**Relatório da fase 1 do projeto de**

**Engenharia de Software**

****



**Índece**

[**Design Patterns**](#_p526qt3zqbj8) **3**

[Abstract Factory Pattern](#_lmrwyzsfzn7j) 3

[Builder Pattern](#_u5txp287or0m) 4

[Command Pattern](#_d86omf1n9aa5) 6

[Iterator Pattern](#_h7h6mg9ayugs) 8

[Memento Pattern](#_3gtv1x7lou98) 9

[Observer Pattern](#_k7zwqumqxvuc) 10

[Singleton](#_gh7l2wokiim0) 11

[Template method](#_1py9brx5ape) 13

[Facade pattern](#_pflhoeexchih) 14

[**Code Smells**](#_wv8lr19q80wl) **16**

[Data Class](#_5svp5tc8vns7) 16

[Data Clump](#_up0fq8z5nii4) 18

[Duplicated Code](#_bevu8wpm75km) 21

[Large class](#_chs2etagcvhp) 23

[Long Method](#_1zcfm1us487o) 24

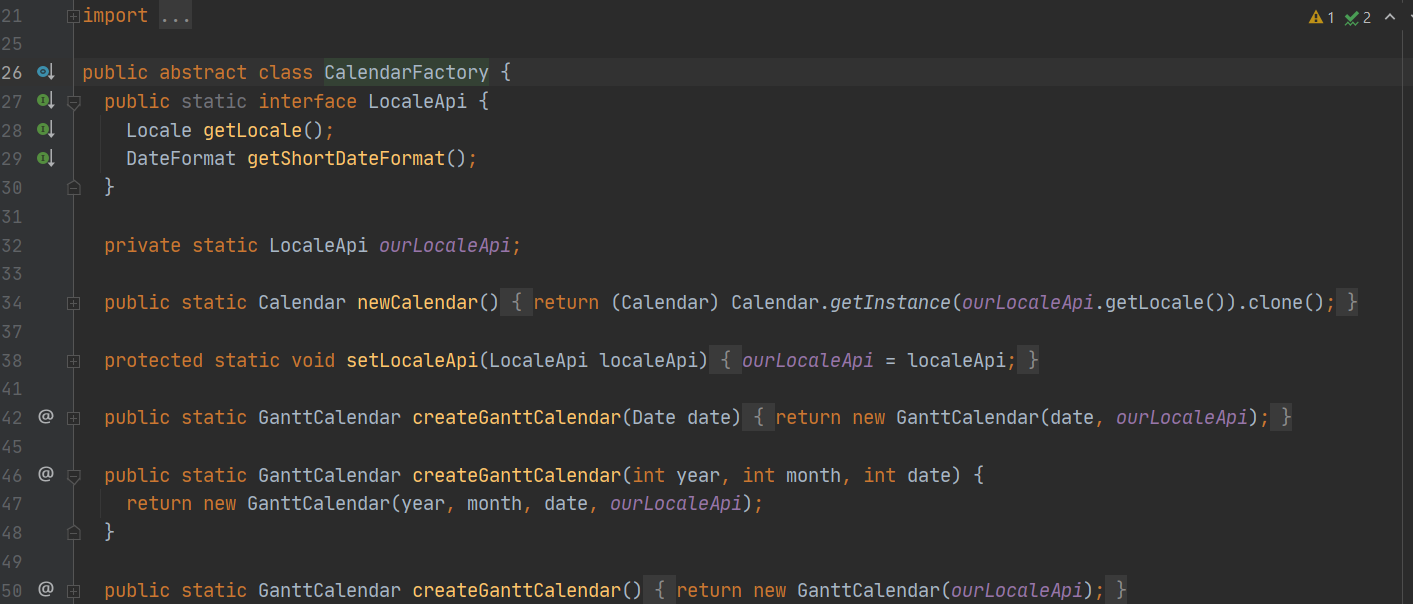
[No comment](#_vroq4bnkndic) 25

[Over comment](#_gj1eh47tcbva) 27

[Switch statement](#_8qypnnecr6wy) 28

# ***Design Patterns***

## **Abstract Factory Pattern**



**Ficheiro:**biz.ganttproject.core\src\main\java\biz\ganttproject\core\time\CalendarFactory

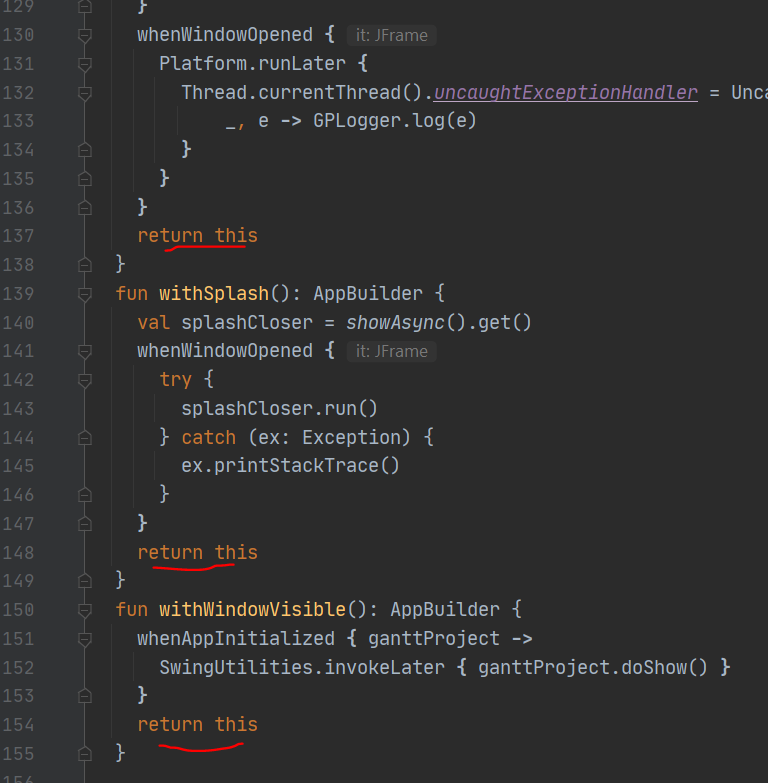
**Texto:** Este cria um Calendario de Gant diferente do tipo GanttCalendar dependendo dos parâmetros usados ou uma nova instância de calendário do tipo Calendar.

**Autor:** Iago Paulo

**Revisão:** Estamos perante um Factory Pattern. Pois o construtor foi alterado para as funcões createGanttCalendar que recebem alguns parâmetros e devolvem um produto **by** João Oliveira

## **Builder Pattern**



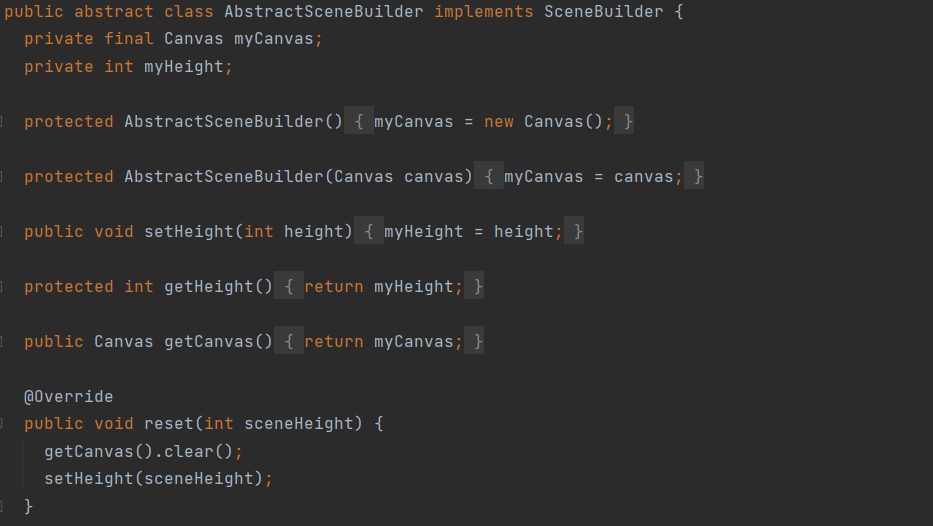
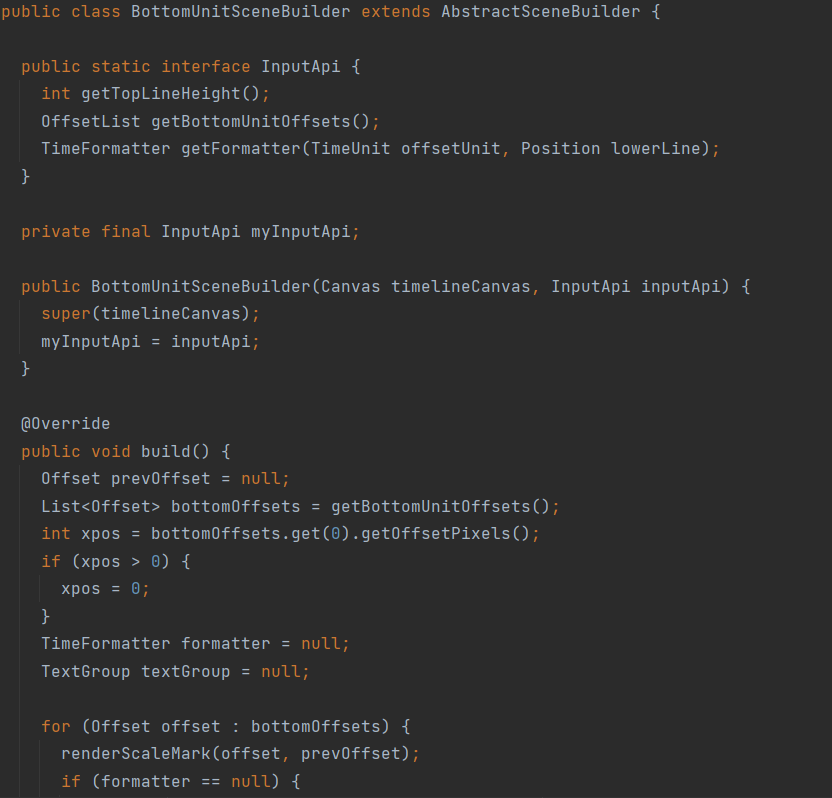


**Ficheiro:**ganttproject\ganttproject\src\main\java\net\sourceforge\ganttproject\App.kt

**Texto:** Este é um builder da aplicação no qual em cada método retorna o proprio builder para poder usar mais metodos em sequencia de adições para o App e finalmente tem um launch para lançar o programa.

**Autor:** Iago Paulo

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



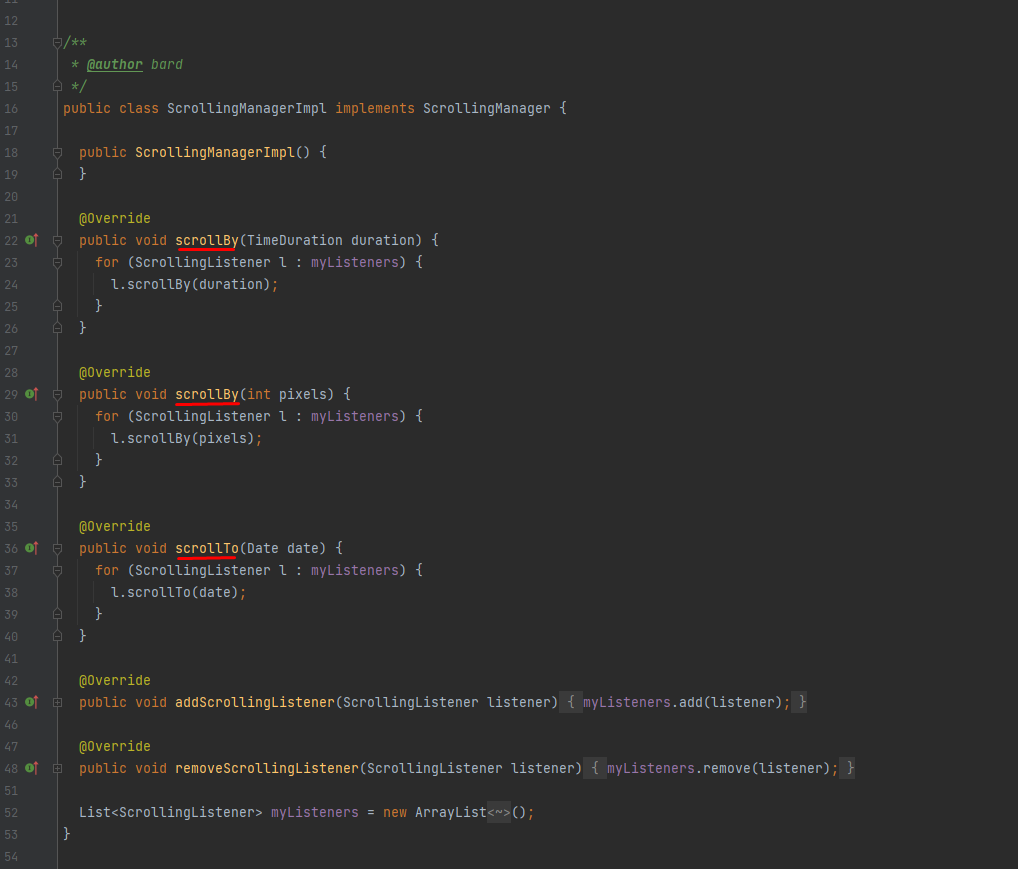
**Ficheiro:**biz.ganttproject.core/src/main/java/biz/ganttproject/core/chart/scene/AbstractSceneBuilder.java

**Texto:** Build pattern em AbstractSceneBuilder e SceneBuilder, que são base para coisas como ChartRendererBase, BottomUnitSceneBuilder, builders para Canvas

**Autor:** Ricardo Gonçalo

**Revisão:** Tem as características de de um builder pattern padrão e aparenta funcionar como tal **by** Iago Paulo

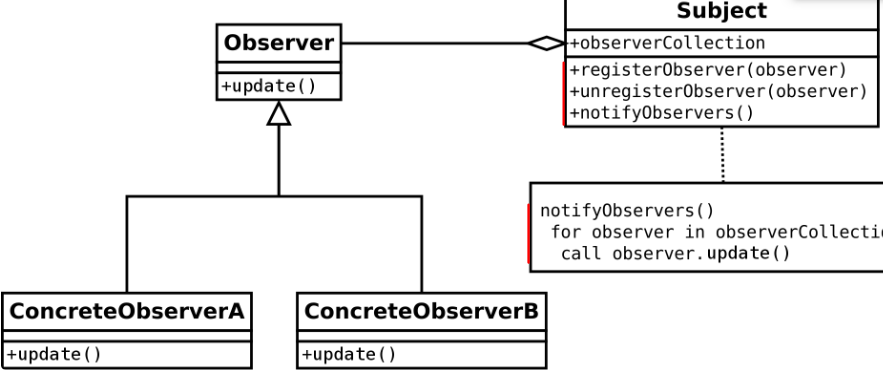
## **Command Pattern**

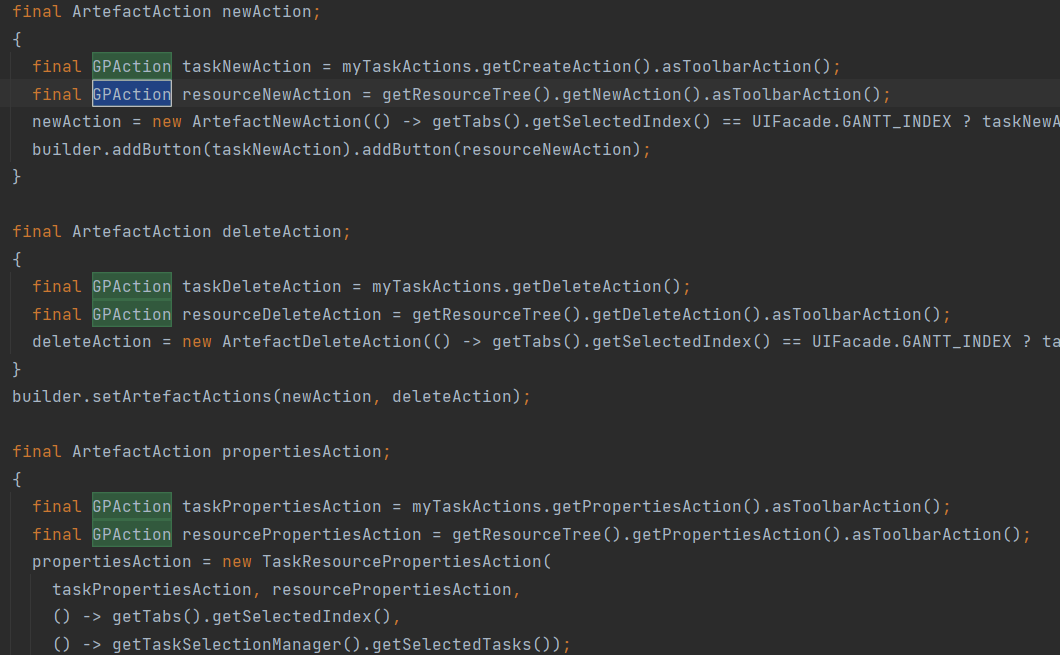
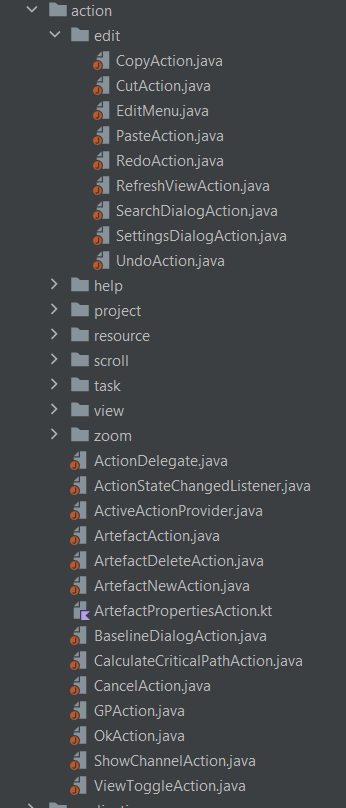


**Ficheiro:**ganttproject/src/main/java/net/sourceforge/ganttproject/gui/scrolling/ ScrollingManagerImpl.java

**Texto:** A classe ScrollingManagerImpl serve como manager dos comandos de "scrolling".

**Autor:** Francisco Vasco

**Revisão:** Francisco diz ser um Command Pattern apesar de que para mim estar mais parecido com um Observer Pattern devido às semelhanças dos métodos com o que vimos na aula prática sobre Observer Pattern:  **by** Iago Paulo



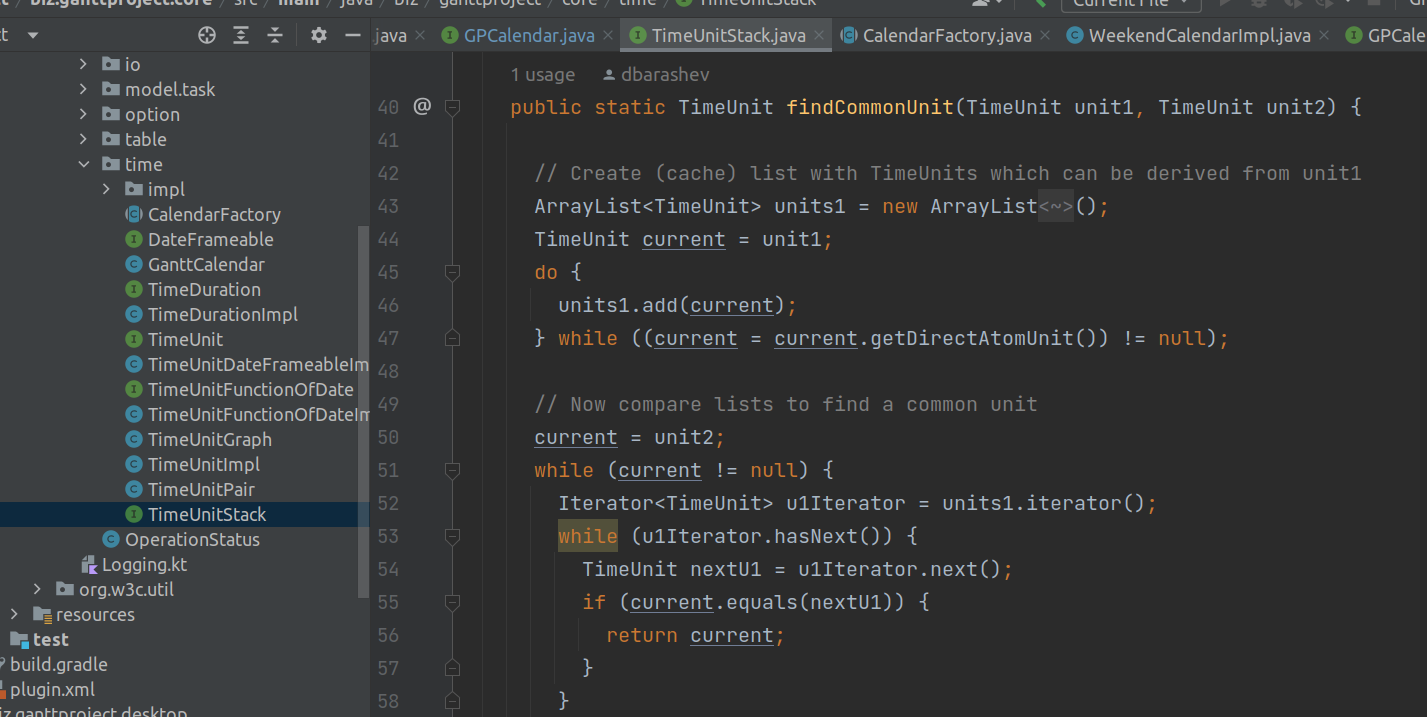
**Ficheiro:** *ganttproject\src\main\java\net\sourceforge\ganttproject\action*

**Texto:** Verifica-se um **Command** pattern, em que ações são classes em vez de apenas funções, verificado por exemplo na utilização destas ações em ganttproject/src/main/java/net/sourceforge/ganttproject/GanttProject.java

**Autor:** Ricardo Gonçalo

**Revisão:** As várias ações do programa, como o zoom, o scroll, etc, estão em classes separadas, encapsulando assim a informação necessária para executar cada uma das ações **by** João Oliveira

## **Iterator Pattern**



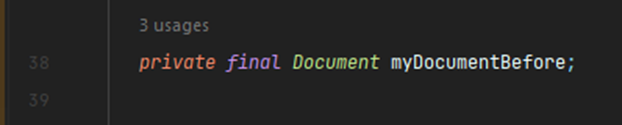
**Ficheiro:**biz.ganttproject.core/src/main/java/biz/ganttproject/core/time/TimeUnitStack.java

**Texto:** A classe Util que é um inner class da Interface TimeUnitStack usa o patter Iterator no método estático findCommonUnit.

**Autor:** James Furtado

**Revisão:** De facto a classe tem um iterador e está a usa-lo para acessar elementos, logo é um Pattern Iterator **by** Iago Paulo

## **Memento Pattern**



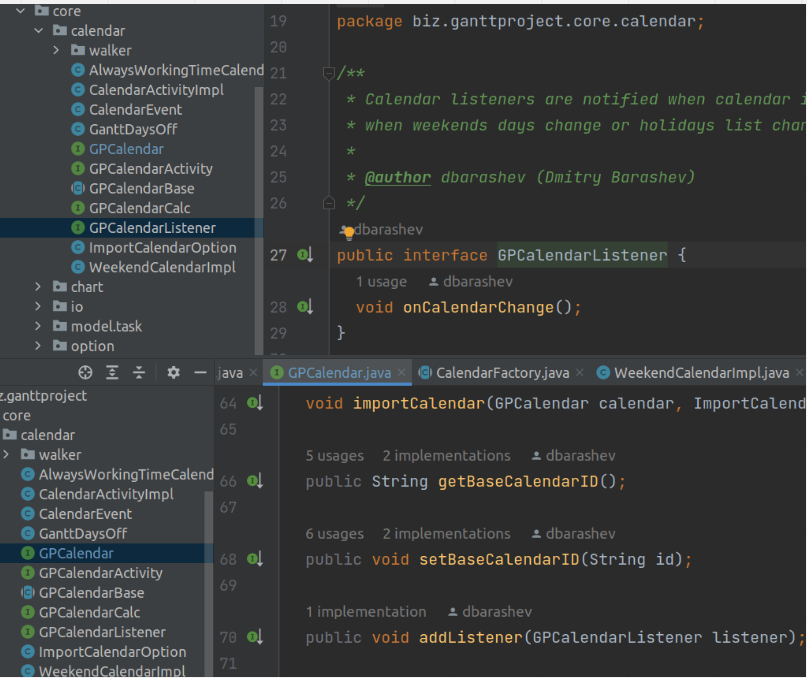
**Ficheiro:**ganttproject/src/main/java/net/sourceforge/ganttproject/undo/UndoableEditImpl.java

**Texto:** Guarda um objeto com o estado anterior do documento, tendo um método undo() que restaura o documento ao seu estado anterior.

**Autor:** João Oliveira

**Revisão:** Correto, é possível restaurar o estado anterior do documento a partir do método undo() ao manter uma snapshot do estado anterior **by** Ricardo Gonçalo

## **Observer Pattern**



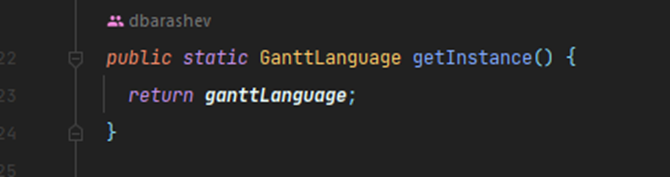
**Ficheiro:**biz.ganttproject.core/src/main/java/biz/ganttproject/core/calendar/

**Texto:** No package biz.ganttproject.core.calendar há um observer pattern onde a interface do Subject é o GPCalendar e a interface do Observer é GPCalenderListener.

**Autor:** James Furtado

**Revisão:** Implementações de GPCalendar (o Subject) mantêm uma coleção de GPCalendarListener (adicionados através do método addListener) e notificam cada um ao efetuar mudanças ao calendário, o padrão parece-me correto **by** Francisco Vasco

## **Singleton**

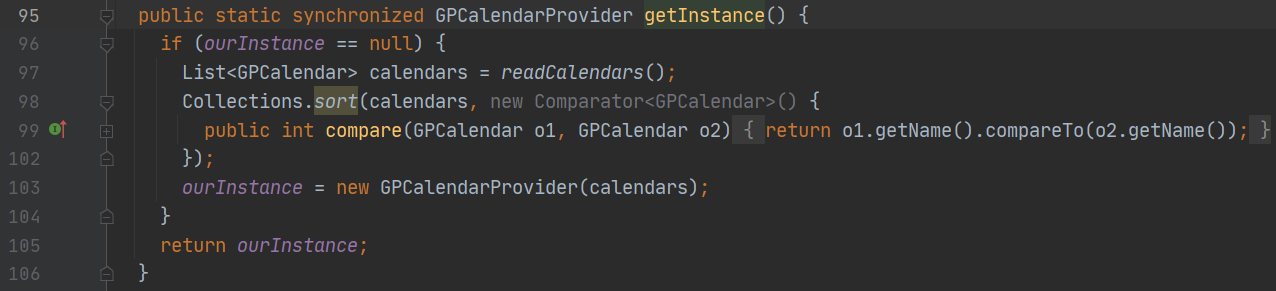


**Ficheiro:**ganttproject/src/main/java/net/sourceforge/ganttproject/parser/AbstractTagHandler.javaganttproject/src/main/java/net/sourceforge/ganttproject/language/GanttLanguage.java

**Texto:** Uso da função getInstance() que retorna uma única instância do objeto, que é inicializado uma única vez, através do uso de uma variável estática (linha 79)

**Autor:** João Oliveira

**Revisão:** Correto, implementação normal com Singleton Pattern **by** Francisco Vasco

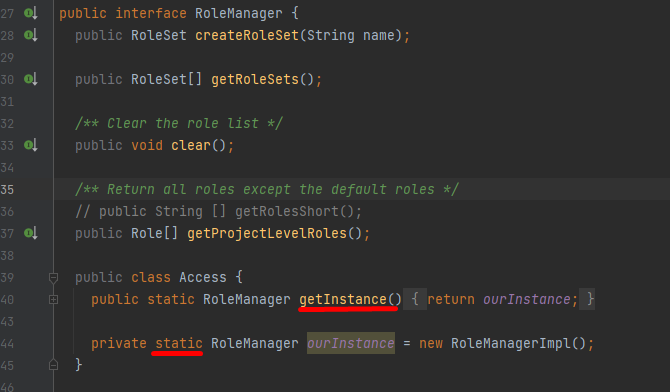


**Ficheiro:**ganttproject\ganttproject\src\main\java\net\sourceforge\ganttproject\calendar\GPCalendarProvider.java

**Texto:** Utilização do método GetInstance implementado da maneira esperada para um pattern Singleton

**Autor:** Iago Paulo

**Revisão:** De facto este método em conjunto com a variável ourInstance assegura que apenas existe uma instância de GPCalendarProvider.  **by** Francisco Vasco

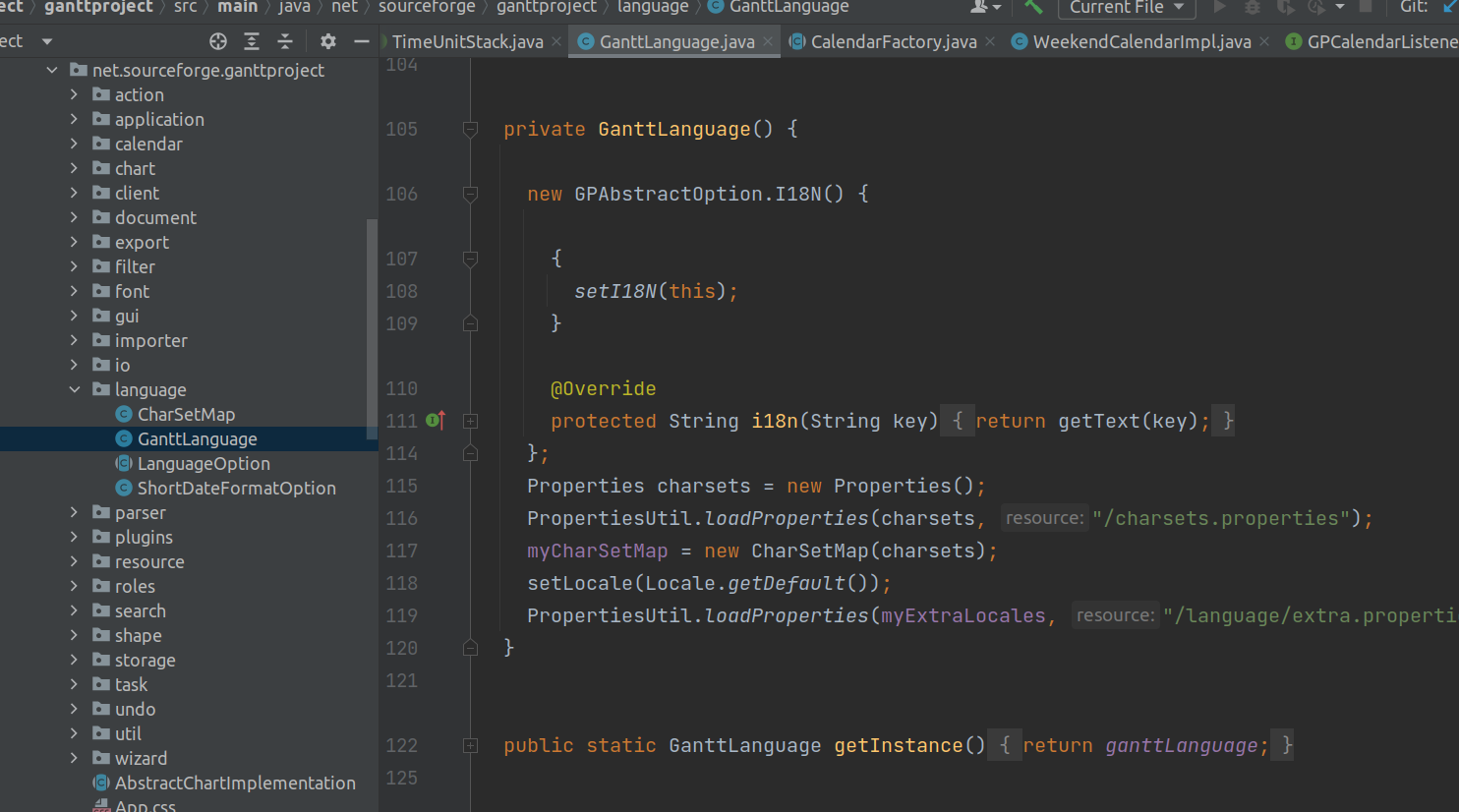


**Ficheiro:**ganttproject/src/main/java/net/sourceforge/ganttproject/roles/RoleManager.java

**Texto:** A classe Access na linha 39-45 assegura a instanciação de apenas um RoleManager

**Autor:** Francisco Vasco

**Revisão:** A inner class Access da class RoleManager usa mesmo o pattern singleton **by** James Furtado



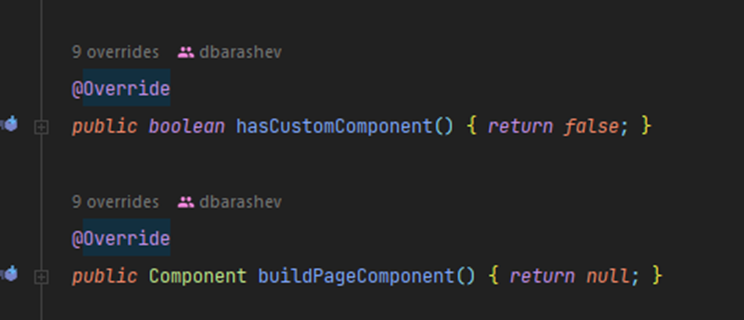
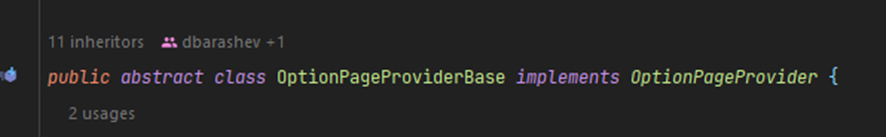
**Ficheiro:**ganttproject/src/main/java/net/sourceforge/ganttproject/language/GanttLanguage.java

**Texto:** A classe GanttLanguage tem um construtor privado e um método estático getIntance() que devolve a única instância da classe.

**Autor:** James Furtado

**Revisão:** Sim, não haveria melhor maneira de dizer que é singleton **by** Ricardo Gonçalo

## **Template method**



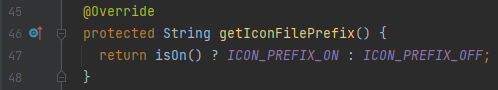
**Ficheiro:**ganttproject/src/main/java/net/sourceforge/ganttproject/gui/options/OptionPageProviderBase.java

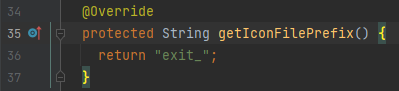
**Texto:** A classe abstrata que é estendida por 11 classes. 9 das classes reescrevem os metodos hasCustomComponent() e buildPageComponent()

**Autor:** João Oliveira

**Revisão:** Sim de facto isto é um template método. A classe é abstrata e as subclasses rescrevem esses e outros métodos **by** James Furtado







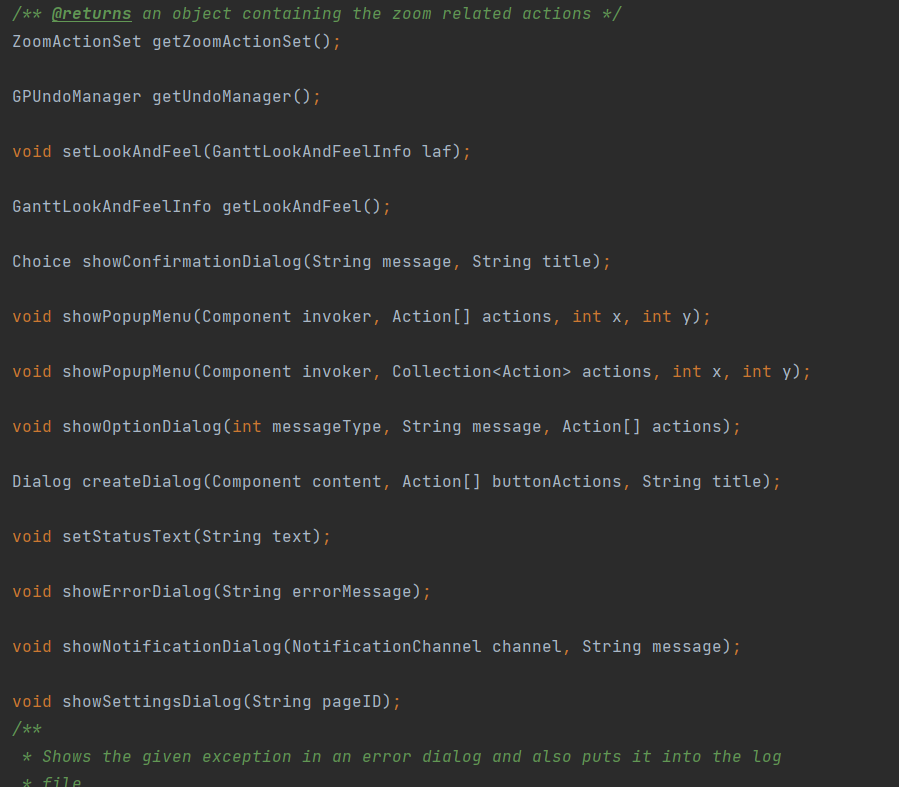
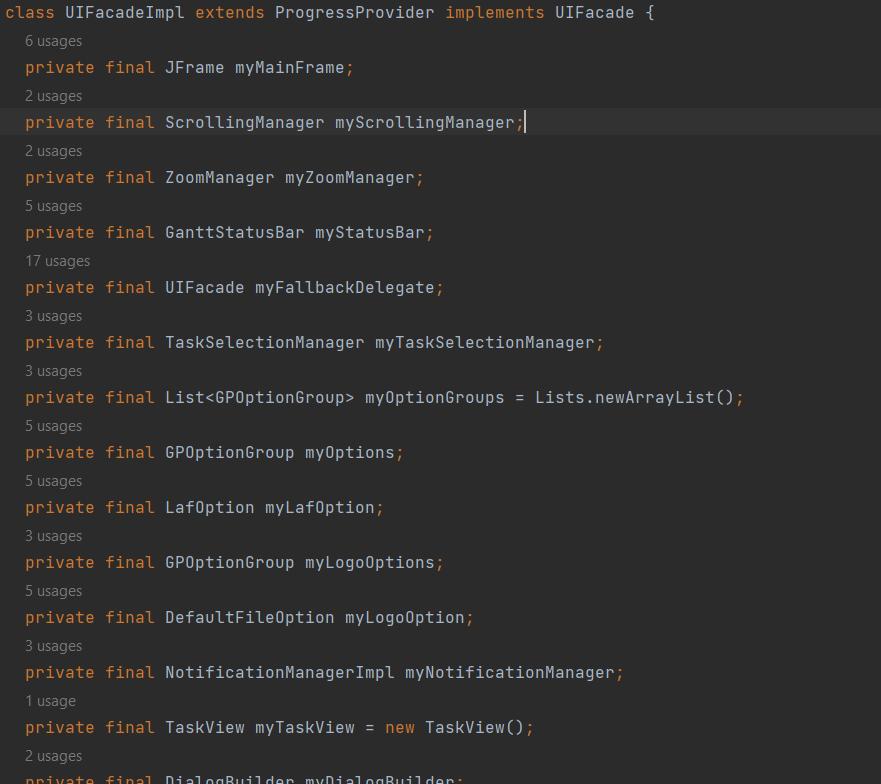
**Ficheiro:**ganttproject/src/main/java/net/sourceforge/ganttproject/action/GPAction.java (linhas 138-151 e 193-195)

**Texto:** O método final createIcon(String iconSize) na classe abstrata GPAction chama o método getIconFilePrefix() cuja implementação concreta varia depois nas classes que estendem GPAction

**Autor:** Francisco Vasco

**Revisão:** Estamos perante um padrão de comportamento onde na classe está o método getIconFilePrefix() que é depois alterado por 19 classes que estendem a classe. **by** João Oliveira

## **Facade pattern**



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

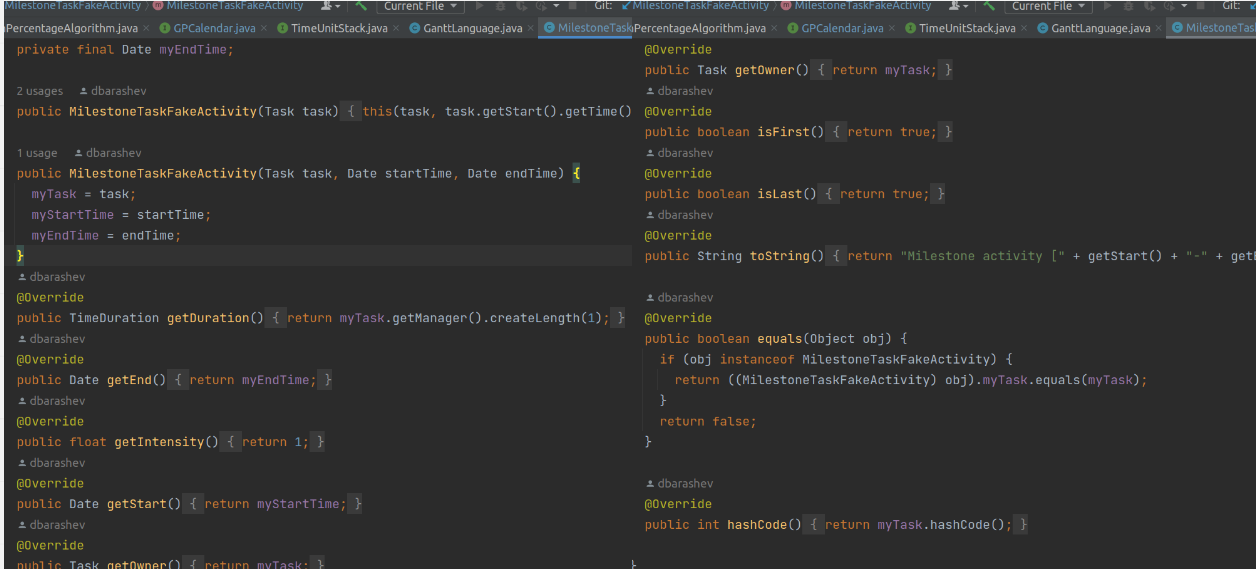
**Texto:** Representam um façade template, que tenta esconder toda a complexidade do UI

**Autor:** Ricardo Gonçalo

**Revisão:** O nome da classe é bem descritivo, tudo o que posso dizer é que concordo **by** James Furtado

# ***Code Smells***

## **Data Class**



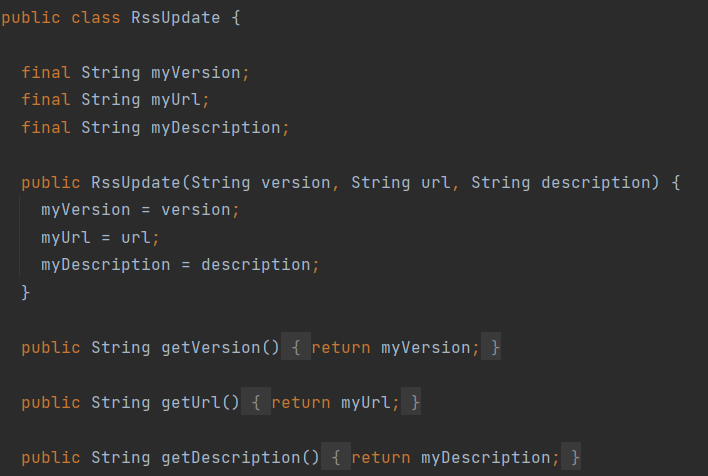
**Ficheiro:**ganttproject/src/main/java/net/sourceforge/ganttproject/chart/MilestoneTaskFakeActivity.java

**Texto:** A class MilestoneTaskFakeActivity só tem getters.

**Solução:** Tentar dar alguma responsabilidade a classe.

**Autor:** James Furtado

**Revisão:** Confirma-se que a classe só tem “getters”. A solução sugerida parece-me aceitável, outra solução seria guardar estas informações nas classes que as utilizam **by** Francisco Vasco



**Ficheiro:** ganttproject/src/main/java/net/sourceforge/ganttproject/client/RssUpdate.java

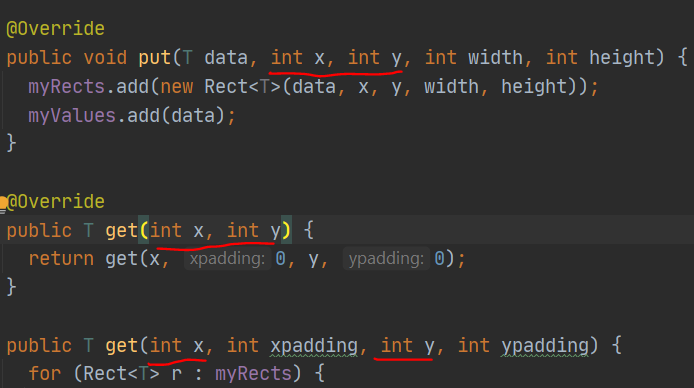
**Texto:** Isto é uma Data Class, totalmente definida pelas 3 primitivas que tem, a sua existência isolada acrescenta complexidade desnecessária

**Solução:** Esta class podia muito bem estar nested na RssParser, já que é apenas aí que é usada

**Autor:** Ricardo Gonçalo

**Revisão:** Seria uma soluçãoviável tendo em conta a reduçãode complexidade **by** Iago Paulo

## **Data Clump**



**Ficheiro:**ganttproject-master\biz.ganttproject.core\src\main\java\biz\ganttproject\core\chart\canvas\DummySpatialIndex.java

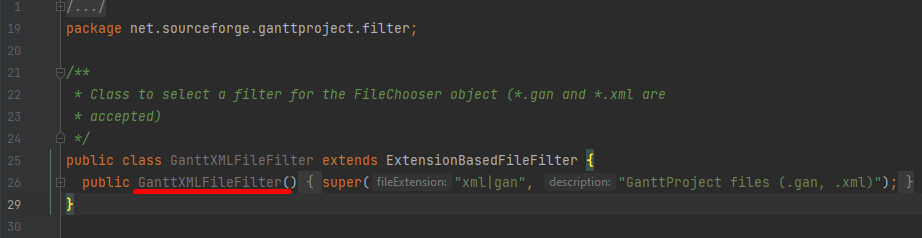
**Texto:**Lista de parâmetros muito longa.

**Solução:**Criar uma classe para agrupar alguns dos argumentos por exemplo o x e o y.

**Autor:** Iago Paulo

**Revisão:** É de facto uma data clump **by** João Oliveira

**DeadCode**



**Ficheiro:**ganttproject/src/main/java/net/sourceforge/ganttproject/filter/GanttXMLFileFilter.java

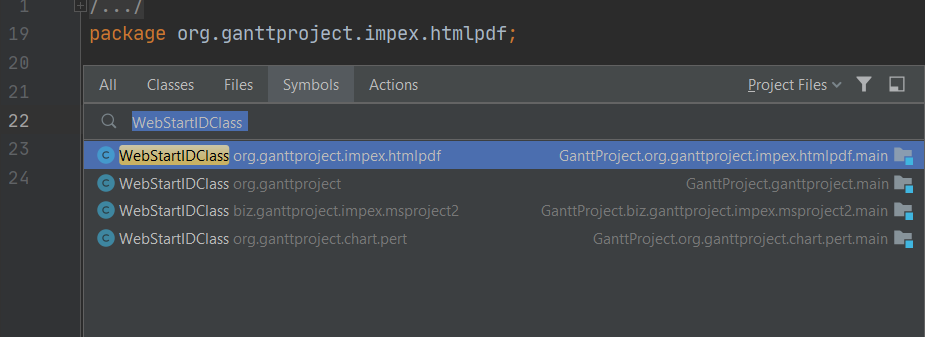
**Texto:** A classe GanttXMLFileFilter nunca é usada

**Solução:** Apagar a classe já que nunca é utilizada.

**Autor:** Francisco Vasco

**Revisão:** A classe GanttXMLFileFilter não está a ser usada pelo que pode ser apagada em segurança sem afetar o programa **by** João Oliveira





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

**Texto:** A classe WebStartIDClass vazia (todas as classes com este nome também estão), talvez também se possa considerar Specular Generality dependendo do contexto.

**Solução:** Apagar as classes… Ou fazer uso delas

**Autor:** Iago Paulo

**Revisão:** Concordo que é DeadCode, não é utilizado e que pode ser também uma situação de preparação/”over-engineering” em antecipação de funcionalidades futuras **by** Francisco Basco



**Ficheiro:**ganttproject/src/main/java/net/sourceforge/ganttproject/action/ArtefactAction.java

**Texto:** A variável myDelegates (private final Action[] myDelegates) é inicializada, mas nunca usada.

**Solução:** Pode ser apagada sem causar nenhum problema.

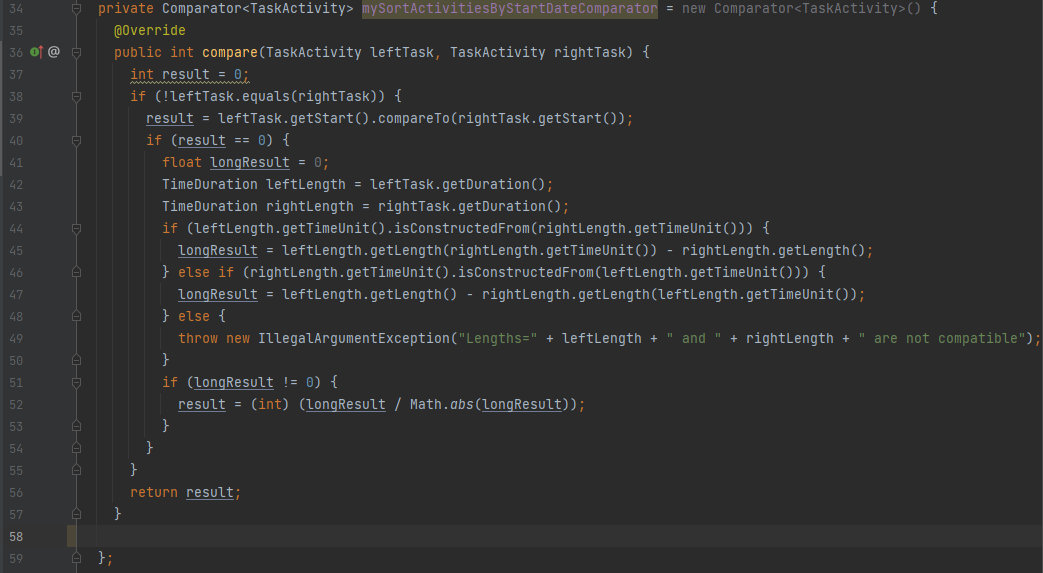
**Autor:** João Oliveira

**Revisão:** Sim realmente é verdade e como é privada não é acessível de fora, a única coisa que poe estar a fazer é a impedir objetos de serem apanhados pelo garbage collector **by** Ricardo Gonçalo

## 

## 

## **Duplicated Code**



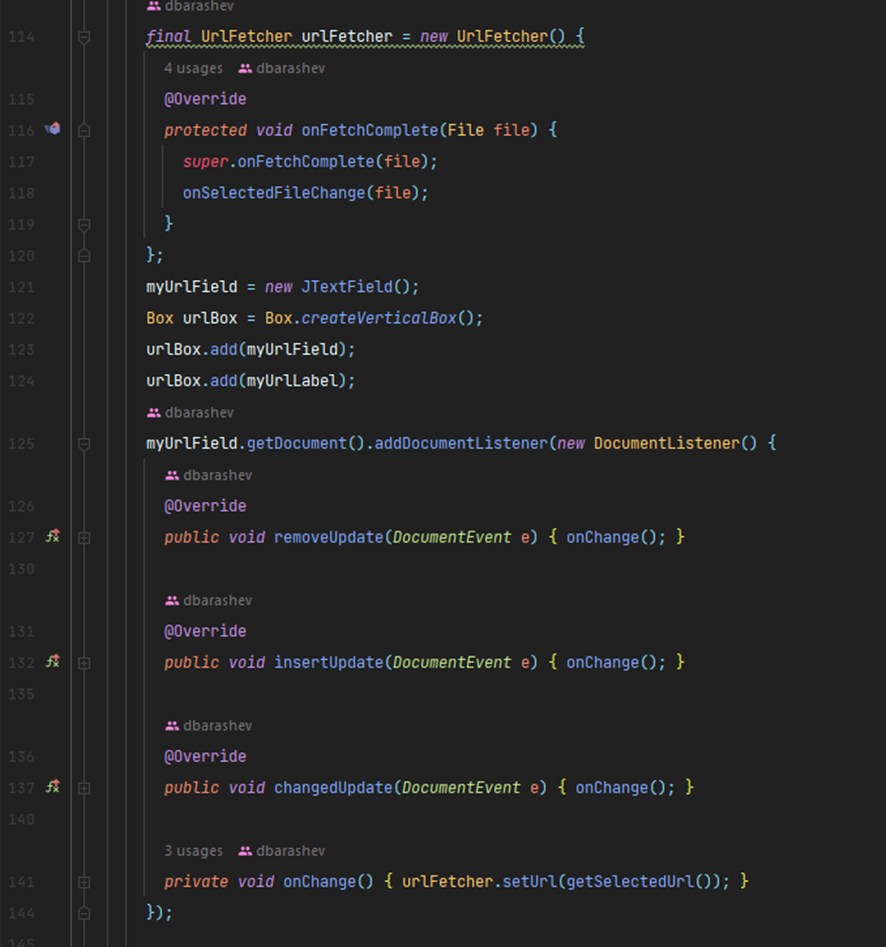
**Ficheiro:**ganttproject/src/main/java/net/sourceforge/ganttproject/task/algorithm/SortTasksAlgorithm.java

**Texto:** Na classe SortTasksAlgorithm o código nas linhas 44-53 é idêntico ao código das linhas 71-80

**Solução:** Extrair parte do código duplicado para um método auxiliar.

**Autor:** Francisco Vasco

**Revisão:** Estes 2 pedaços de código são de facto exatamente iguais, um método auxiliar seria uma solução viável **by** Iago Paulo



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

**Texto:** Código na linha 114 da primeira classe está duplicado em ambas as classes.

**Solução:** O código podia ser extraído para um método ou para uma classe auxiliar

**Autor:** João Oliveira

**Revisão:** Parece-me correta a identificação. Embora se trate de apenas uma linha podem surgir erros após um futuro refactoring que se evitam com esta solução. **by** Francisco Vasco

## **Large class**



**Ficheiro:**ganttproject/src/main/java/net/sourceforge/ganttproject/GPTreeTableBase.java

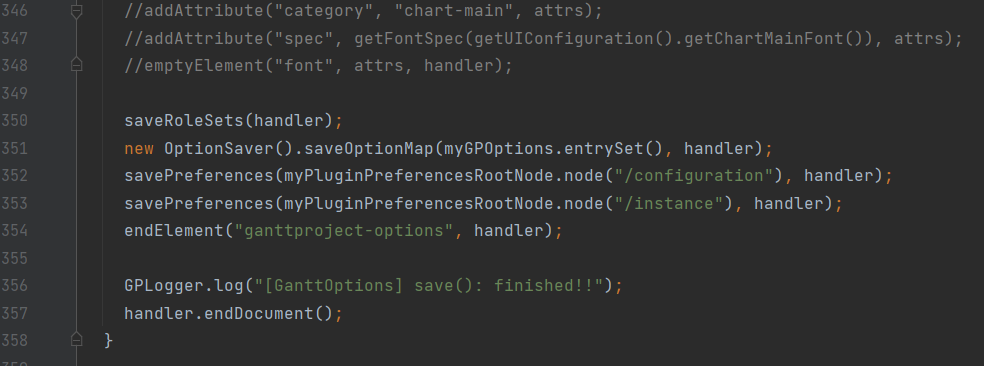
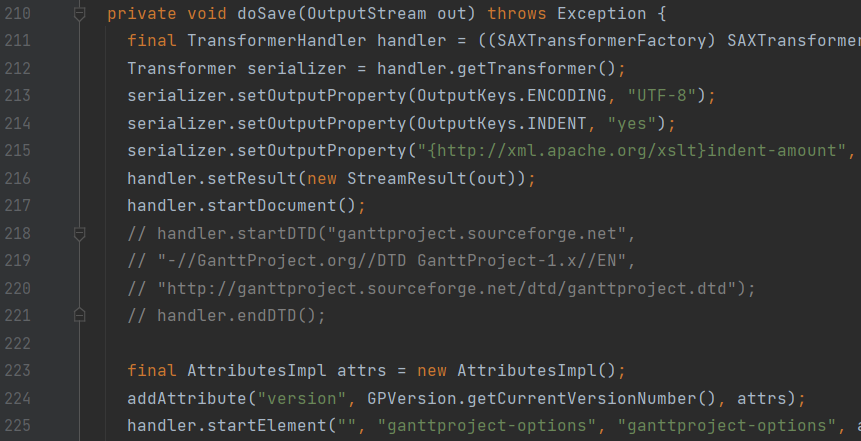
**Texto:** A class GPTreeTableBase tem mais de 1000 linha de código além de ter muitas outras classes dentro dela mesma e métodos muito longos.

**Solução:** Dividir a class em outras classes mais pequenas.

**Autor:** James Furtado

**Revisão:** Classe tem mais de 1200 linhas e é composto por 6 classes dentro do ficheiro, de facto é uma classe longa. **by** Iago Paulo

## **Long Method**



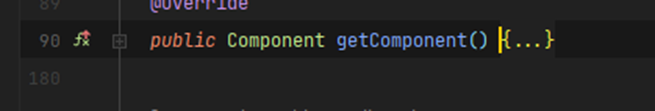
**Ficheiro:** ganttproject/src/main/java/net/sourceforge/ganttproject/GanttOptions.java

**Texto:** Em no método doSave, linhas .210 - 358, tem-se método demasiado grande e código comentado

**Solução:** Simplificar as tarefas da função em subtarefas, para se tornar de mais fácil leitura e compreensão

**Autor:** Ricardo Gonçalo

**Revisão:** O método está de facto demasiado grande, apesar de ter todo o objetivo de inicializar a UI. Podiam ter sido usadas funções auxiliares para tornar mais claro o que é feito no método. Há também pedaços de código comentados que deviam ser apagados. **by** João Oliveira



**Ficheiro:**ganttproject/src/main/java/net/sourceforge/ganttproject/gui/FileChooserPageBase.java

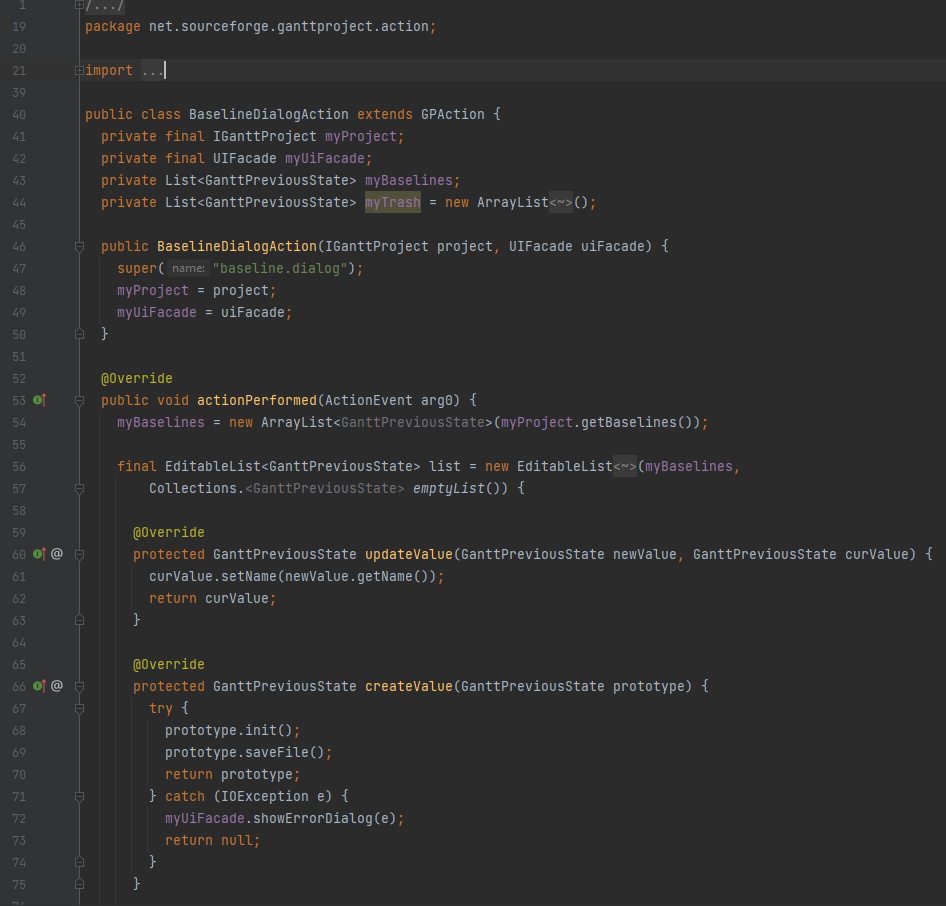
**Texto:** O método getComponent() é muito longo (90 linhas).

**Solução:** Devia ser criado métodos auxiliares.

**Autor:** João Oliveira

**Revisão:** Também acho que isto pode ser considerado um code smell **by** James Hertz

## **No comment**



(a classe é grande por isso apresenta-se apenas uma parte aqui)

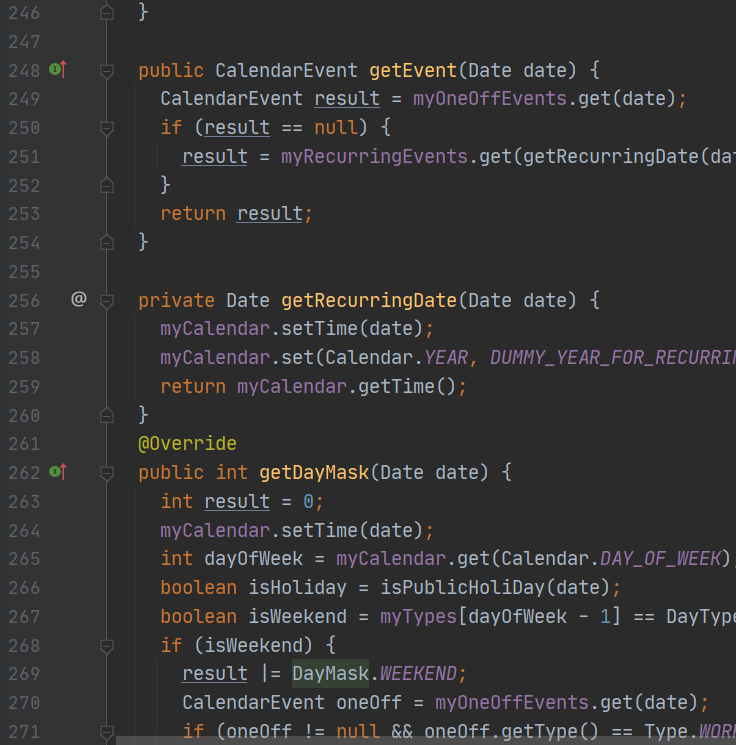
**Ficheiro:**ganttproject/src/main/java/net/sourceforge/ganttproject/action/BaselineDialogAction.java

**Texto:** A classe BaselineDialogAction sem comentários apesar da sua complexidade.

**Solução:**

**Autor:** Francisco Vasco

**Revisão:** A classe não tem nenhum comentário. Então eu concordo que isso é um code smell **by** James Hertz



**Ficheiro:**biz.ganttproject.core/src/main/java/biz/ganttproject/core/calendar/WeekendCalendarImpl.java

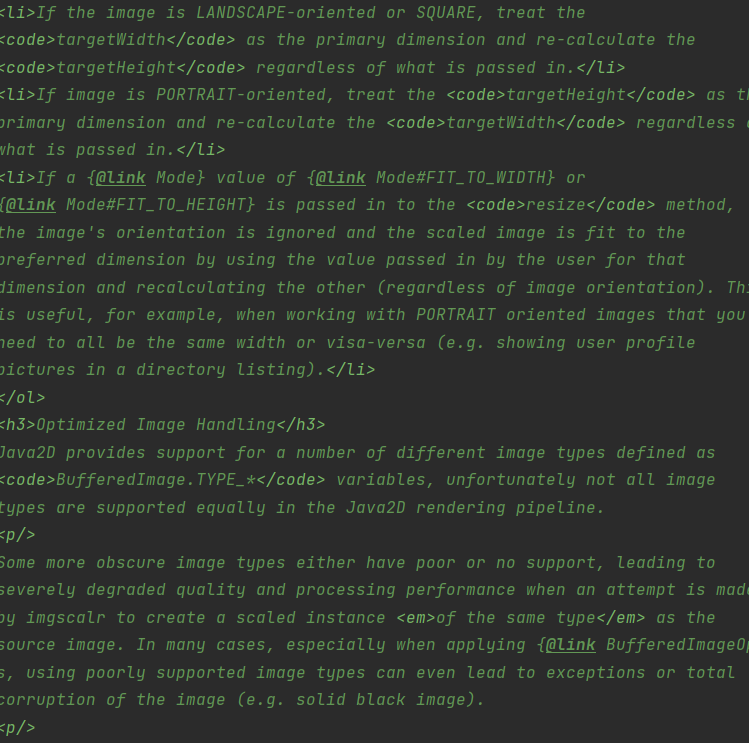
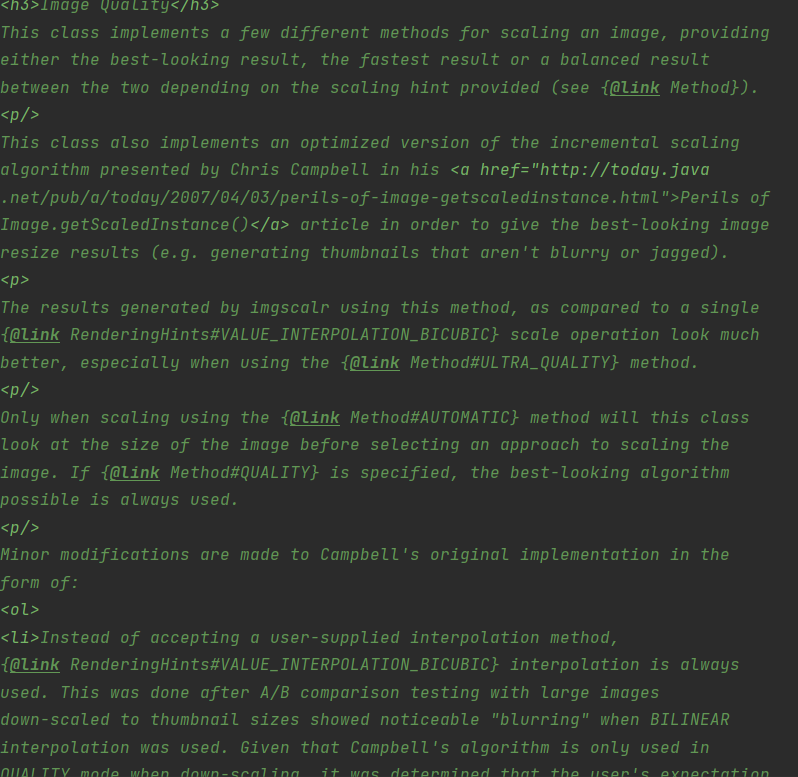
**Texto:** Essa classe não tem comentários.

**Solução:** Meter comentários

**Autor:** Iago Paulo

**Revisão:** Sim, para uma class de 380 linhas tem muitos poucos comentários **by** Ricardo Gonçalo

## **Over comment**



**Ficheiro:** ganttproject/src/main/java/org/imgscalr/Scalr.java

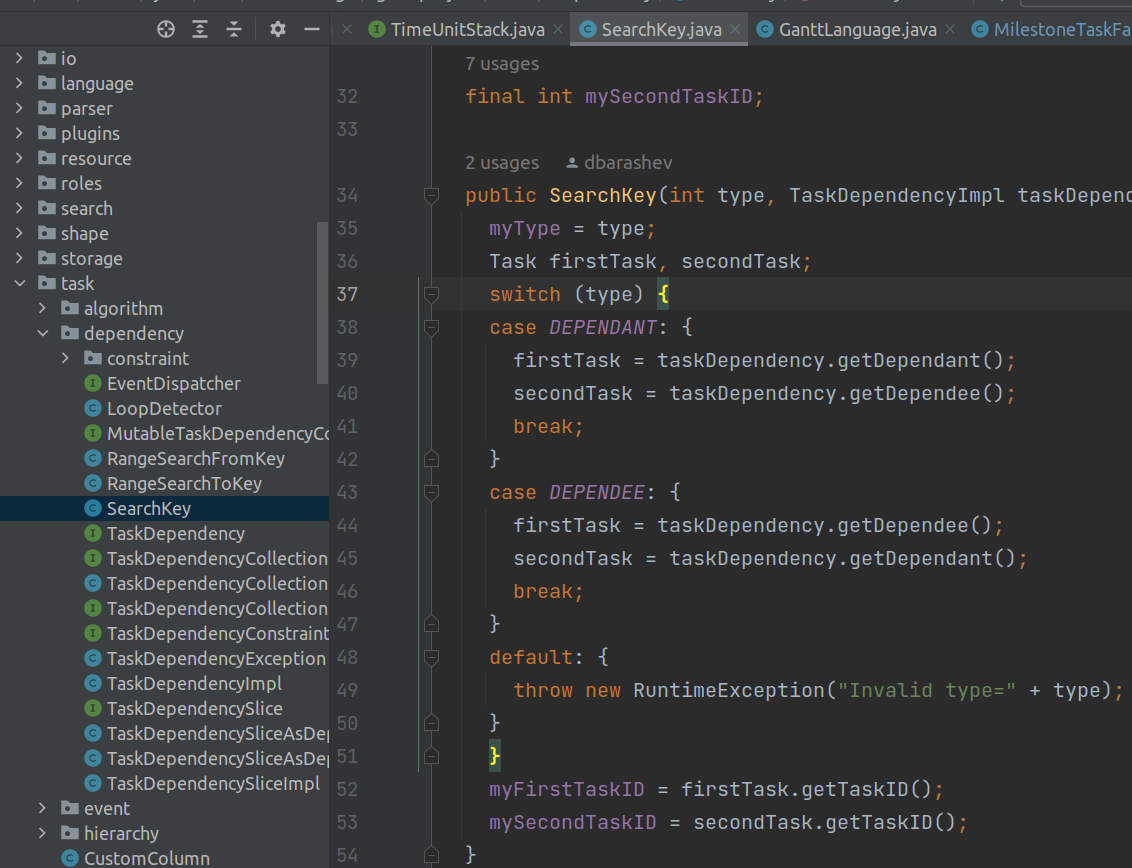
**Texto:** Tem comentários muito grandes. Por exemplo o que está logo em cima da classe tem mais de 130 linhas.

**Solução:** Simplificar estes comentários, há mesmo necessidade de explicações tão extensivas para o funcionamento desta classe? Claro que não, muita informação desnecessária

**Autor:** Ricardo Gonçalo

**Revisão:** Ao contrário do restante do código (que quase não tem comentários) essa classe tem comentários e são muitos, assim sendo, claramente é um code smell **by** James Furtado

## **Switch statement**



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

**Texto:** A classe tem um atributo myType que é um inteiro. As constantes DEPENDEE e DEPENDANT são usadas como tipos. E tem um switch statement no construtor para os dois casos e uma exceção caso contrário.

**Solução:** Fazer duas subclasses. Além de eliminar o atributo myType vai eliminar a exceção o que já é muito bom.

**Autor:** James Furtado

**Revisão:** Sim, no mínimo substituir por um Enum **by** Ricardo Gonçalo