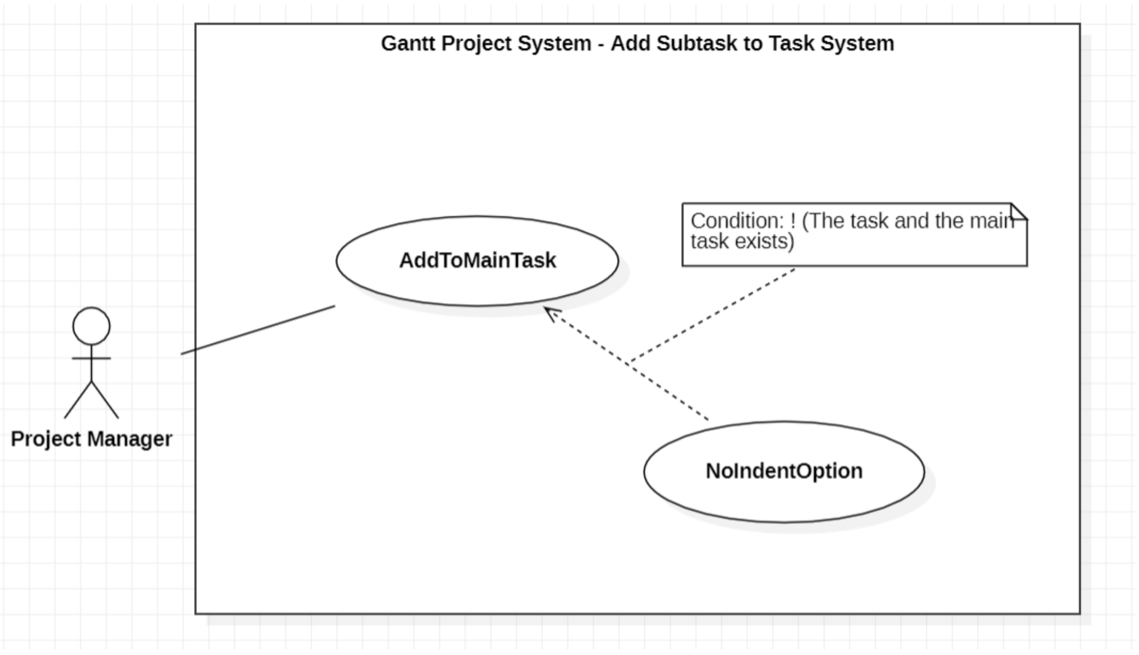
1. The use case begins when the project manager selects a task that is strictly under the main task.

2. The project manager selects the “Indent”/”Avançar” button

Alternative flows: None

Post-conditions: A new subtask was added to a main task

e) Diagram


Gantt Project System - Add Subtask to Task System
AddToMainTask
Condition: ! (The task and the main task exists)
Project Manager
NoIndentOption

Done by: Jaime Russo

Reviewed by: Filipe Santo

Use Case f)

Name: ShowCriticalPath

Id: 6

Description: Checks the project’s critical path at any stage of the project

Actors:

Main: Project Manager

Secondary: None

Pre-conditions:

1. The project already has associated tasks.

Main flow:

1. The use case starts when the project manager selects the “Show critical path” button.

2. The system shows the project’s critical path by adding lines to the critical tasks.

Alternative flows: None

Post-conditions: The project’s critical path is now displayed.

Use Case g)

Name: HideCriticalPath

Id: 7

Description: Checks the project’s critical path at any stage of the project

Actors:

Main: Project Manager

Secondary: None

Pre-conditions:

1. The critical path is displayed.

Main flow:

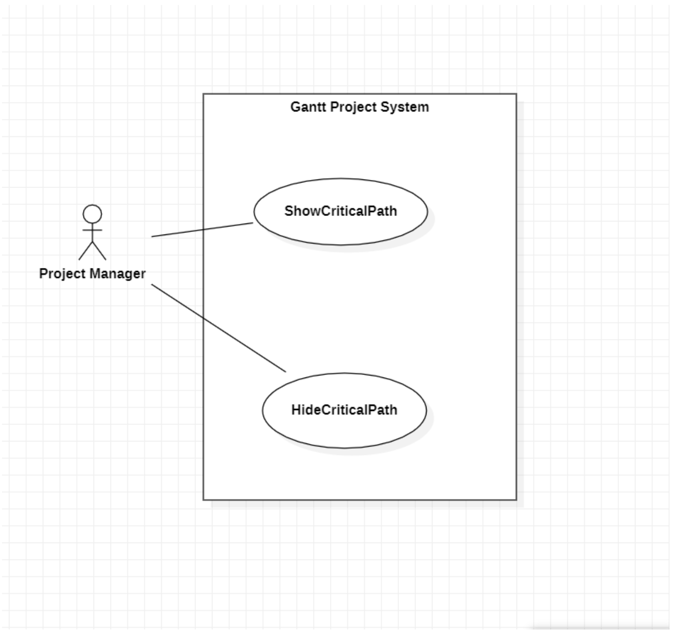
2. The use case starts when the project manager selects the “Hide critical path” button.

3. The system hides the project’s critical path, by take back the lines of the critical tasks.

Alternative flows: None

Post-conditions: The project’s critical path is now hidden.

f), g) Diagram



Done by: Margarida Carvalho

Reviewed by: Jaime Russo

Use Case h)

Name: ShowTaskProperties

Id: 8

Description: The system displays the properties about the selected task

Actors:

Main: Project Manager

Secondary: None

Pre-conditions:

1. A task has been selected

2. The project has at least one task

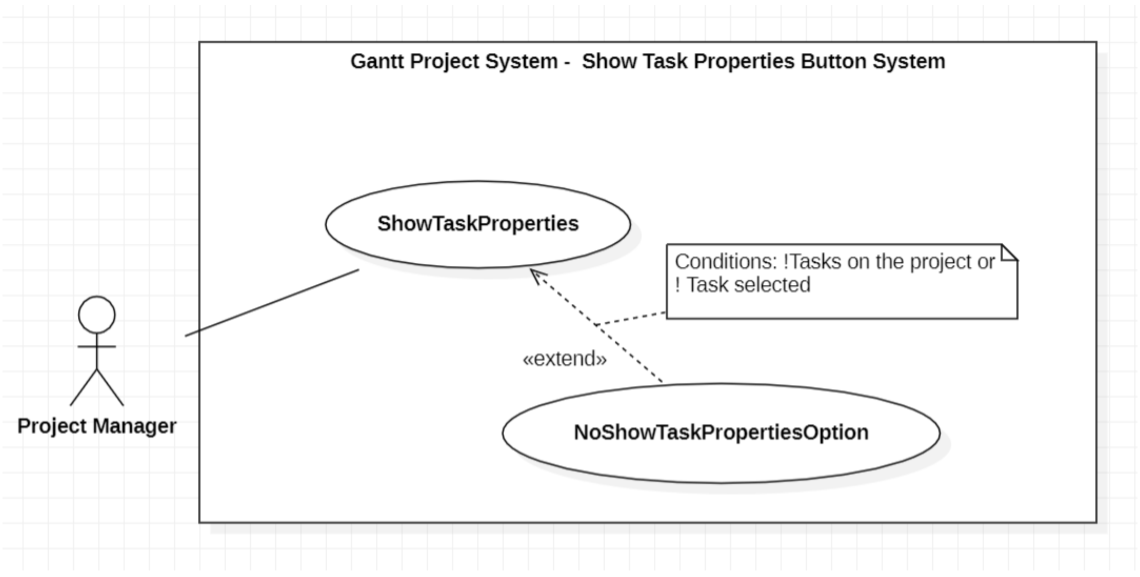
Main flow:

1. The use case starts when the project manager selects a task and then selects the task properties button.

2. A dialog with the correspondent task properties popup. Alternative flows:

Post-conditions: The properties dialog box is displayed.

h) Diagram

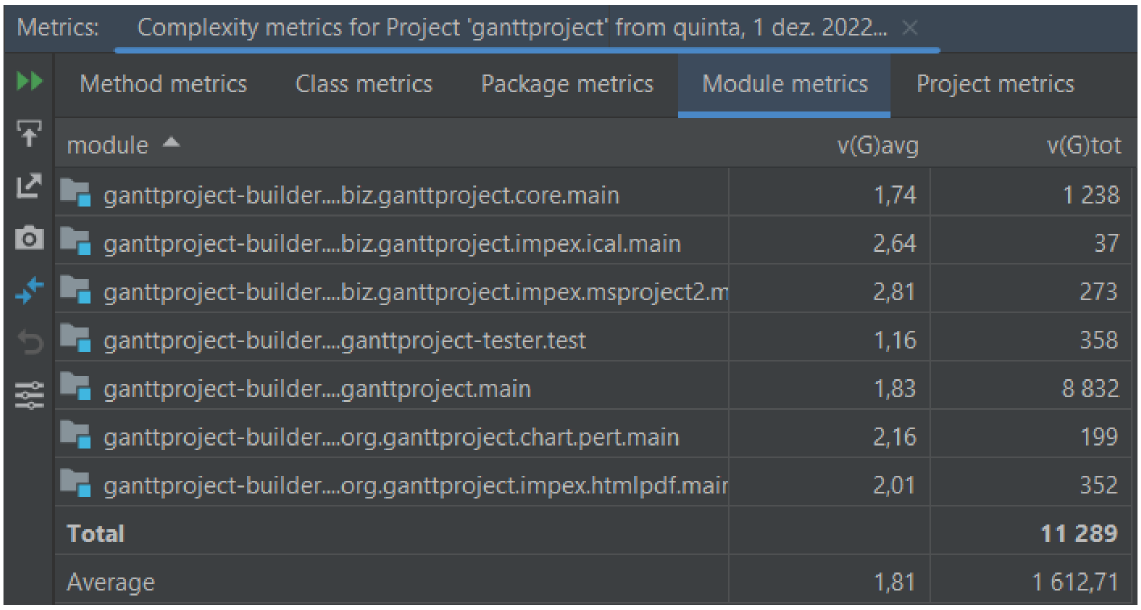

Gantt Project System - Show Task Properties Button System
ShowTaskProperties
Conditions: !Tasks on the project or ! Task selected
«extend»
NoShowTaskPropertiesOption
Project Manager

Done by: Rui Capareira

Reviewed by: Carolina Simonet

Codebase Metrics

Complexity Metrics



Metrics: Complexity metrics for Project 'ganttproject' from quinta, 1 dez. 2022...

Method metrics | Class metrics | Package metrics | Module metrics | Project metrics

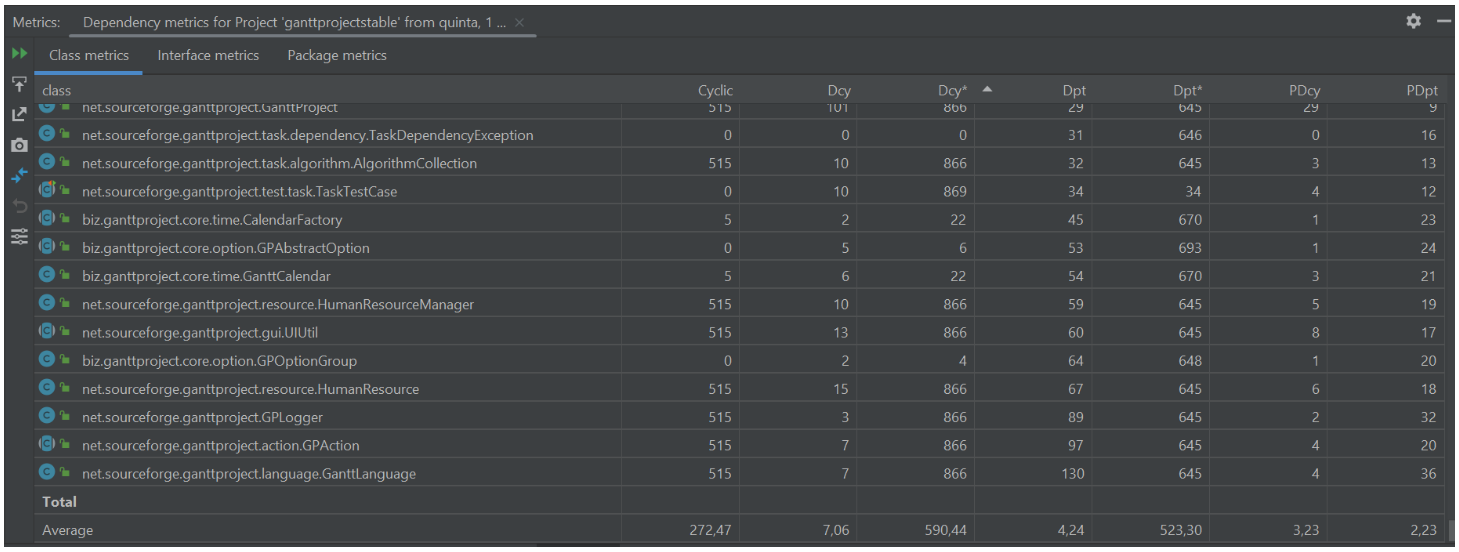
module	v(G)avg	v(G)tot
ganttproject-builder....biz.ganttproject.core.main	1,74	1 238
ganttproject-builder....biz.ganttproject.impex.ical.main	2,64	37
ganttproject-builder....biz.ganttproject.impex.msproject2.m	2,81	273
ganttproject-builder....ganttproject-tester.test	1,16	358
ganttproject-builder....ganttproject.main	1,83	8 832
ganttproject-builder....org.ganttproject.chart.pert.main	2,16	199
ganttproject-builder....org.ganttproject.impex.htmlpdf.mair	2,01	352
Total		**11 289**
Average	1,81	1 612,71

The Complexity Metrics predict critical information about the trust and maintenance of software systems. We can measure them by determining the project's number of components and the interactions with the system. The complexity measure is cyclomatic, which means that a module's complexity is the independent cycles' number in the graphical representation of a control flow, and it tries to catch the difficulty's level of understanding the project's modules.
In the module complexity metrics of this project, we can see a high total value of cyclomatic complexity, which means it has a high level of complexity.

Done by: Margarida Carvalho

Reviewed by: Filipe Santo

Dependency Metrics



Metrics: Dependency metrics for Project 'ganttprojectstable' from quinta, 1 ...

Class metrics | Interface metrics | Package metrics

class	Cyclic	Dcy	Dcy*	Dpt	Dpt*	PDcy	PDpt
net.sourceforge.ganttproject.GanttProject	515	101	866	29	645	29	9
net.sourceforge.ganttproject.task.dependency.TaskDependencyException	0	0	0	31	646	0	16
net.sourceforge.ganttproject.task.algorithm.AlgorithmCollection	515	10	866	32	645	3	13
net.sourceforge.ganttproject.test.task.TaskTestCase	0	10	869	34	34	4	12
biz.ganttproject.core.time.CalendarFactory	5	2	22	45	670	1	23
biz.ganttproject.core.option.GPAbstractOption	0	5	6	53	693	1	24
biz.ganttproject.core.time.GanttCalendar	5	6	22	54	670	3	21
net.sourceforge.ganttproject.resource.HumanResourceManager	515	10	866	59	645	5	19
net.sourceforge.ganttproject.gui.UIUtil	515	13	866	60	645	8	17
biz.ganttproject.core.option.GPOptionGroup	0	2	4	64	648	1	20
net.sourceforge.ganttproject.resource.HumanResource	515	15	866	67	645	6	18
net.sourceforge.ganttproject.GPLogger	515	3	866	89	645	2	32
net.sourceforge.ganttproject.action.GPAction	515	7	866	97	645	4	20
net.sourceforge.ganttproject.language.GanttLanguage	515	7	866	130	645	4	36
Total							
Average	272,47	7,06	590,44	4,24	523,30	3,23	2,23

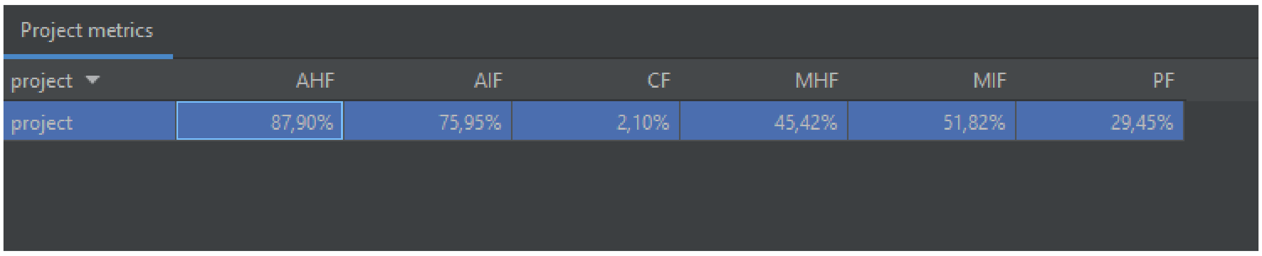
Dependency metrics are a measure of how many components a given class, module or method depends on to operate. This indicates that a component may be difficult to observe, reuse, test and maintain. For instance, it's unlikely that development teams will be able to update a class with many dependencies without also updating each class it depends on.

As we can see in the caption, we have a big number of average dependencies. This results in hard work for the teams who want to update this Gantt Project and is also a code smell Shoot Gun Surgery, because even if you just want to make a small change you have to make changes in many places.

Done by: Carolina Simonet

Reviewed by: Jaime Russo

MOOD metrics



Project metrics

project	AHF	AIF	CF	MHF	MIF	PF
project	87,90%	75,95%	2,10%	45,42%	51,82%	29,45%

Label:

-AHF: Attribute Hiding Factor,

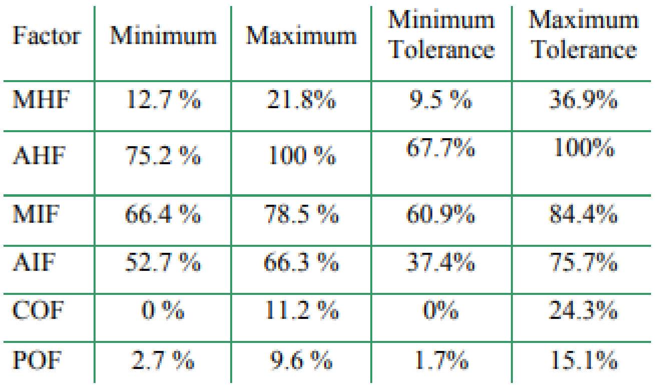
-AIF: Attribute Inheritance Factor,

-CF or COF: Coupling Factor,

-MHF: Method Hiding Factor,

-MIF: Method Inheritance Factor,

-PF or POF: Polymorphism Factor;



Factor	Minimum	Maximum	Minimum Tolerance	Maximum Tolerance
MHF	12.7 %	21.8%	9.5 %	36.9%
AHF	75.2 %	100 %	67.7%	100%
MIF	66.4 %	78.5 %	60.9%	84.4%
AIF	52.7 %	66.3 %	37.4%	75.7%
COF	0 %	11.2 %	0%	24.3%
POF	2.7 %	9.6 %	1.7%	15.1%

- **AHF, MHF:**

-AHF: as we can see, the majority of the attributes are not visible for the rest of the classes, the ideal AHF would be 100%. In our case we didn't get a bad value.

-MHF: this value means that half of our methods are visible to the rest of the classes and half are not. We were looking for a 8% - 25%.

- **AIF, MIF:**

-AIF: in our project a lot of attributes are inherited to classes from classes. The ideal range of this factor would be from 0% up to 48%.

-MIF: our classes inherit some methods of the parents super classes. We got 51,82% that is in the range of a good MIF (20% - 80%)

- **PF:**

-PF: this factor is associated with method overriding and not associated with method overloading. In our case, having a relatively high polymorphism factor, it means that we have a better code quality, but in the other side, we have a more complex system.

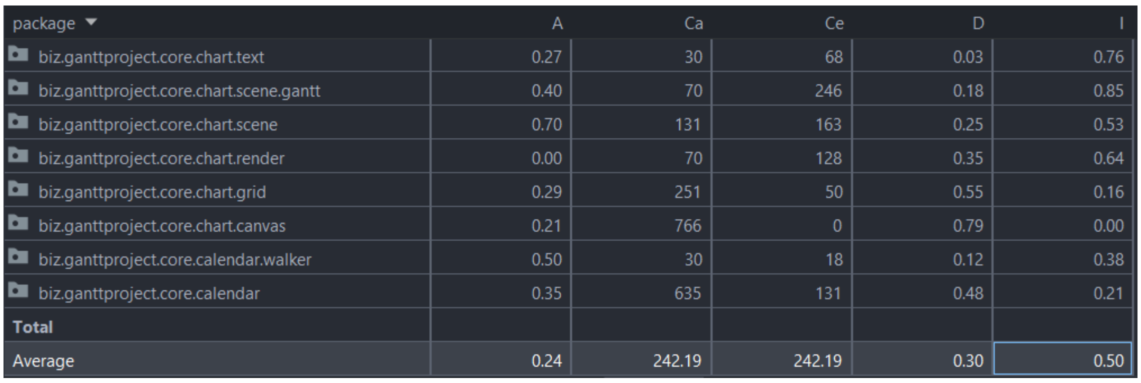
- **CF:**

-CF: Many functionalities of the system can be done with the help of coupled classes. Having many independent classes might be a sign of bad practice coding. In the other side, having many classes dependent from others, can lead to some code smells, like shotgun surgery, if we want to change something inside a class, we also need to change other classes. In our project, we have a coupling factor of 2,10%, when the ideal range is from 0% to 11,2%.

Done by: Jaime Russo

Reviewed by: Rui Capareira

Martin Packaging Metrics



package	A	Ca	Ce	D	I
biz.ganttproject.core.chart.text	0.27	30	68	0.03	0.76
biz.ganttproject.core.chart.scene.gantt	0.40	70	246	0.18	0.85
biz.ganttproject.core.chart.scene	0.70	131	163	0.25	0.53
biz.ganttproject.core.chart.render	0.00	70	128	0.35	0.64
biz.ganttproject.core.chart.grid	0.29	251	50	0.55	0.16
biz.ganttproject.core.chart.canvas	0.21	766	0	0.79	0.00
biz.ganttproject.core.calendar.walker	0.50	30	18	0.12	0.38
biz.ganttproject.core.calendar	0.35	635	131	0.48	0.21
Total					
Average	0.24	242.19	242.19	0.30	0.50

This is a software package metric that focuses on identifying packages which are hard to maintain and reuse. The metric measures how hard it is to change the package, measured by workload.

As shown in the image above, the GanttProject has a high average of Afferent and Efferent couplings, which means there’s a large number of classes outside the package which depend on it and a large number of classes outside the package which the package depends upon.

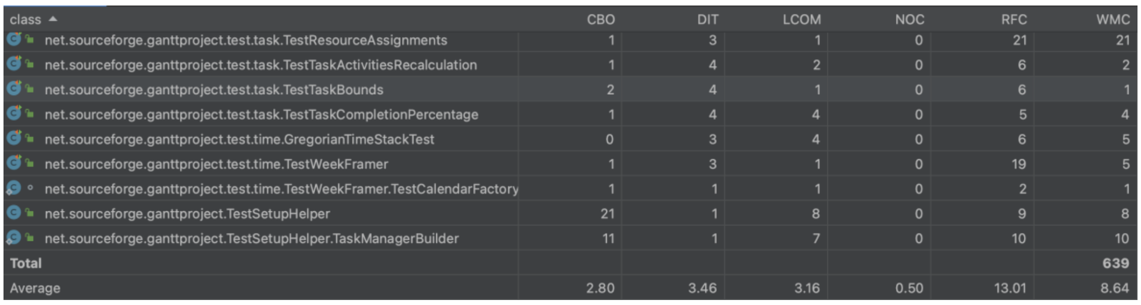
Stable packages are hard to change but they also should be abstract so they can be extended. On the other hand an unstable code should be concrete so the code can be easily edited.

The normalized distance from main sequence (D) defines a relationship between the levels of Abstractness and Instability shown in the GanttProject metrics, we could say that a higher value of Instability could reduce the workload required to make changes to the project.

Done by: Rui Capareira

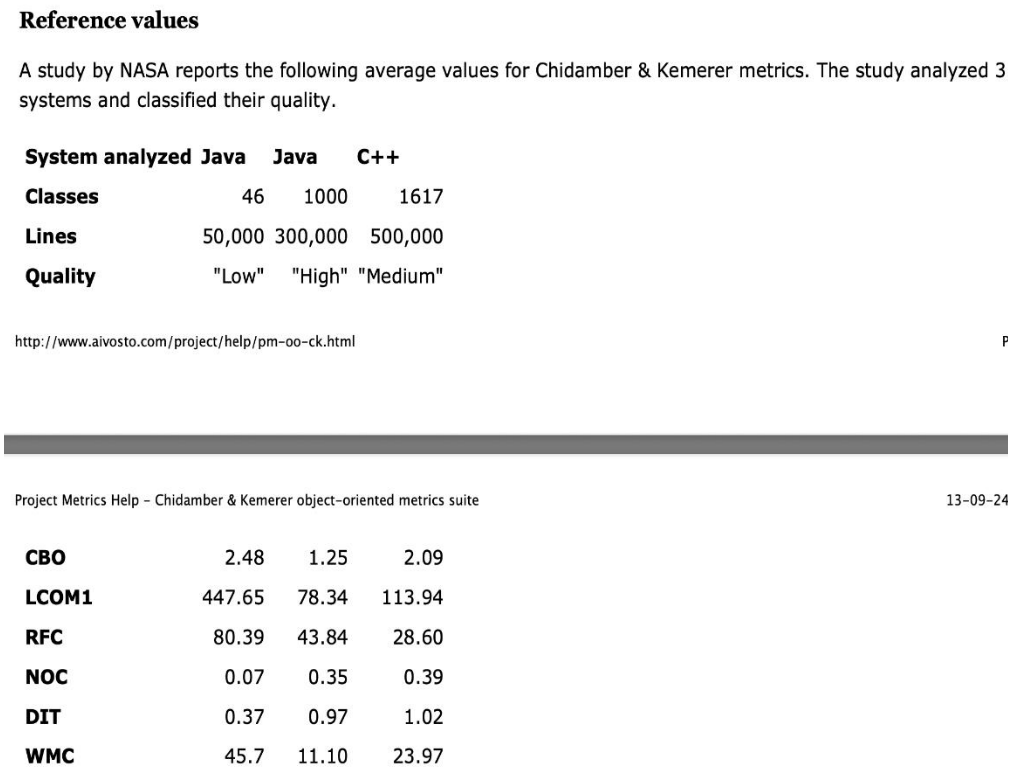
Reviewed by: Margarida Carvalho

Chidamber and Kemerer Metrics



class	CBO	DIT	LCOM	NOC	RFC	WMC
net.sourceforge.ganttproject.test.task.TestResourceAssignments	1	3	1	0	21	21
net.sourceforge.ganttproject.test.task.TestTaskActivitiesRecalculation	1	4	2	0	6	2
net.sourceforge.ganttproject.test.task.TestTaskBounds	2	4	1	0	6	1
net.sourceforge.ganttproject.test.task.TestTaskCompletionPercentage	1	4	4	0	5	4
net.sourceforge.ganttproject.test.time.GregorianTimeStackTest	0	3	4	0	6	5
net.sourceforge.ganttproject.test.time.TestWeekFramer	1	3	1	0	19	5
net.sourceforge.ganttproject.test.time.TestWeekFramer.TestCalendarFactory	1	1	1	0	2	1
net.sourceforge.ganttproject.TestSetupHelper	21	1	8	0	9	8
net.sourceforge.ganttproject.TestSetupHelper.TaskManagerBuilder	11	1	7	0	10	10
Total						639
Average	2.80	3.46	3.16	0.50	13.01	8.64

Chidamber and Kemerer metrics consists of six metrics calculated for each class. CBO (number of classes to which a class is coupled), two classes are coupled when methods declared in one class use methods defined by the other, an high CBO is undesirable. DIT (maximum inheritance path from the class to the root class), indicates the deeper a class is in the hierarchy, a high DIT is related to increase in faults, a recommended DIT is 5 or less. LCOM (lack of cohesion of methods), is the lack of cohesion of methods. NOC (number of immediate sub-classes of a class), is the number of immediate child class derived from a base class, a high NOC is related to fewer faults. RFC (response for a class), is a set of methods that can potentially be executed in response to a message received. WMC (number of methods defined in class), it's a predictor of how much time and effort is required to maintain the class, a high WMC is related to lead to more faults, and usually indicates that the class could be divided into more.



Reference values

A study by NASA reports the following average values for Chidamber & Kemerer metrics. The study analyzed 3 systems and classified their quality.

System analyzed	Java	Java	C++
Classes	46	1000	1617
Lines	50,000	300,000	500,000
Quality	"Low"	"High"	"Medium"

http://www.aivosto.com/project/help/pm-oo-ck.html P

Project Metrics Help - Chidamber & Kemerer object-oriented metrics suite 13-09-24

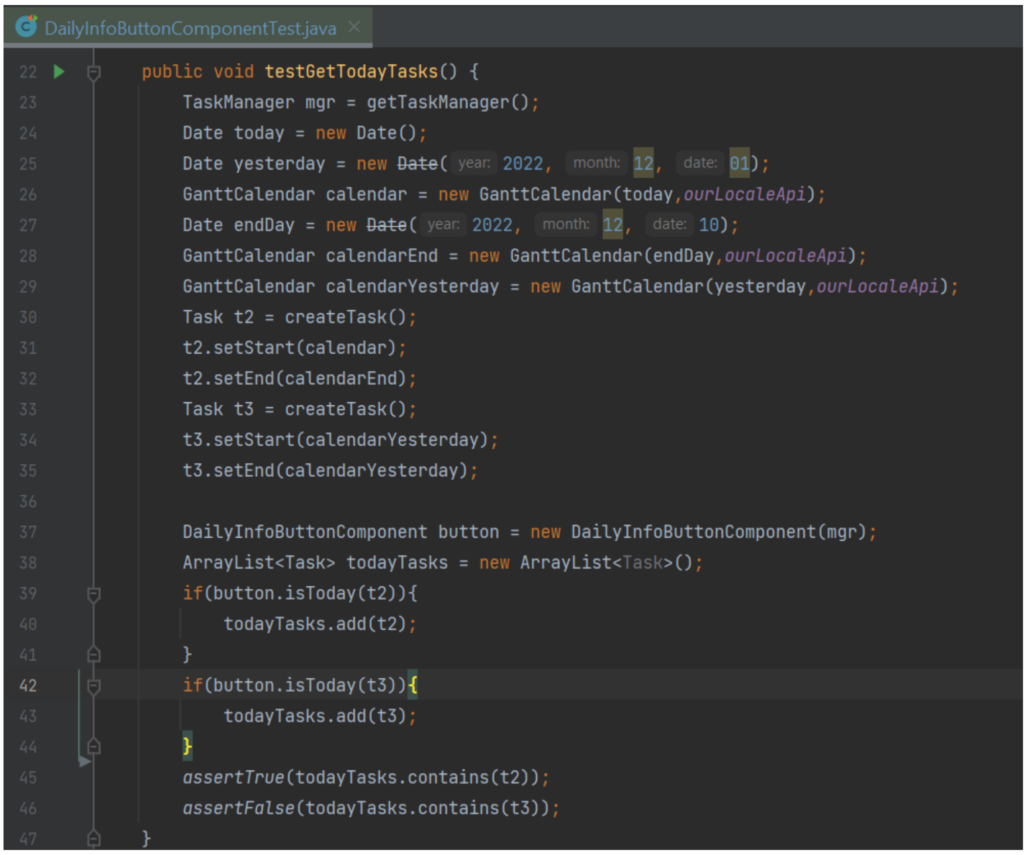
CBO	2.48	1.25	2.09
LCOM1	447.65	78.34	113.94
RFC	80.39	43.84	28.60
NOC	0.07	0.35	0.39
DIT	0.37	0.97	1.02
WMC	45.7	11.10	23.97

With this is mind, we can evaluate that the code has a high CBO (bad), recommended DIT (good), low LCOM (good), high NOC (good), and a low WMC (good). Suggesting that by the Chidamber and Kemerer Metrics the quality of the code is medium to good in quality of the system.

Done by: Filipe Santo

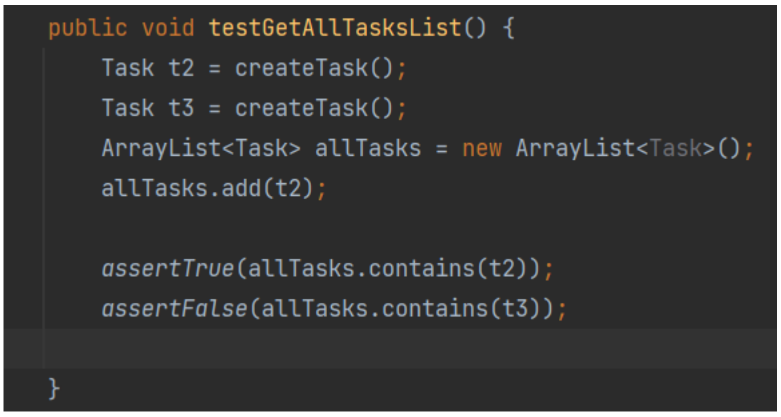
Reviewed by: Carolina Simonet

Unit Tests

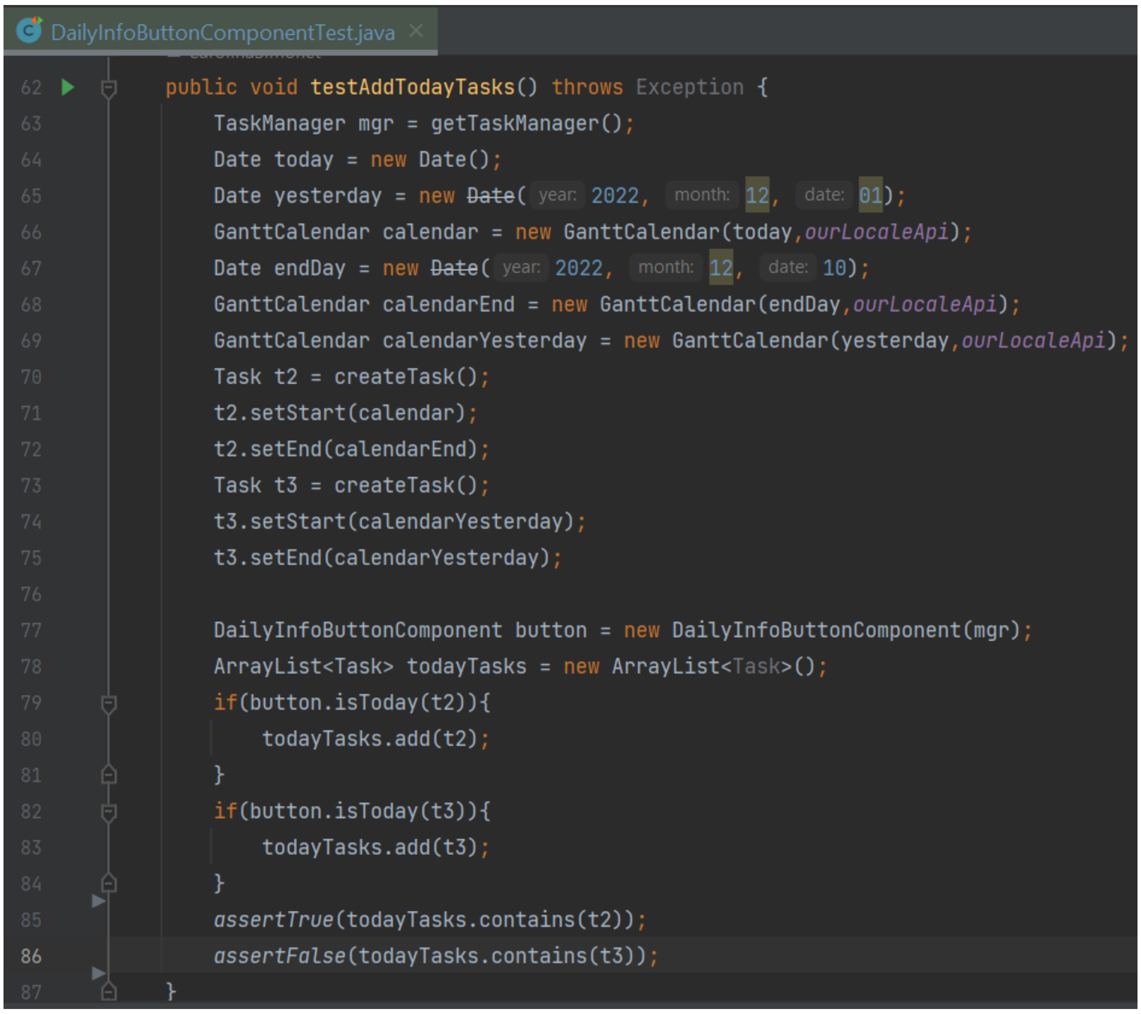

DailyInfoButtonComponentTest.java

```
public void testGetTodayTasks() {
    TaskManager mgr = getTaskManager();
    Date today = new Date();
    Date yesterday = new Date( year: 2022,  month: 12,  date: 01);
    GanttCalendar calendar = new GanttCalendar(today,ourLocaleApi);
    Date endDay = new Date( year: 2022,  month: 12,  date: 10);
    GanttCalendar calendarEnd = new GanttCalendar(endDay,ourLocaleApi);
    GanttCalendar calendarYesterday = new GanttCalendar(yesterday,ourLocaleApi);
    Task t2 = createTask();
    t2.setStart(calendar);
    t2.setEnd(calendarEnd);
    Task t3 = createTask();
    t3.setStart(calendarYesterday);
    t3.setEnd(calendarYesterday);

    DailyInfoButtonComponent button = new DailyInfoButtonComponent(mgr);
    ArrayList<Task> todayTasks = new ArrayList<Task>();
    if(button.isToday(t2)){
        todayTasks.add(t2);
    }
    if(button.isToday(t3)){
        todayTasks.add(t3);
    }
    assertTrue(todayTasks.contains(t2));
    assertFalse(todayTasks.contains(t3));
}
```

In this test we create two tasks, one of them occurs in the current day and the other not. The test will check if the task that occurs in the current day was added to the array of the current tasks.


```
public void testGetAllTasksList() {
    Task t2 = createTask();
    Task t3 = createTask();
    ArrayList<Task> allTasks = new ArrayList<Task>();
    allTasks.add(t2);

    assertTrue(allTasks.contains(t2));
    assertFalse(allTasks.contains(t3));

}
```

In this case we are just checking if the created tasks were added to the array will every task.


DailyInfoButtonComponentTest.java

```java
    public void testAddTodayTasks() throws Exception {
        TaskManager mgr = getTaskManager();
        Date today = new Date();
        Date yesterday = new ~~Date~~( year: 2022, month: 12, date: 01);
        GanttCalendar calendar = new GanttCalendar(today,ourLocaleApi);
        Date endDay = new ~~Date~~( year: 2022, month: 12, date: 10);
        GanttCalendar calendarEnd = new GanttCalendar(endDay,ourLocaleApi);
        GanttCalendar calendarYesterday = new GanttCalendar(yesterday,ourLocaleApi);
        Task t2 = createTask();
        t2.setStart(calendar);
        t2.setEnd(calendarEnd);
        Task t3 = createTask();
        t3.setStart(calendarYesterday);
        t3.setEnd(calendarYesterday);

        DailyInfoButtonComponent button = new DailyInfoButtonComponent(mgr);
        ArrayList<Task> todayTasks = new ArrayList<Task>();
        if(button.isToday(t2)){
            todayTasks.add(t2);
        }
        if(button.isToday(t3)){
            todayTasks.add(t3);
        }
        assertTrue(todayTasks.contains(t2));
        assertFalse(todayTasks.contains(t3));
    }
```

In this test we are checking if the tasks are well added to the today tasks array following the condition imposed by isToday.

Conclusion

To summarize, with this project we were able to really put in practice our learnings about Agile Methods. We found out that it was crucial to have an organized team, in order to produce a successful project.

Since we were a team of only five members, it was important that each on of us had a cross-functional role.

In the beginning, we had some difficulties to get used to this planning method and to discover what was behind the project that was given to us. After the teachers released the stable version tutorial this got easier to us.

One of the team members had some issues running the Gantt Project App because there wasn't available any Gradle version for his software, so this put our team in a challenging situation. But as a team we overcame it by working more together in online meetings.

Before the releasing of the stable version, we already had a repository with some commits that can be checked at:

https://github.com/MargaridaCarv/ganttproject.git

We all agreed that this method helped us and we worked better together and at a good rhythm with it.

The S Team

1st Part

Carolina Simonet, 59748

Filipe Santo, 64859

Jaime Russo, 60062

Margarida Carvalho, 60437

Rui Capareira, 57046

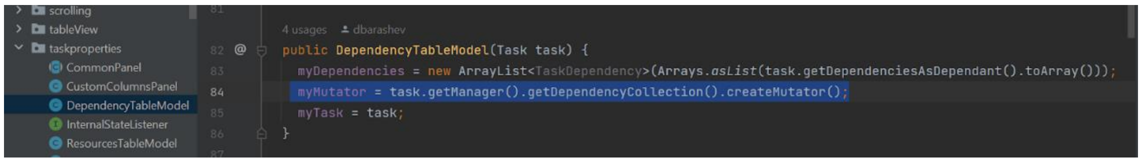
Code Smells

Carolina Simonet, 59748

1. Message Chains
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/gui/taskproperties/DependencyTableModel.java

   Explanation:
   We are getting an object, and again getting another object back and again calling another method.

   Reviewed by Margarida Carvalho.

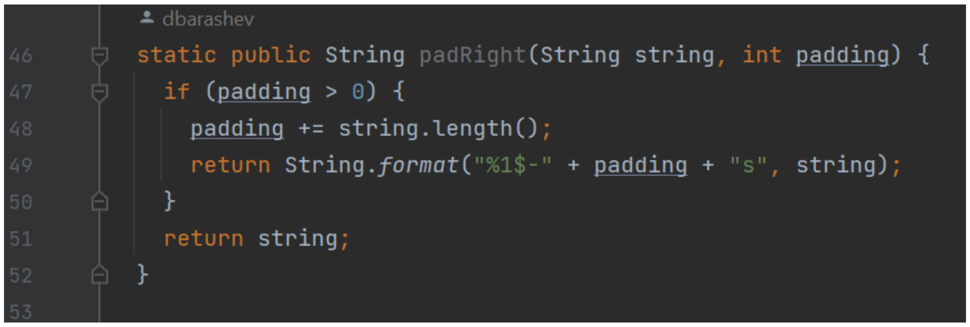


```java
public DependencyTableModel(Task task) {
  myDependencies = new ArrayList<TaskDependency>(Arrays.asList(task.getDependenciesAsDependant().toArray()));
  myMutator = task.getManager().getDependencyCollection().createMutator();
  myTask = task;
}
```

2. Dead Code
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/util/StringUtils.java

   Explanation:
   The code is not used for now, and we already have a similar method padLeft.So we could delete this method.

   Reviewed by Filipe Santo.

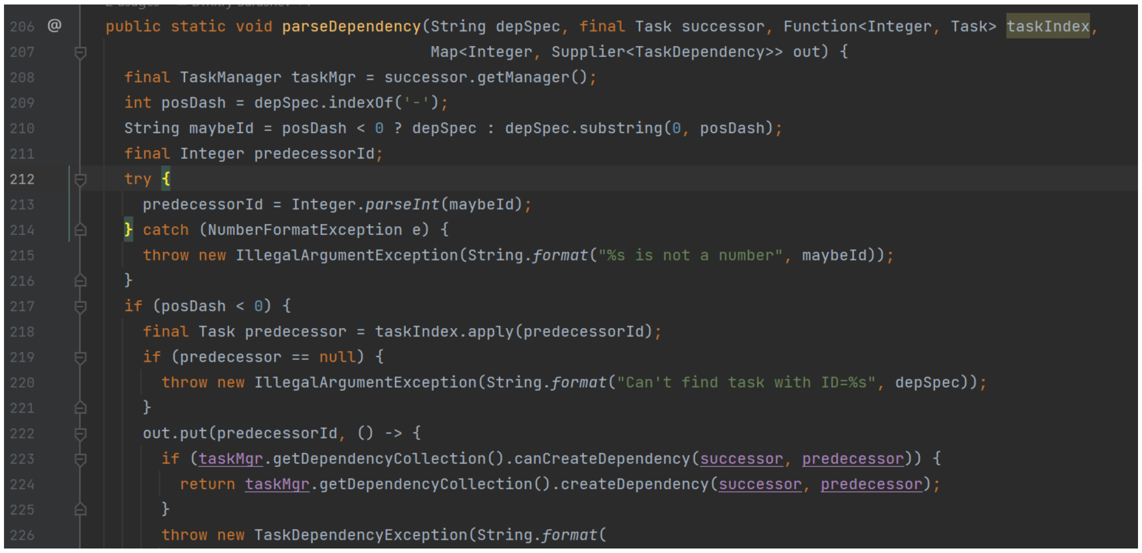


```java
static public String padRight(String string, int padding) {
  if (padding > 0) {
    padding += string.length();
    return String.format("%1$-" + padding + "s", string);
  }
  return string;
}
```

3. Long Method
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/task/TaskProperties.java

   Explanation:
   This method is very long with 79 lines and it’s confusing to understand it because it has a lot of if statements and no comments. The method is responsible for doing more things than it actually should.

   Reviewed by Rui Capareira.



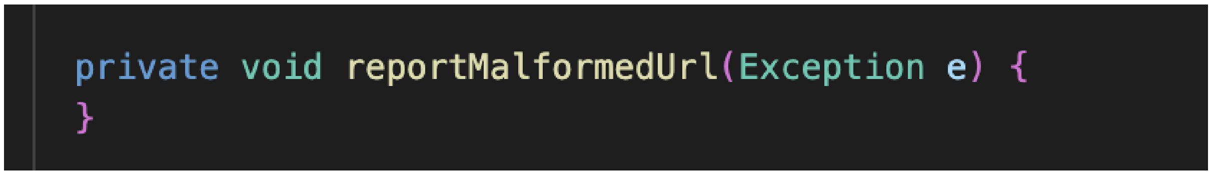
```java
public static void parseDependency(String depSpec, final Task successor, Function<Integer, Task> taskIndex,
                                   Map<Integer, Supplier<TaskDependency>> out) {
  final TaskManager taskMgr = successor.getManager();
  int posDash = depSpec.indexOf('-');
  String maybeId = posDash < 0 ? depSpec : depSpec.substring(0, posDash);
  final Integer predecessorId;
  try {
    predecessorId = Integer.parseInt(maybeId);
  } catch (NumberFormatException e) {
    throw new IllegalArgumentException(String.format("%s is not a number", maybeId));
  }
  if (posDash < 0) {
    final Task predecessor = taskIndex.apply(predecessorId);
    if (predecessor == null) {
      throw new IllegalArgumentException(String.format("Can't find task with ID=%s", depSpec));
    }
    out.put(predecessorId, () -> {
      if (taskMgr.getDependencyCollection().canCreateDependency(successor, predecessor)) {
        return taskMgr.getDependencyCollection().createDependency(successor, predecessor);
      }
      throw new TaskDependencyException(String.format(
```

Filipe Santo, 64859

1. Dead Code
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/wizard/AbstractFileChooser.java

   Explanation:
   This function doesn’t do nothing, is a dead code, it should be deleted.

   Reviewed by Jaime Russo.

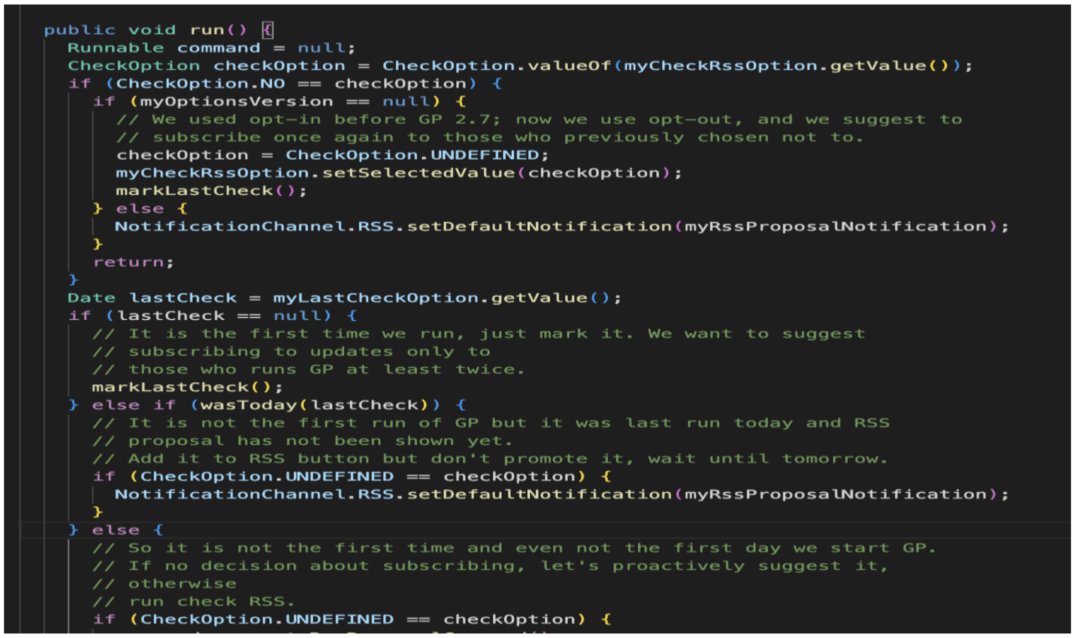


```java
private void reportMalformedUrl(Exception e) {
}
```

2. <u>Comments</u>
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/client/RssFeedChecker.java

   Explanation:
   Instead of having the comments explaining the following complex expression, the expression should go to another function and make it name self explanatory. The comments should be deleted.

   Reviewed by Carolina Simonet.

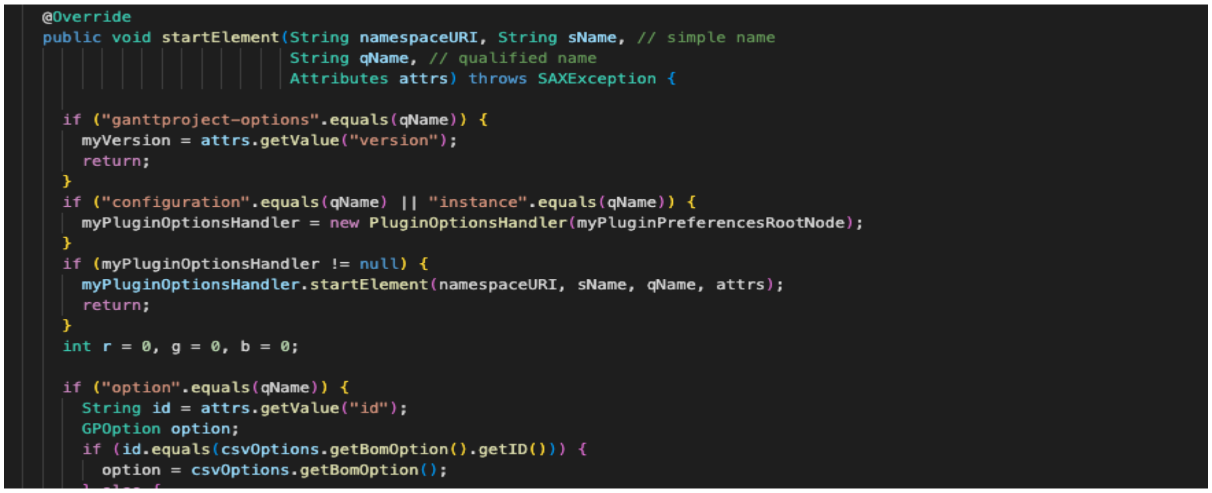


```
  public void run() {
    Runnable command = null;
    CheckOption checkOption = CheckOption.valueOf(myCheckRssOption.getValue());
    if (CheckOption.NO == checkOption) {
      if (myOptionsVersion == null) {
        // We used opt-in before GP 2.7; now we use opt-out, and we suggest to
        // subscribe once again to those who previously chosen not to.
        checkOption = CheckOption.UNDEFINED;
        myCheckRssOption.setSelectedValue(checkOption);
        markLastCheck();
      } else {
        NotificationChannel.RSS.setDefaultNotification(myRssProposalNotification);
      }
      return;
    }
    Date lastCheck = myLastCheckOption.getValue();
    if (lastCheck == null) {
      // It is the first time we run, just mark it. We want to suggest
      // subscribing to updates only to
      // those who runs GP at least twice.
      markLastCheck();
    } else if (wasToday(lastCheck)) {
      // It is not the first run of GP but it was last run today and RSS
      // proposal has not been shown yet.
      // Add it to RSS button but don't promote it, wait until tomorrow.
      if (CheckOption.UNDEFINED == checkOption) {
        NotificationChannel.RSS.setDefaultNotification(myRssProposalNotification);
      }
    } else {
      // So it is not the first time and even not the first day we start GP.
      // If no decision about subscribing, let's proactively suggest it,
      // otherwise
      // run check RSS.
      if (CheckOption.UNDEFINED == checkOption) {
```

3. <u>Long Method</u>
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/GanttOptions.java

   Explanation:
   This method is very long with 204 lines. It should be divided into multiple methods and if possible follow the SOLID principles because it accumulates a lot of responsibility making it very hard to understand.

   Reviewed by Margarida Carvalho.



```
@Override
public void startElement(String namespaceURI, String sName, // simple name
                         String qName, // qualified name
                         Attributes attrs) throws SAXException {

  if ("ganttproject-options".equals(qName)) {
    myVersion = attrs.getValue("version");
    return;
  }
  if ("configuration".equals(qName) || "instance".equals(qName)) {
    myPluginOptionsHandler = new PluginOptionsHandler(myPluginPreferencesRootNode);
  }
  if (myPluginOptionsHandler != null) {
    myPluginOptionsHandler.startElement(namespaceURI, sName, qName, attrs);
    return;
  }
  int r = 0, g = 0, b = 0;

  if ("option".equals(qName)) {
    String id = attrs.getValue("id");
    GPOption option;
    if (id.equals(csvOptions.getBomOption().getID())) {
      option = csvOptions.getBomOption();
```

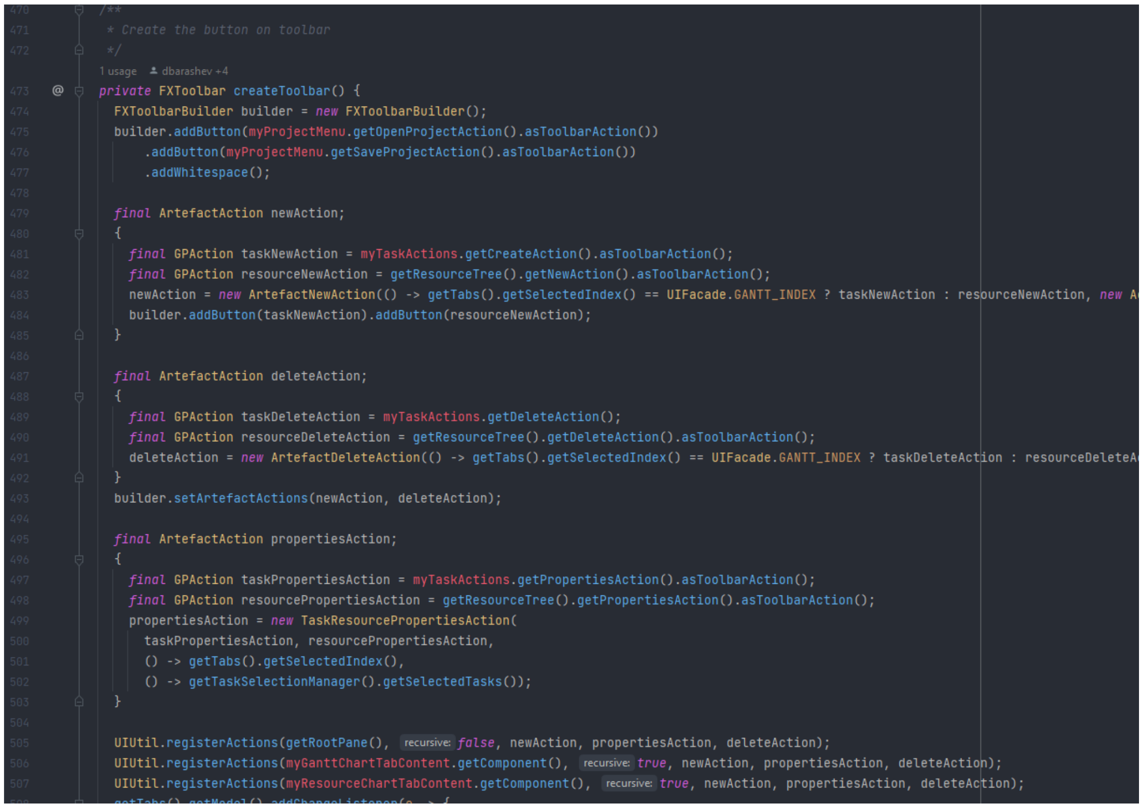
Jaime Russo, 60062

1. <u>Long Method</u>

   Location: ganttproject/src/main/java/net/sourceforge/ganttproject/GanttProject

   Explanation: This method has 60 lines, which we can identify as a long method code smell. To prevent this smell we could create additional auxiliar methods to help.

   Reviewed by Rui Capareira.

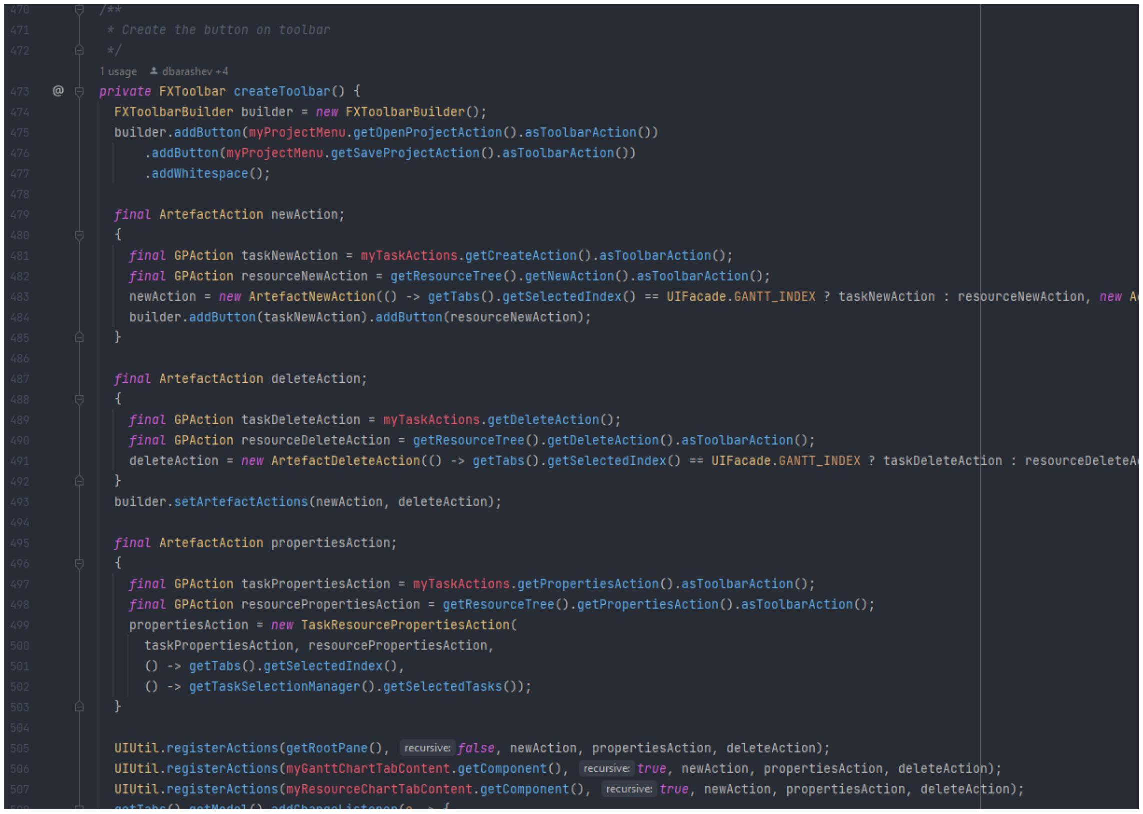


```
/**
 * Create the button on toolbar
 */
1 usage  dbarashev +4
private FXToolbar createToolbar() {
  FXToolbarBuilder builder = new FXToolbarBuilder();
  builder.addButton(myProjectMenu.getOpenProjectAction().asToolbarAction())
      .addButton(myProjectMenu.getSaveProjectAction().asToolbarAction())
      .addWhitespace();

  final ArtefactAction newAction;
  {
    final GPAction taskNewAction = myTaskActions.getCreateAction().asToolbarAction();
    final GPAction resourceNewAction = getResourceTree().getNewAction().asToolbarAction();
    newAction = new ArtefactNewAction(() -> getTabs().getSelectedIndex() == UIFacade.GANTT_INDEX ? taskNewAction : resourceNewAction, new A
    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 ? taskDeleteAction : resourceDeleteA
  }
  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());
  }

  UIUtil.registerActions(getRootPane(), recursive: false, newAction, propertiesAction, deleteAction);
  UIUtil.registerActions(myGanttChartTabContent.getComponent(), recursive: true, newAction, propertiesAction, deleteAction);
  UIUtil.registerActions(myResourceChartTabContent.getComponent(), recursive: true, newAction, propertiesAction, deleteAction);
  getTabs().getModel().addChangeListener(e -> {
```

2. <u>No Comments</u>
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/GanttProject

   Explanation: This method don't have any comments at all, only pure code, it may be confusing even for the coder.

   Reviewed by Margarida Carvalho.



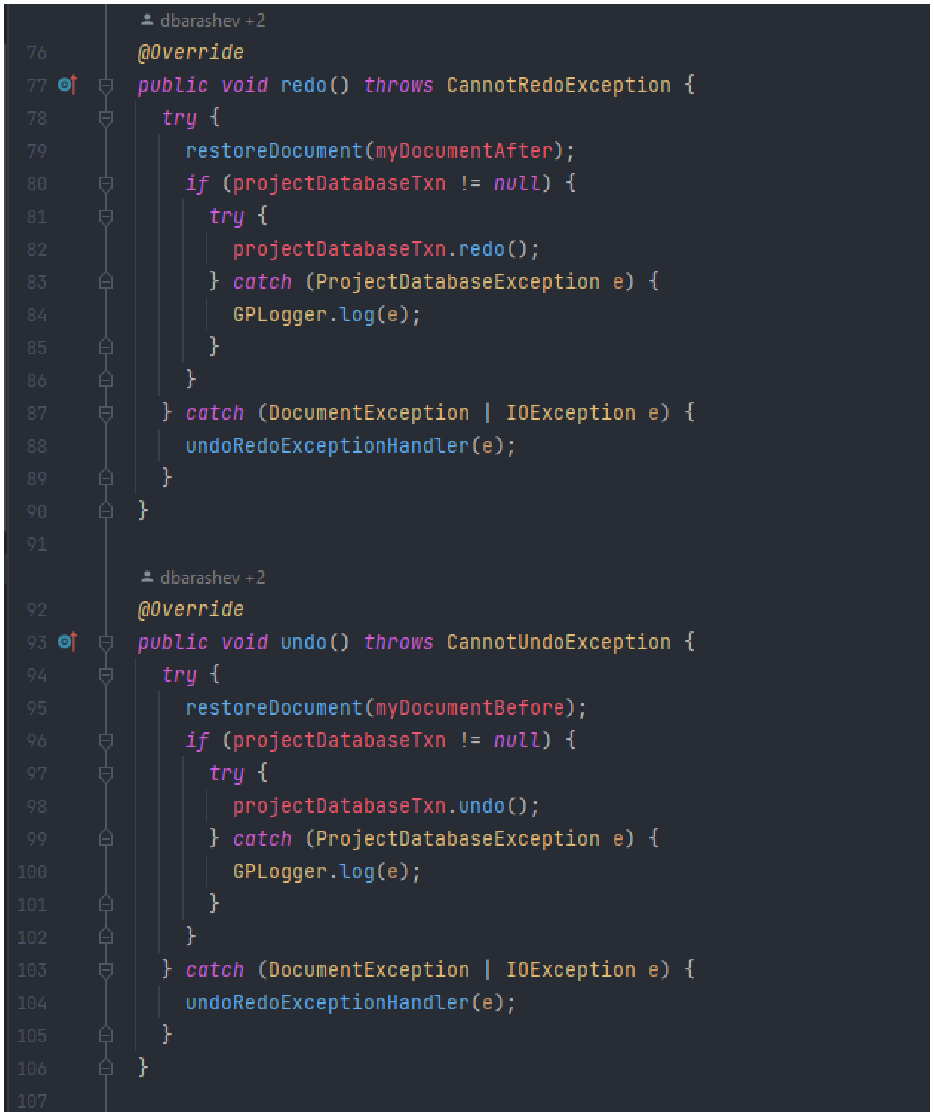
```java
/**
 * Create the button on toolbar
 */
1 usage   dbarashev +4
private FXToolbar createToolbar() {
  FXToolbarBuilder builder = new FXToolbarBuilder();
  builder.addButton(myProjectMenu.getOpenProjectAction().asToolbarAction())
      .addButton(myProjectMenu.getSaveProjectAction().asToolbarAction())
      .addWhitespace();

  final ArtefactAction newAction;
  {
    final GPAction taskNewAction = myTaskActions.getCreateAction().asToolbarAction();
    final GPAction resourceNewAction = getResourceTree().getNewAction().asToolbarAction();
    newAction = new ArtefactNewAction(() -> getTabs().getSelectedIndex() == UIFacade.GANTT_INDEX ? taskNewAction : resourceNewAction, new A
    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 ? taskDeleteAction : resourceDeleteA
  }
  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());
  }

  UIUtil.registerActions(getRootPane(), recursive: false, newAction, propertiesAction, deleteAction);
  UIUtil.registerActions(myGanttChartTabContent.getComponent(), recursive: true, newAction, propertiesAction, deleteAction);
  UIUtil.registerActions(myResourceChartTabContent.getComponent(), recursive: true, newAction, propertiesAction, deleteAction);
```

3. <u>Repeated Code</u>

   Location: ganttproject/src/main/java/net/sourceforge/ganttproject/undo/UndoableEditImpl

   Explanation: These methods have only 1 different line from one to the other, we could only have 1 method with a condition to implement 1 line or the other.

   Reviewed by Filipe Santo.



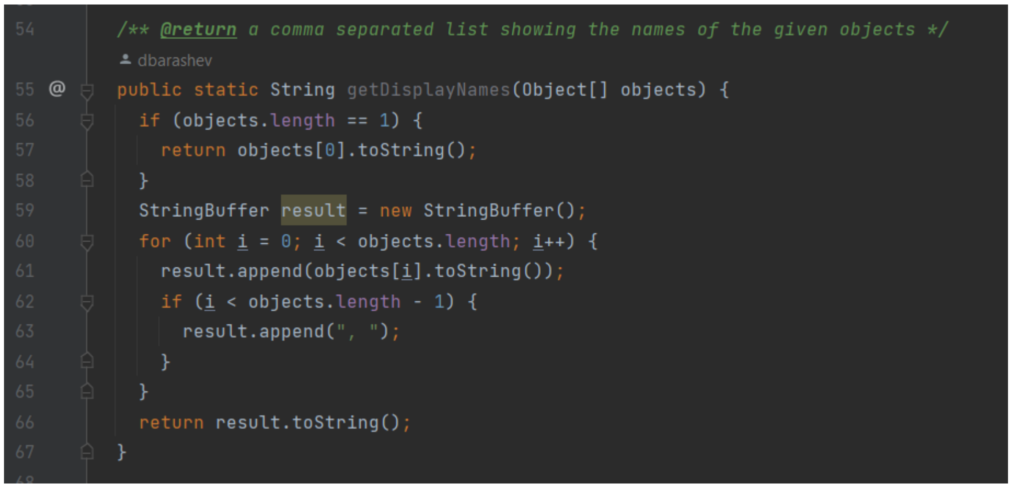
```
dbarashev +2
@Override
public void redo() throws CannotRedoException {
  try {
    restoreDocument(myDocumentAfter);
    if (projectDatabaseTxn != null) {
      try {
        projectDatabaseTxn.redo();
      } catch (ProjectDatabaseException e) {
        GPLogger.log(e);
      }
    }
  } catch (DocumentException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

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);
  }
}
```

Margarida Carvalho, 60437

1. Dead Code
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/util/StringUtils.java

   Explanation:
   The method is not used. To fix this code smell, the method should be deleted.

   Reviewed by Carolina Simonet.

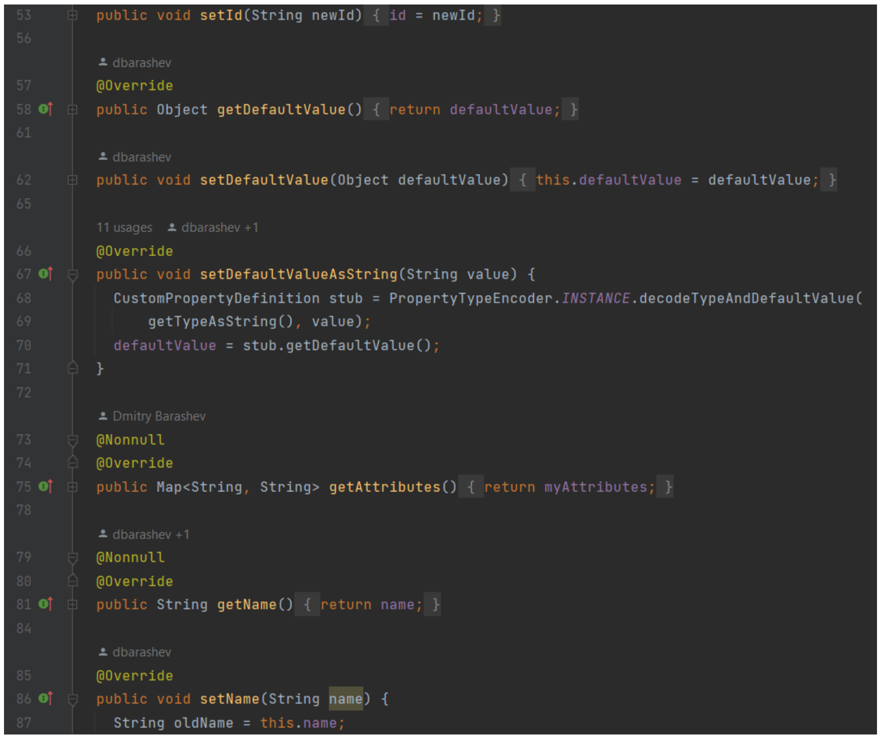


```
/** @return a comma separated list showing the names of the given objects */
dbarashev
public static String getDisplayNames(Object[] objects) {
  if (objects.length == 1) {
    return objects[0].toString();
  }
  StringBuffer result = new StringBuffer();
  for (int i = 0; i < objects.length; i++) {
    result.append(objects[i].toString());
    if (i < objects.length - 1) {
      result.append(", ");
    }
  }
  return result.toString();
}
```

2. <u>Data Class</u>
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/util/CustomColumn.java

   Explanation:
   This class doesn't contain real functionality. Besides the equals and hashCode methods, it only has getter and setter methods.

   Reviewed by Rui Capareira.

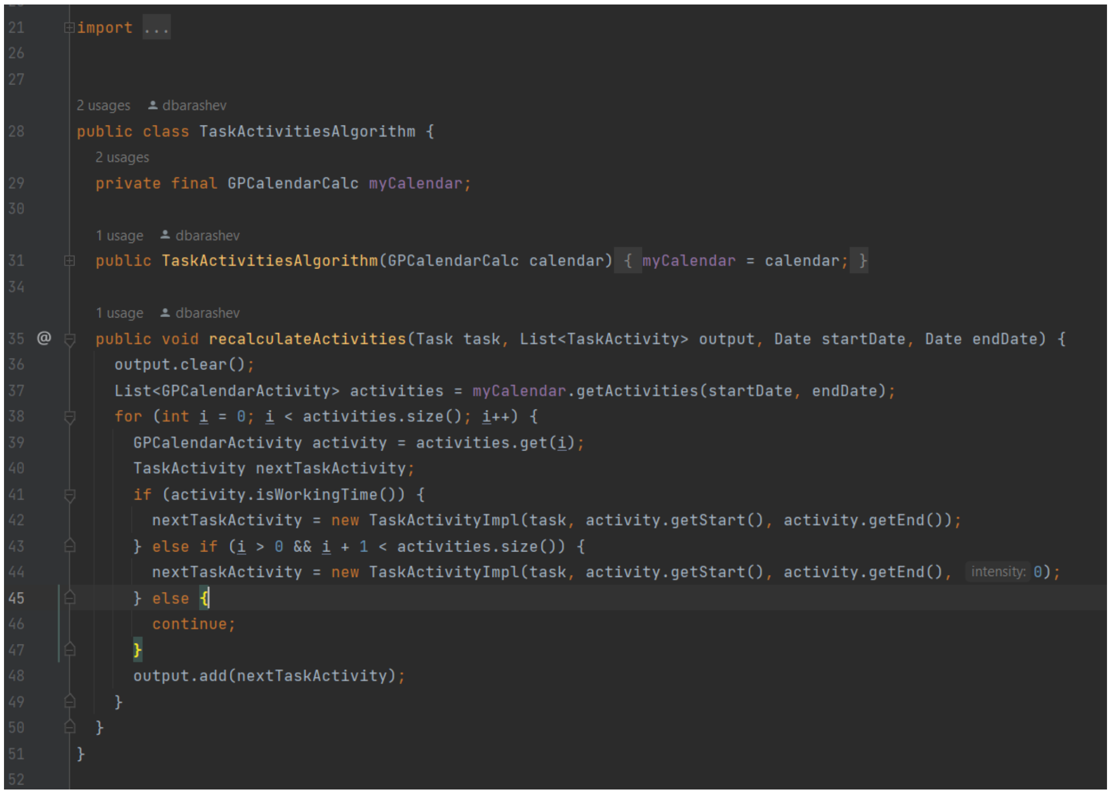


```
public void setId(String newId) { id = newId; }

dbarashev
@Override
public Object getDefaultValue() { return defaultValue; }

dbarashev
public void setDefaultValue(Object defaultValue) { this.defaultValue = defaultValue; }

11 usages  dbarashev +1
@Override
public void setDefaultValueAsString(String value) {
  CustomPropertyDefinition stub = PropertyTypeEncoder.INSTANCE.decodeTypeAndDefaultValue(
      getTypeAsString(), value);
  defaultValue = stub.getDefaultValue();
}

Dmitry Barashev
@Nonnull
@Override
public Map<String, String> getAttributes() { return myAttributes; }

dbarashev +1
@Nonnull
@Override
public String getName() { return name; }

dbarashev
@Override
public void setName(String name) {
  String oldName = this.name;
```

3. <u>Feature Envy</u>
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/task/TaskActivitiesAlgorithm.java

   Explanation:
   This method manipulates the data of another class. It could have been made in the TaskActivity class.

   Reviewed by Jaime Russo.



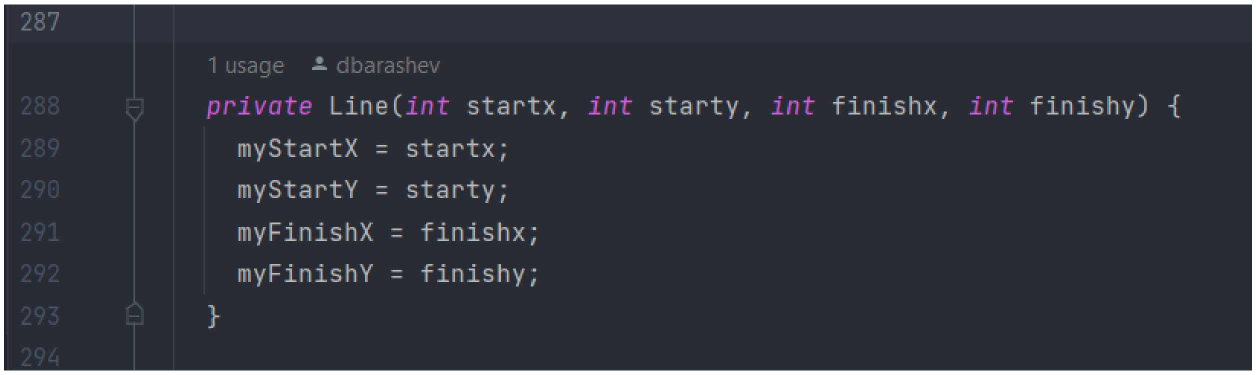
```
import ...


2 usages  dbarashev
public class TaskActivitiesAlgorithm {
  2 usages
  private final GPCalendarCalc myCalendar;

  1 usage  dbarashev
  public TaskActivitiesAlgorithm(GPCalendarCalc calendar) { myCalendar = calendar; }

  1 usage  dbarashev
  public void recalculateActivities(Task task, List<TaskActivity> output, Date startDate, Date endDate) {
    output.clear();
    List<GPCalendarActivity> activities = myCalendar.getActivities(startDate, endDate);
    for (int i = 0; i < activities.size(); i++) {
      GPCalendarActivity activity = activities.get(i);
      TaskActivity nextTaskActivity;
      if (activity.isWorkingTime()) {
        nextTaskActivity = new TaskActivityImpl(task, activity.getStart(), activity.getEnd());
      } else if (i > 0 && i + 1 < activities.size()) {
        nextTaskActivity = new TaskActivityImpl(task, activity.getStart(), activity.getEnd(), intensity: 0);
      } else {
        continue;
      }
      output.add(nextTaskActivity);
    }
  }
}
```

Rui Capareira, 57046

1. <u>Data Clump</u>
   Location:
   ganttproject\biz.ganttproject.core\src\main\java\biz\ganttproject\core\chart\canvas\Canvas.java

   Explanation: This method could be rewritten by passing two 2DPoint objects as arguments, instead it uses a group of variables passed as a clump.

   Reviewed by Filipe Santo.

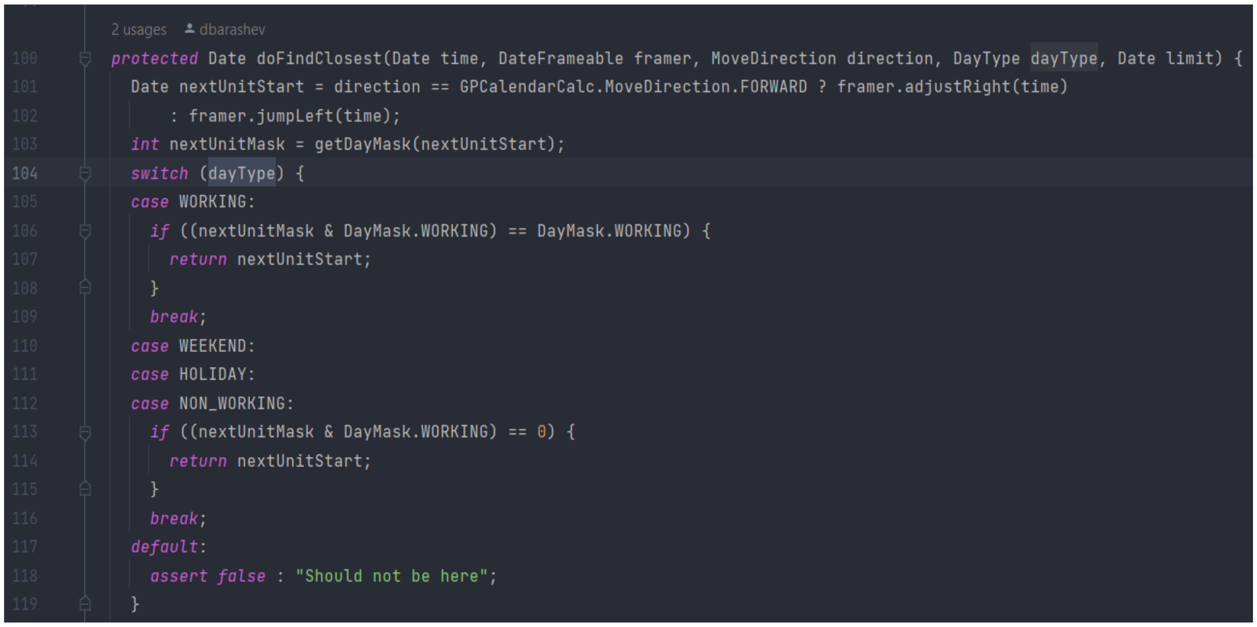


```
1 usage   dbarashev
private Line(int startx, int starty, int finishx, int finishy) {
  myStartX = startx;
  myStartY = starty;
  myFinishX = finishx;
  myFinishY = finishy;
}
```

2. <u>Switch Statements</u>
   Location:
   ganttproject\biz.ganttproject.core\src\main\java\biz\ganttproject\core\calendar\GPCalendarBase.java

   Explanation:
   Switch statement with conditionals checking on type instead of reducing conditionals down to a design that uses polymorphism. Two switch cases evaluated without any code being executed.

   Reviewed by Jaime Russo.

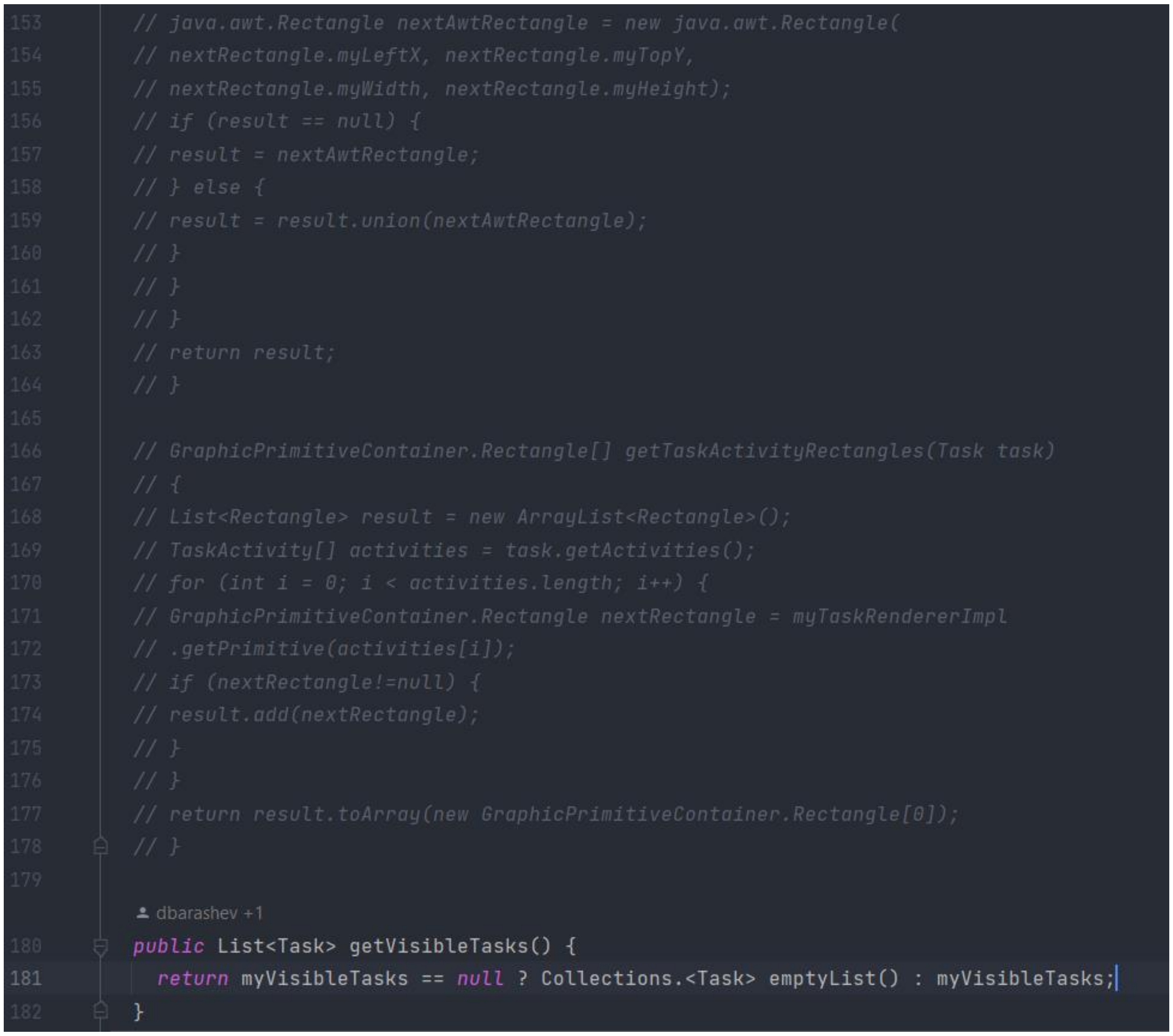

2 usages dbarashev

```java
protected Date doFindClosest(Date time, DateFrameable framer, MoveDirection direction, DayType dayType, Date limit) {
  Date nextUnitStart = direction == GPCalendarCalc.MoveDirection.FORWARD ? framer.adjustRight(time)
      : framer.jumpLeft(time);
  int nextUnitMask = getDayMask(nextUnitStart);
  switch (dayType) {
  case WORKING:
    if ((nextUnitMask & DayMask.WORKING) == DayMask.WORKING) {
      return nextUnitStart;
    }
    break;
  case WEEKEND:
  case HOLIDAY:
  case NON_WORKING:
    if ((nextUnitMask & DayMask.WORKING) == 0) {
      return nextUnitStart;
    }
    break;
  default:
    assert false : "Should not be here";
  }
```

3. <u>Comments</u>
   Location:
   ganttproject\ganttproject\src\main\java\net\sourceforge\ganttproject\chart\ChartModelImpl.java

   Explanation:
   This section of code shows several lines of comments that take on a reminder nature, showing that something needs to be done or this code needs to be updated later.

   Reviewed by Carolina Simonet.



```java
// 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);
// }
// }
// return result.toArray(new GraphicPrimitiveContainer.Rectangle[0]);
// }

public List<Task> getVisibleTasks() {
  return myVisibleTasks == null ? Collections.<Task> emptyList() : myVisibleTasks;
}
```

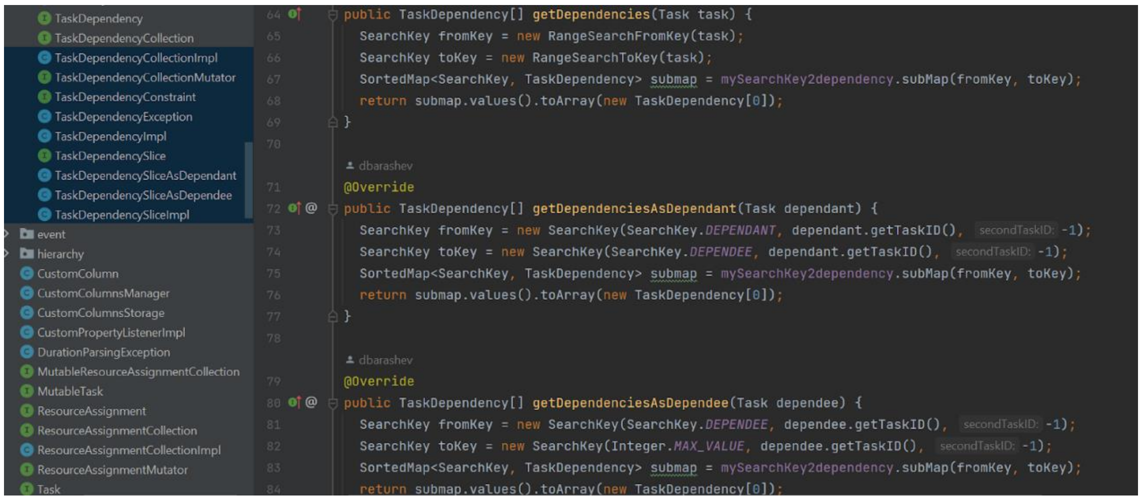
Design Patterns

Carolina Simonet, 59748

1. <u>Facade Pattern</u>
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/task/dependency/TaskDependencyCollectionImpl.java

   Explanation:
   This class works with a subsystem of other classes to be easier to access all of them through this TaskDependencyCollectionImpl class.

   Reviewed by Filipe Santo.

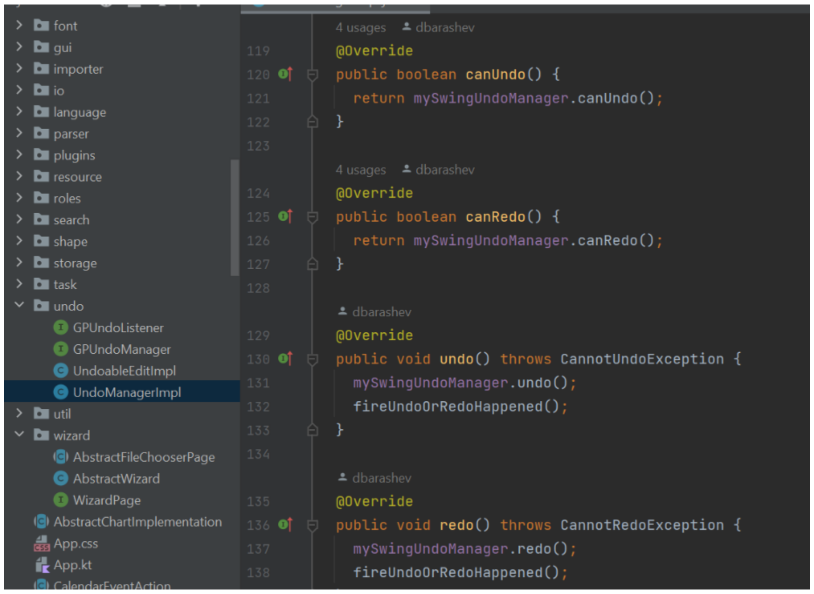


```java
public TaskDependency[] getDependencies(Task task) {
  SearchKey fromKey = new RangeSearchFromKey(task);
  SearchKey toKey = new RangeSearchToKey(task);
  SortedMap<SearchKey, TaskDependency> submap = mySearchKey2dependency.subMap(fromKey, toKey);
  return submap.values().toArray(new TaskDependency[0]);
}

dbarashev
@Override
public TaskDependency[] getDependenciesAsDependant(Task dependant) {
  SearchKey fromKey = new SearchKey(SearchKey.DEPENDANT, dependant.getTaskID(), secondTaskID: -1);
  SearchKey toKey = new SearchKey(SearchKey.DEPENDEE, dependant.getTaskID(), secondTaskID: -1);
  SortedMap<SearchKey, TaskDependency> submap = mySearchKey2dependency.subMap(fromKey, toKey);
  return submap.values().toArray(new TaskDependency[0]);
}

dbarashev
@Override
public TaskDependency[] getDependenciesAsDependee(Task dependee) {
  SearchKey fromKey = new SearchKey(SearchKey.DEPENDEE, dependee.getTaskID(), secondTaskID: -1);
  SearchKey toKey = new SearchKey(Integer.MAX_VALUE, dependee.getTaskID(), secondTaskID: -1);
  SortedMap<SearchKey, TaskDependency> submap = mySearchKey2dependency.subMap(fromKey, toKey);
  return submap.values().toArray(new TaskDependency[0]);
```

2. <u>Command Pattern</u>
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/undo/UndoManagerImpl.java

   Explanation:
   Commands are being manipulated as objects, allows to do and undo operations.

   Reviewed by Rui Capareira.



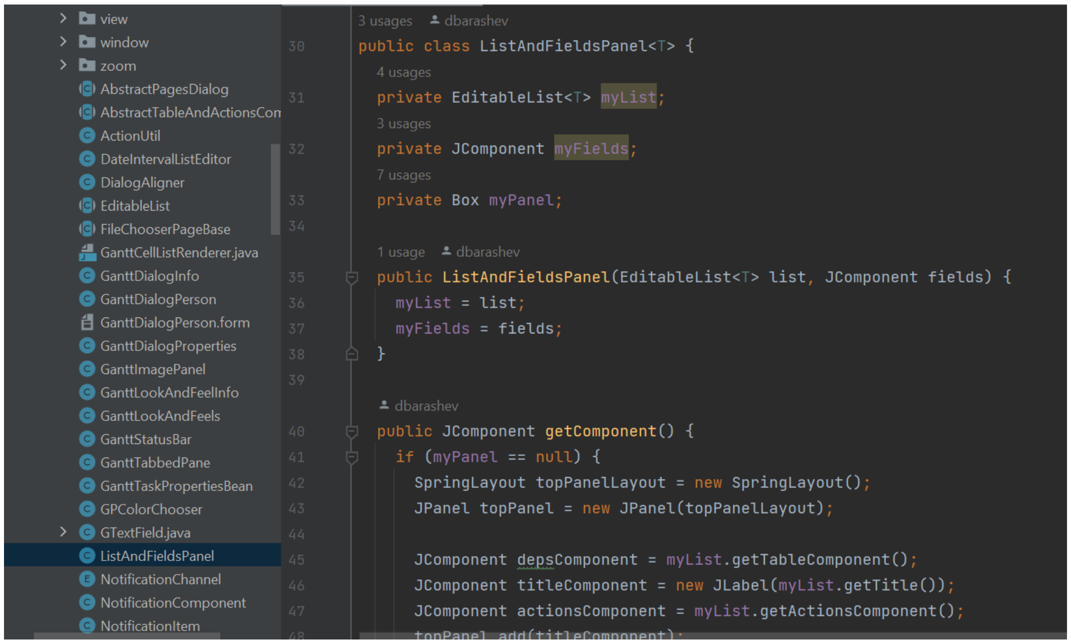
```
font
gui
importer
io
language
parser
plugins
resource
roles
search
shape
storage
task
undo
  GPUndoListener
  GPUndoManager
  UndoableEditImpl
  UndoManagerImpl
util
wizard
  AbstractFileChooserPage
  AbstractWizard
  WizardPage
AbstractChartImplementation
App.css
App.kt
CalendarEventAction
```

```
    4 usages  dbarashev
    @Override
    public boolean canUndo() {
      return mySwingUndoManager.canUndo();
    }

    4 usages  dbarashev
    @Override
    public boolean canRedo() {
      return mySwingUndoManager.canRedo();
    }

    dbarashev
    @Override
    public void undo() throws CannotUndoException {
      mySwingUndoManager.undo();
      fireUndoOrRedoHappened();
    }

    dbarashev
    @Override
    public void redo() throws CannotRedoException {
      mySwingUndoManager.redo();
      fireUndoOrRedoHappened();
```

3. <u>Singleton Pattern</u>
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/gui/ListAndFieldsPanel.java

   Explanation:
   Creates an object, the class has only one instance that generates a new box and returns it.

   Reviewed by Jaime Russo.



```
view
window
zoom
AbstractPagesDialog
AbstractTableAndActionsCom
ActionUtil
DateIntervalListEditor
DialogAligner
EditableList
FileChooserPageBase
GanttCellListRenderer.java
GanttDialogInfo
GanttDialogPerson
GanttDialogPerson.form
GanttDialogProperties
GanttImagePanel
GanttLookAndFeelInfo
GanttLookAndFeels
GanttStatusBar
GanttTabbedPane
GanttTaskPropertiesBean
GPColorChooser
GTextField.java
ListAndFieldsPanel
NotificationChannel
NotificationComponent
NotificationItem
```

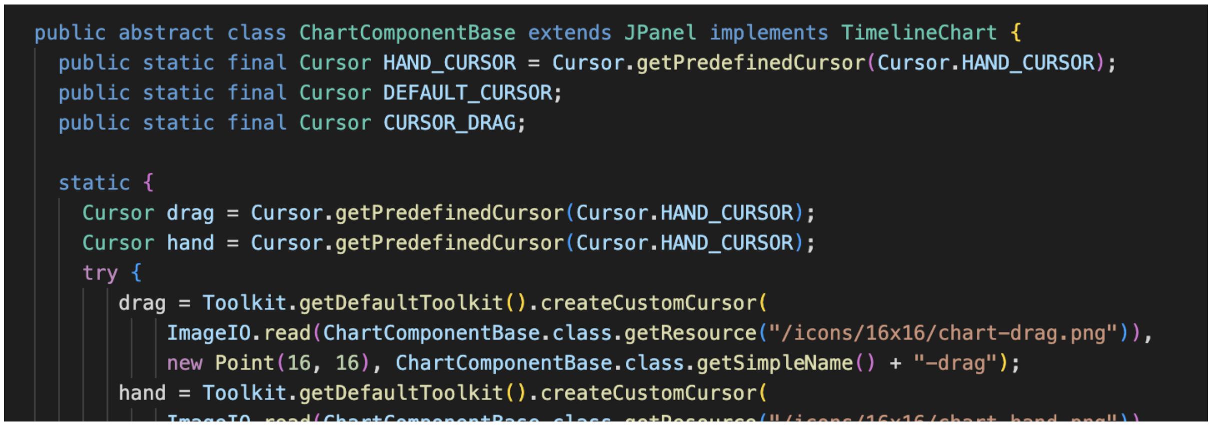
```
3 usages  dbarashev
public class ListAndFieldsPanel<T> {
  4 usages
  private EditableList<T> myList;
  3 usages
  private JComponent myFields;
  7 usages
  private Box myPanel;

  1 usage  dbarashev
  public ListAndFieldsPanel(EditableList<T> list, JComponent fields) {
    myList = list;
    myFields = fields;
  }

  dbarashev
  public JComponent getComponent() {
    if (myPanel == null) {
      SpringLayout topPanelLayout = new SpringLayout();
      JPanel topPanel = new JPanel(topPanelLayout);

      JComponent depsComponent = myList.getTableComponent();
      JComponent titleComponent = new JLabel(myList.getTitle());
      JComponent actionsComponent = myList.getActionsComponent();
      topPanel.add(titleComponent);
```

Filipe Santo, 64859

1. <u>Prototype Pattern</u>
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/ChartComponentBase.java

   Explanation:
   This class lets us create copies of objects without depending on the concrete class.

   Reviewed by Jaime Russo.

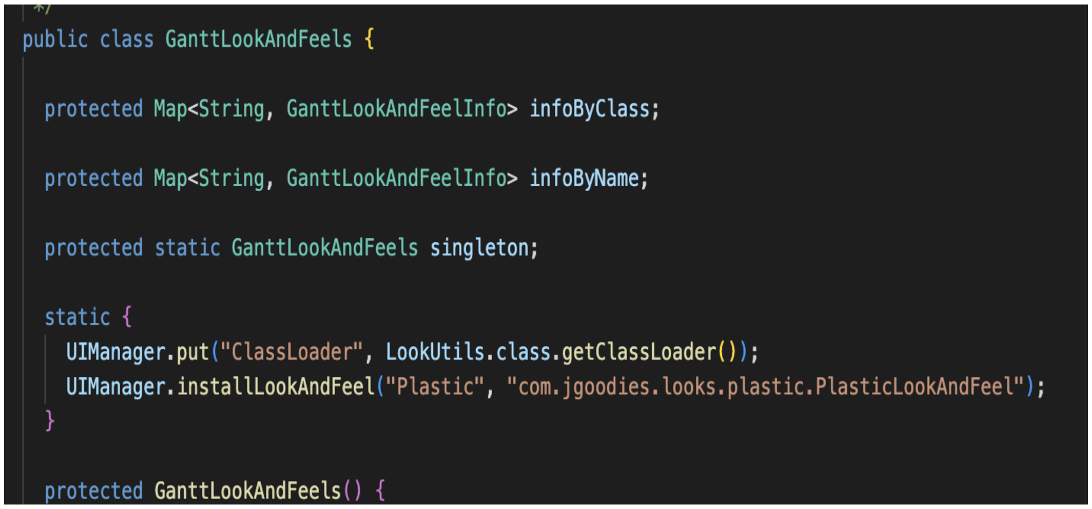


```java
public abstract class ChartComponentBase extends JPanel implements TimelineChart {
  public static final Cursor HAND_CURSOR = Cursor.getPredefinedCursor(Cursor.HAND_CURSOR);
  public static final Cursor DEFAULT_CURSOR;
  public static final Cursor CURSOR_DRAG;

  static {
    Cursor drag = Cursor.getPredefinedCursor(Cursor.HAND_CURSOR);
    Cursor hand = Cursor.getPredefinedCursor(Cursor.HAND_CURSOR);
    try {
       drag = Toolkit.getDefaultToolkit().createCustomCursor(
           ImageIO.read(ChartComponentBase.class.getResource("/icons/16x16/chart-drag.png")),
           new Point(16, 16), ChartComponentBase.class.getSimpleName() + "-drag");
       hand = Toolkit.getDefaultToolkit().createCustomCursor(
           ImageIO.read(ChartComponentBase.class.getResource("/icons/16x16/chart-hand.png")),
```

2. <u>Singleton Pattern</u>
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/gui/GanttLookAndFeels.java

   Explanation:
   This singleton class provides us global access to the info about the installed LookAndFeels.

   Reviewed by Carolina Simonet.

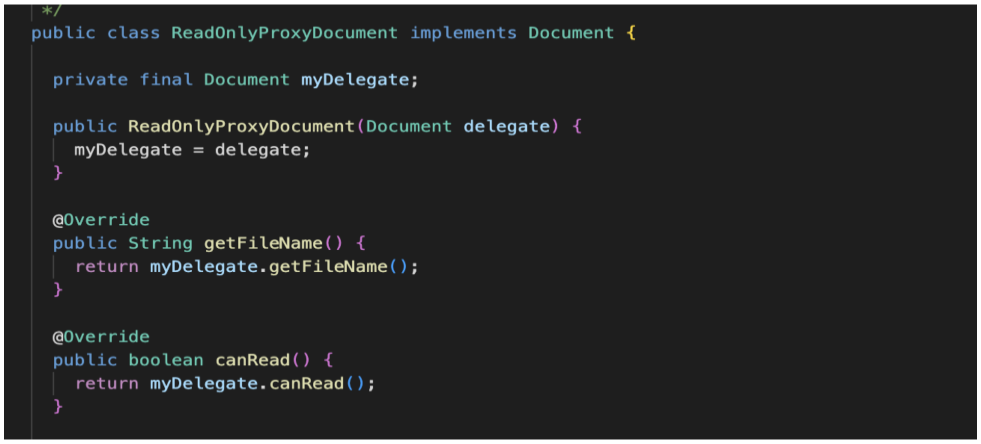


```java
 */
public class GanttLookAndFeels {

  protected Map<String, GanttLookAndFeelInfo> infoByClass;

  protected Map<String, GanttLookAndFeelInfo> infoByName;

  protected static GanttLookAndFeels singleton;

  static {
    UIManager.put("ClassLoader", LookUtils.class.getClassLoader());
    UIManager.installLookAndFeel("Plastic", "com.jgoodies.looks.plastic.PlasticLookAndFeel");
  }

  protected GanttLookAndFeels() {
```

3. <u>Proxy Pattern</u>
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/document/ReadOnlyProxyDocument.java

   Explanation:
   This class has the same interface as the original service object, and when it's updated it passes from this class to the original documents object. Delegating it all the work to it.

   Reviewed by Margarida Carvalho.



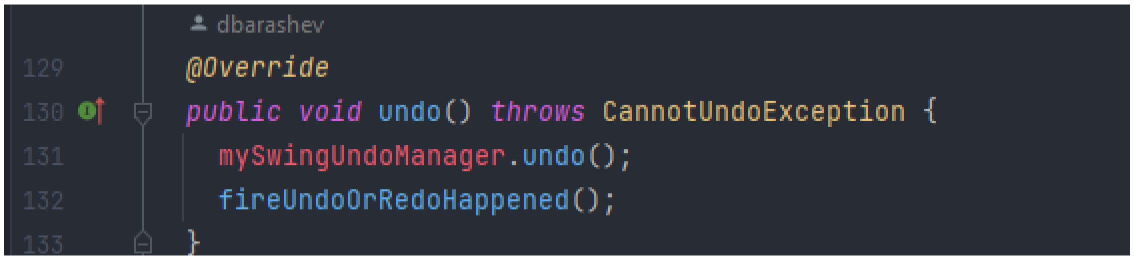
```java
 */
public class ReadOnlyProxyDocument implements Document {

  private final Document myDelegate;

  public ReadOnlyProxyDocument(Document delegate) {
    myDelegate = delegate;
  }

  @Override
  public String getFileName() {
    return myDelegate.getFileName();
  }

  @Override
  public boolean canRead() {
    return myDelegate.canRead();
  }
```

Jaime Russo, 60062

1. <u>Memento Pattern</u>
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/undo/UndoManagerImpl

   Explanation: In this algorithm, there is a class that saves the last state of the object, so we can undo and redo whenever we need.

   Reviewed by Rui Capareira.

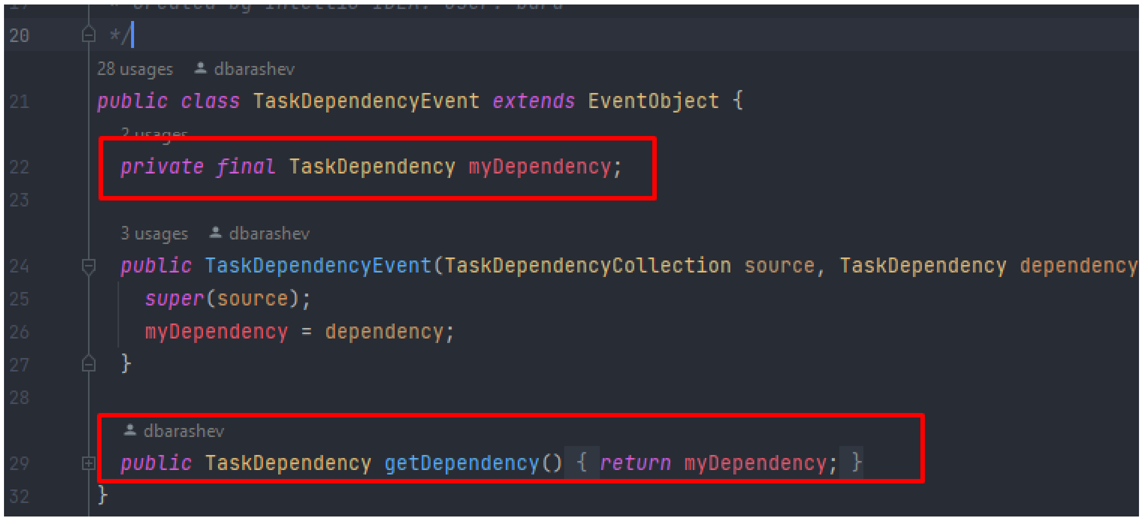


```
dbarashev
@Override
public void undo() throws CannotUndoException {
  mySwingUndoManager.undo();
  fireUndoOrRedoHappened();
}
```

2. <u>Singleton Pattern</u>
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/task/event/TaskDependencyEvent

   Explanation: This class only have one instance and that instance can be accessed through a method.

   Reviewed by Margarida Carvalho.

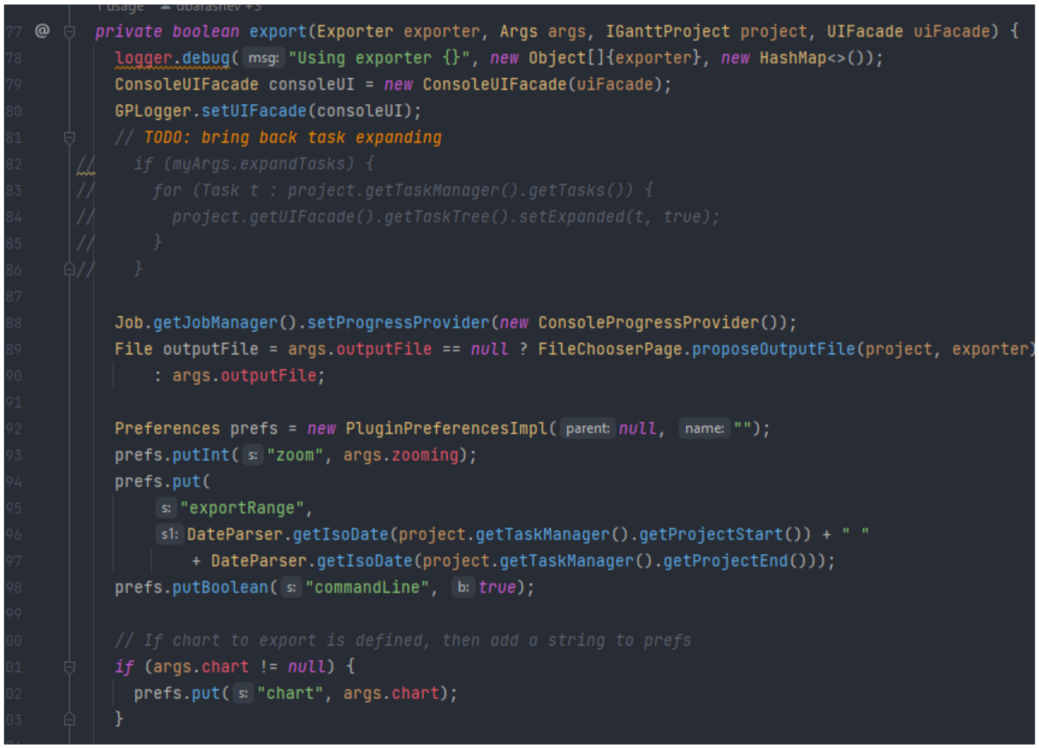


```
*/
28 usages  dbarashev
public class TaskDependencyEvent extends EventObject {
  2 usages
  private final TaskDependency myDependency;

  3 usages  dbarashev
  public TaskDependencyEvent(TaskDependencyCollection source, TaskDependency dependency
    super(source);
    myDependency = dependency;
  }

  dbarashev
  public TaskDependency getDependency() { return myDependency; }
}
```

3. <u>Command Pattern</u>
   Location: ganttproject/src/main/java/net/sourceforge/ganttproject/export/CommandLineExportApplication

   Explanation: This class will export the command line text, other class will cal this one to do that job.

   Reviewed by Carolina Simonet.



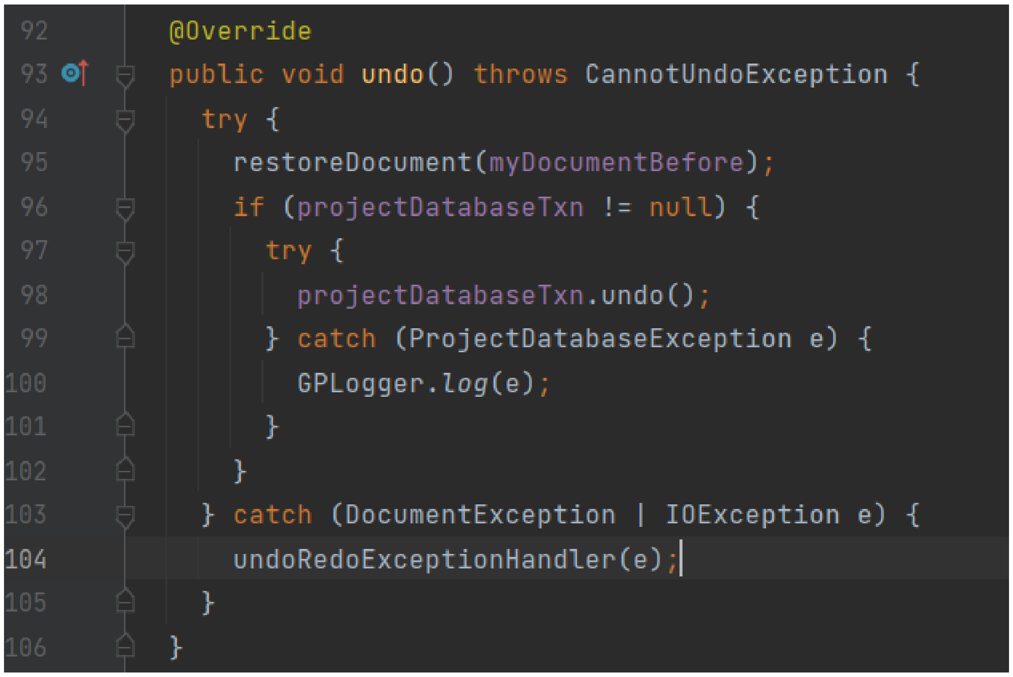
```java
private boolean export(Exporter exporter, Args args, IGanttProject project, UIFacade uiFacade) {
  logger.debug("Using exporter {}", new Object[]{exporter}, new HashMap<>());
  ConsoleUIFacade consoleUI = new ConsoleUIFacade(uiFacade);
  GPLogger.setUIFacade(consoleUI);
  // TODO: bring back task expanding
//    if (myArgs.expandTasks) {
//      for (Task t : project.getTaskManager().getTasks()) {
//        project.getUIFacade().getTaskTree().setExpanded(t, true);
//      }
//    }

  Job.getJobManager().setProgressProvider(new ConsoleProgressProvider());
  File outputFile = args.outputFile == null ? FileChooserPage.proposeOutputFile(project, exporter)
      : args.outputFile;

  Preferences prefs = new PluginPreferencesImpl(null, "");
  prefs.putInt("zoom", args.zooming);
  prefs.put(
      "exportRange",
      DateParser.getIsoDate(project.getTaskManager().getProjectStart()) + " "
          + DateParser.getIsoDate(project.getTaskManager().getProjectEnd()));
  prefs.putBoolean("commandLine", true);

  // If chart to export is defined, then add a string to prefs
  if (args.chart != null) {
    prefs.put("chart", args.chart);
  }
```

Margarida Carvalho, 60437

1. <u>Memento Pattern</u>
   Location: ganttproject/src/main/java/net/sourceforge/ganttproject/undo/UndoableEditImpl.java

   Explanation:
   With this method, we are able to access a previous state of an object and return it.

   Reviewed by Carolina Simonet.

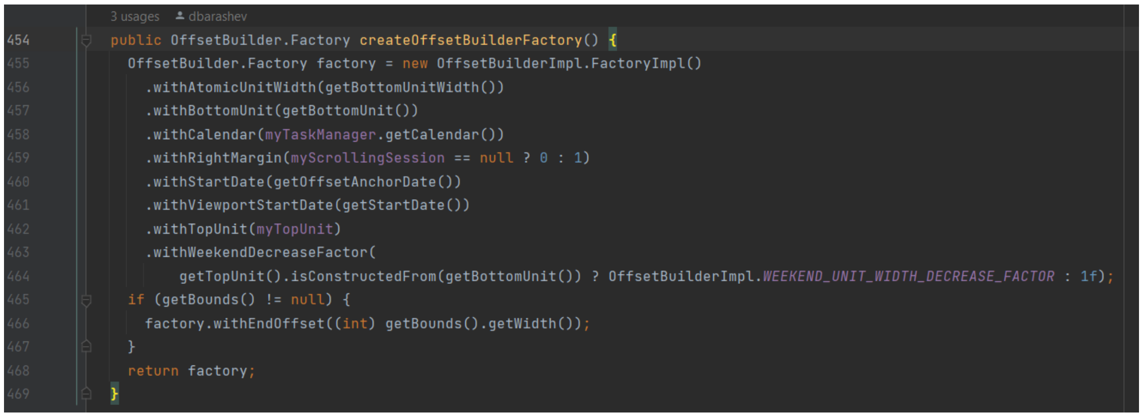


```java
@Override
public void undo() throws CannotUndoException {
  try {
    restoreDocument(myDocumentBefore);
    if (projectDatabaseTxn != null) {
      try {
        projectDatabaseTxn.undo();
      } catch (ProjectDatabaseException e) {
        GPLogger.log(e);
      }
    }
  } catch (DocumentException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

2. <u>Factory Method</u>
   Location:
   ganttproject/src/main/java/net/sourceforge/ganttproject/chart/ChartModelBase.java

   Explanation:
   It allows the creation of objects in a superclass.

   Reviewed by Filipe Santo.

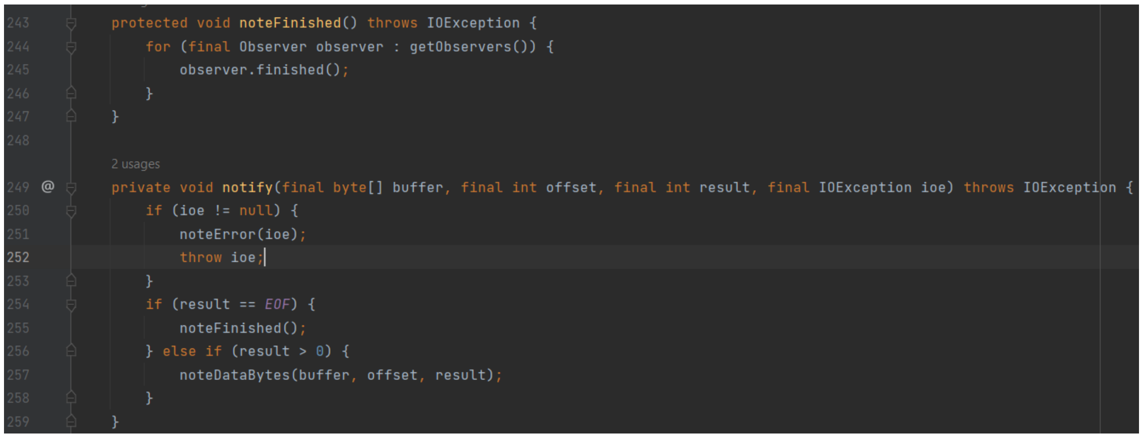


```java
3 usages  dbarashev
public OffsetBuilder.Factory createOffsetBuilderFactory() {
  OffsetBuilder.Factory factory = new OffsetBuilderImpl.FactoryImpl()
    .withAtomicUnitWidth(getBottomUnitWidth())
    .withBottomUnit(getBottomUnit())
    .withCalendar(myTaskManager.getCalendar())
    .withRightMargin(myScrollingSession == null ? 0 : 1)
    .withStartDate(getOffsetAnchorDate())
    .withViewportStartDate(getStartDate())
    .withTopUnit(myTopUnit)
    .withWeekendDecreaseFactor(
        getTopUnit().isConstructedFrom(getBottomUnit()) ? OffsetBuilderImpl.WEEKEND_UNIT_WIDTH_DECREASE_FACTOR : 1f);
  if (getBounds() != null) {
    factory.withEndOffset((int) getBounds().getWidth());
  }
  return factory;
}
```

3. <u>Observer Pattern</u>
   Location:
   org/apache/commons/io/input/ObservableInputStream.java

   Explanation:
   It is a mechanism that let us notify multiple objects about anything that happens to the observed object.

   Reviewed by Rui Capareira.



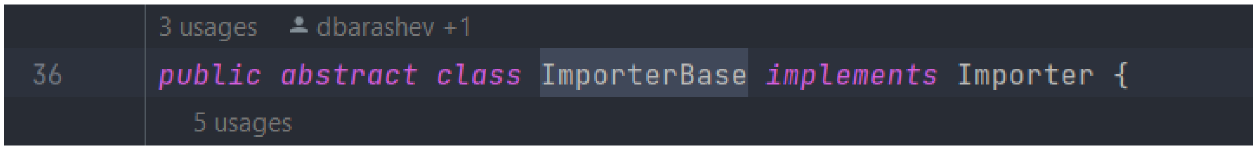
```
    protected void noteFinished() throws IOException {
        for (final Observer observer : getObservers()) {
            observer.finished();
        }
    }

    2 usages
    private void notify(final byte[] buffer, final int offset, final int result, final IOException ioe) throws IOException {
        if (ioe != null) {
            noteError(ioe);
            throw ioe;
        }
        if (result == EOF) {
            noteFinished();
        } else if (result > 0) {
            noteDataBytes(buffer, offset, result);
        }
    }
```

Rui Capareira, 57046

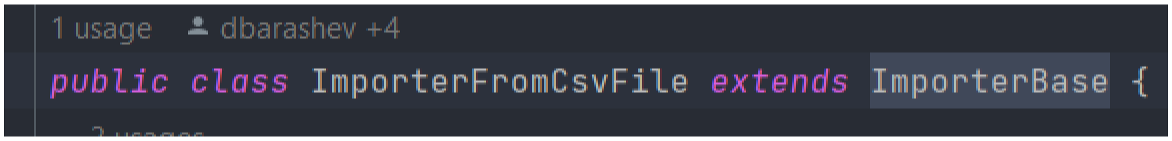
1. <u>Template Method Pattern</u>
   Location:
   ganttproject\ganttproject\src\main\java\net\sourceforge\ganttproject\importer\ImporterBase.java

   Explanation:
   Defines a general implementation for import related features, deferring the implementation of more specific steps to subclasses.

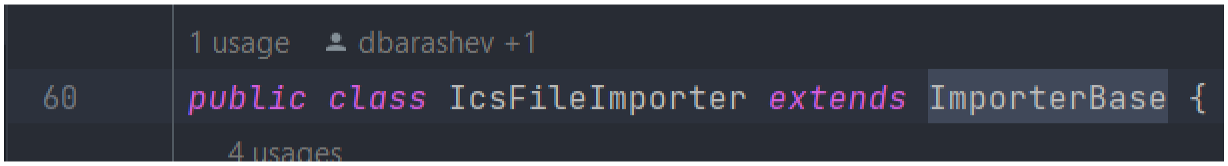
   Reviewed by Margarida Carvalho.



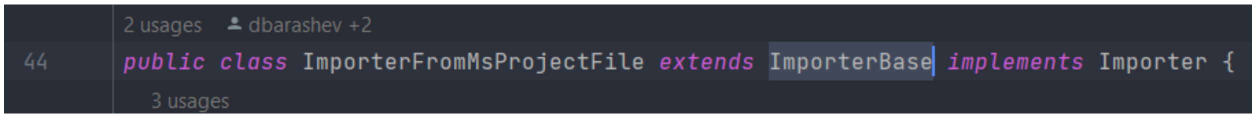
```
3 usages  dbarashev +1
public abstract class ImporterBase implements Importer {
   5 usages
```



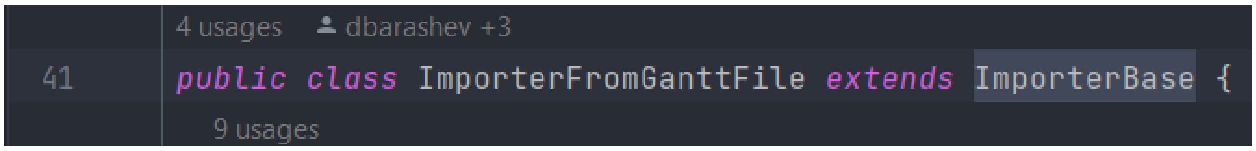
```
1 usage  dbarashev +4
public class ImporterFromCsvFile extends ImporterBase {
```



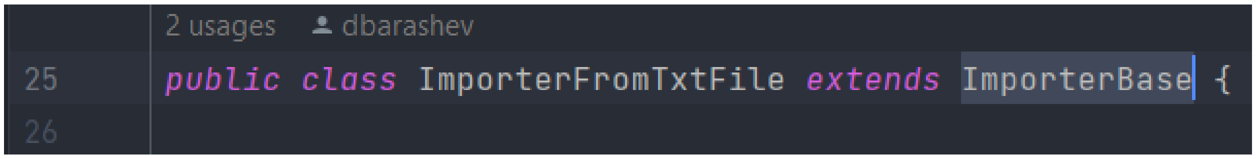
```
1 usage  dbarashev +1
public class IcsFileImporter extends ImporterBase {
    4 usages
```



```
2 usages  dbarashev +2
public class ImporterFromMsProjectFile extends ImporterBase implements Importer {
    3 usages
```



```
4 usages  dbarashev +3
public class ImporterFromGanttFile extends ImporterBase {
    9 usages
```

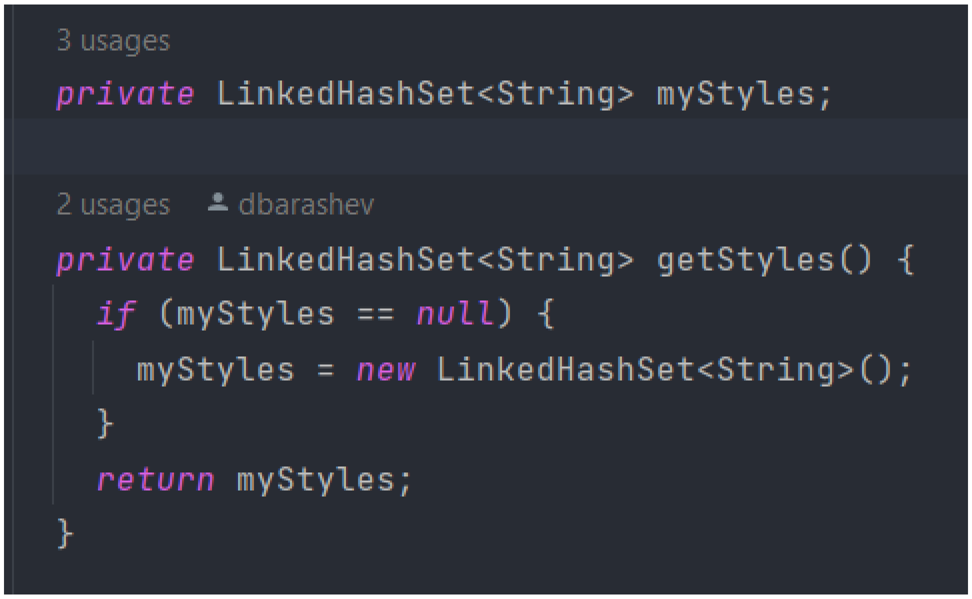


```
2 usages  dbarashev
public class ImporterFromTxtFile extends ImporterBase {
```

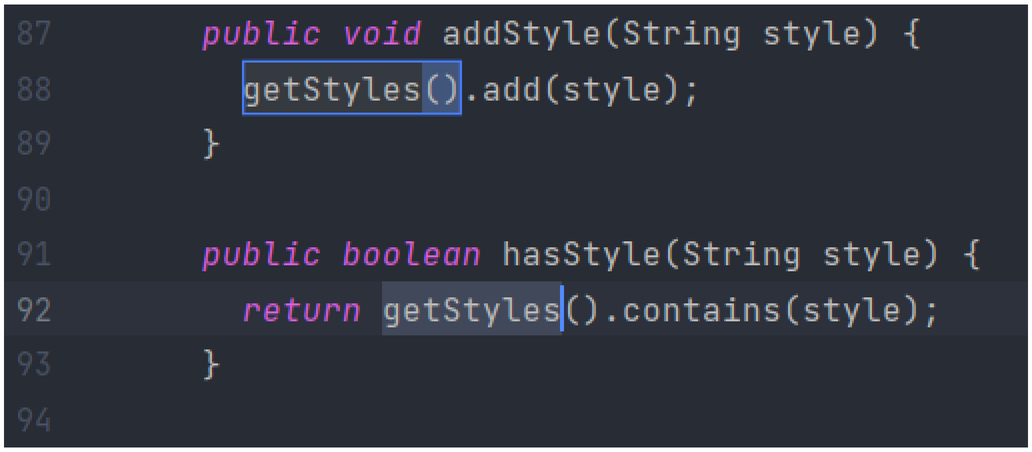
2. <u>Singleton Pattern</u>
   Location:
   ganttproject\biz.ganttproject.core\src\main\java\biz\ganttproject\core\chart\canvas\Canvas.java

   Explanation:
   The Singleton myStyles can only be accessed through its instance operation. The constructor is private and the public methods instantiate the singleton if it still doesn't exist.

   Reviewed by Jaime Russo.



```
3 usages
private LinkedHashSet<String> myStyles;

2 usages   dbarashev
private LinkedHashSet<String> getStyles() {
  if (myStyles == null) {
    myStyles = new LinkedHashSet<String>();
  }
  return myStyles;
}
```

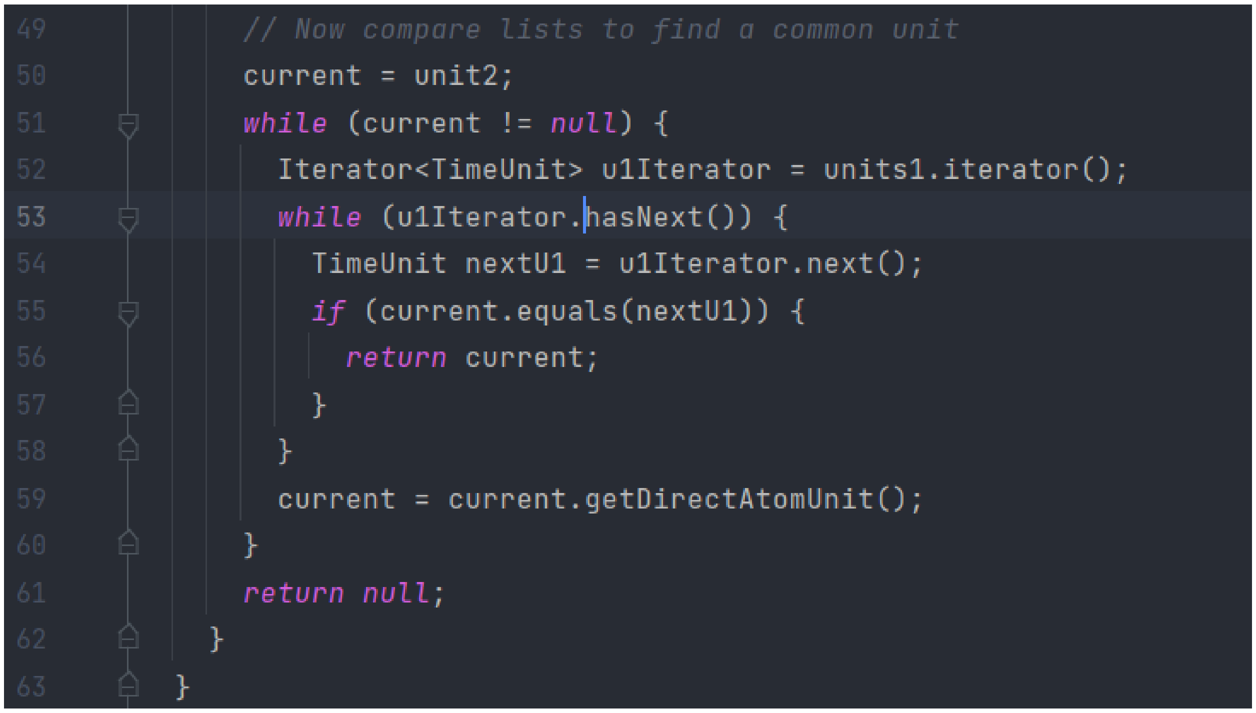


```
        public void addStyle(String style) {
          getStyles().add(style);
        }

        public boolean hasStyle(String style) {
          return getStyles().contains(style);
        }

```

3. <u>Iterator Pattern</u>
   Location:
   ganttproject\biz.ganttproject.core\src\main\java\biz\ganttproject\core\time\TimeUnitStack.java

   Explanation:
   The iterator is used to traverse a collection of elements and access them without exposing the underlying representation of the data structure.

   Reviewed by Filipe Santo.



```java
    // 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;
        }
      }
      current = current.getDirectAtomUnit();
    }
    return null;
  }
}
```