-> A duplicação de código de maneira a preencher todos os dias de um calendário retrata uma má prática de código.

Com o objetivo de eliminar a repetição de código, é possível recorrer a um ciclo {for each} de maneira a percorrer todos os valores pertencentes ao Enum {Day},

chamando desta forma a função {calendar.setWorkingDay(…)} uma vez apenas.

Exemplo:

private void exportWeekends(ProjectCalendar calendar) {

ProjectCalendarHours workingDayHours = calendar.getCalendarHours(Day.MONDAY);calendar.setWorkingDay(Day.MONDAY, isWorkingDay(Calendar.MONDAY));

calendar.setWorkingDay(Day.TUESDAY, isWorkingDay(Calendar.TUESDAY));

calendar.setWorkingDay(Day.WEDNESDAY, isWorkingDay(Calendar.WEDNESDAY));

calendar.setWorkingDay(Day.THURSDAY, isWorkingDay(Calendar.THURSDAY));

calendar.setWorkingDay(Day.FRIDAY, isWorkingDay(Calendar.FRIDAY));

calendar.setWorkingDay(Day.SATURDAY, isWorkingDay(Calendar.SATURDAY));

if (calendar.isWorkingDay(Day.SATURDAY)) {

copyHours(workingDayHours, calendar.addCalendarHours(Day.SATURDAY));

}

calendar.setWorkingDay(Day.SUNDAY, isWorkingDay(Calendar.SUNDAY));

if (calendar.isWorkingDay(Day.SUNDAY)) {

copyHours(workingDayHours, calendar.addCalendarHours(Day.SUNDAY));

}

Localização: ganttproject/biz.ganttproject.impex.msproject2/src/main/java/biz/ganttproject/impex/msproject2/ProjectFileExporter.java

Linhas 110-124

------------------------------------------------------------------------------------------------------------------------------

-> Classe com multiplos métodos definida dentro de uma interface;

Na Iterface ColumnList podemos encontrar definidas mais uma interface e uma classe com multiplos métodos. Isto torna o código bastante confuso e dificil de ler.

Sendo assim, seria preferível separar esta classe e interface extra.

Exemplo:

public interface Column {...}

class ColumnStub implements ColumnList.Column {...}

Localização: ganttproject.biz.ganttproject.core.src.main.java.biz.ganttproject.core.table.ColumnList.java

Linhas 25 e 60, respetivamente.

------------------------------------------------------------------------------------------------------------------------------

-> A utilização de comentários para informar os outros programadores que um método se encontra incompleto ou com problemas é também um code smell.

A retirada deste pedaço de código ou o término deste permitirá que os outros programadores olhem para o código de maneira menos confusa,

sendo também mais acessível para estes darem o seu contributo.

Exemplo:

private static Duration convertLag(TaskDependency dep) {

// TODO(dbarashev): Get rid of days

return Duration.getInstance(dep.getDifference(), TimeUnit.DAYS);

}

Localização: ganttproject/biz.ganttproject.impex.msproject2/src/main/java/biz/ganttproject/impex/msproject2/ProjectFileExporter.java

Linhas 303-306