

# Preuves sur l'accomplissement des tâches.

Je maîtrise les règles de nommage Java.

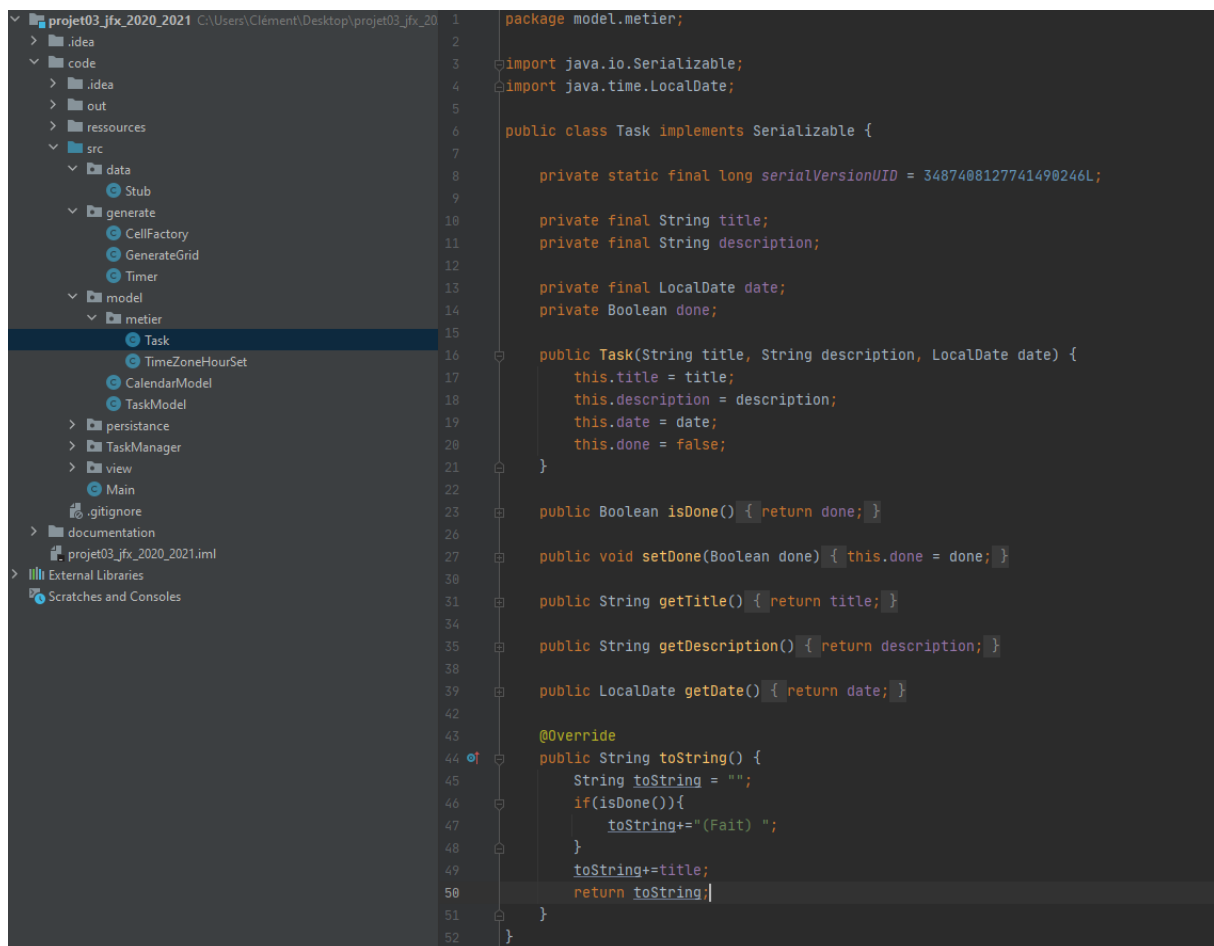
Je sais coder une classe Java en respectant des contraintes de qualité de lecture de code.

Je sais éviter la duplication de code.

Je sais maintenir, dans un projet, une responsabilité unique pour chacune de mes classes.

Je sais utiliser le type statique adéquat pour mes attributs ou variables.

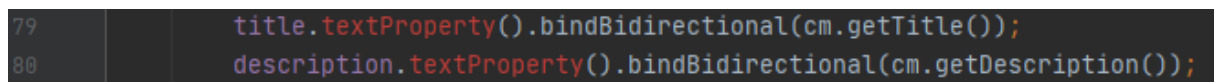
Je sais utiliser les collections.



The screenshot shows an IDE with a project structure on the left and a Java class file on the right. The project structure includes folders like 'code', 'out', 'ressources', 'src', 'data', 'generate', 'model', and 'metier'. The 'Task' class is selected in the 'metier' folder. The code for the 'Task' class is displayed on the right, showing imports for 'Serializable' and 'LocalDate', and the implementation of the class with attributes, constructors, getters, setters, and a toString method.

```
1 package model.metier;
2
3 import java.io.Serializable;
4 import java.time.LocalDate;
5
6 public class Task implements Serializable {
7
8     private static final long serialVersionUID = 3487408127741490246L;
9
10    private final String title;
11    private final String description;
12
13    private final LocalDate date;
14    private Boolean done;
15
16    public Task(String title, String description, LocalDate date) {
17        this.title = title;
18        this.description = description;
19        this.date = date;
20        this.done = false;
21    }
22
23    public Boolean isDone() { return done; }
24
25    public void setDone(Boolean done) { this.done = done; }
26
27    public String getTitle() { return title; }
28
29    public String getDescription() { return description; }
30
31    public LocalDate getDate() { return date; }
32
33    @Override
34    public String toString() {
35        String toString = "";
36        if(isDone()){
37            toString+="(Fait) ";
38        }
39        toString+=title;
40        return toString;
41    }
42
43 }
```

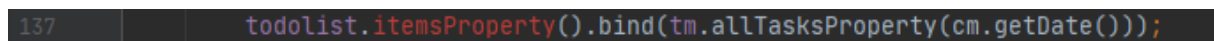
Je sais binder bidirectionnellement deux propriétés JavaFX. (Cf : Calendar.java) (title et description sont des TextField/TextArea)



The screenshot shows two lines of Java code. Line 79 binds the 'title' text property of a JavaFX control to the 'getTitle()' method of a 'cm' object. Line 80 binds the 'description' text property to the 'getDescription()' method of the same 'cm' object.

```
79 title.textProperty().bindBidirectional(cm.getTitle());
80 description.textProperty().bindBidirectional(cm.getDescription());
```

Je sais binder unidirectionnellement deux propriétés JavaFX. (Cf : MainWindow.java) (On bind ici la ListProperty<Task> qui correspond à la date demandée dans une ListView)



The screenshot shows a single line of Java code. Line 137 binds the 'itemsProperty()' of a 'todolist' JavaFX control to the 'allTasksProperty()' of a 'tm' object, which is passed the 'getDate()' method of a 'cm' object.

```
137 todolist.itemsProperty().bind(tm.allTasksProperty(cm.getDate()));
```

Je sais contraindre les éléments de ma vue, avec du binding FXML.

```
<Label fx:id="hiddenLabel" text="{yearCb.id}" textFill="WHITE" BorderPane.alignment="CENTER" GridPane.columnIndex="1">
```

Je sais définir une CellFactory fabriquant des cellules qui se mettent à jour au changement du modèle. (Cf : CellFactory.java)

```
public static void setTableView(TableView tv){
    setTableAppearance(tv);

    ObservableList<TimeZoneHourSet> timeZoneHourSets = Timer.getTimerProperty();

    TableColumn<String, String> timeZone = new TableColumn("TimeZone");
    TableColumn<SimpleStringProperty, String> hourProperty = new TableColumn("Heure");

    tv.getColumns().add(timeZone);
    tv.getColumns().add(hourProperty);

    timeZone.setCellValueFactory(new PropertyValueFactory<>("timeZone"));
    hourProperty.setCellValueFactory(new PropertyValueFactory<>("hourProperty"));

    tv.setItems(timeZoneHourSets);
}
```

Je sais développer une application graphique en JavaFX en utilisant FXML. (Cf : Calendar.fxml)

Je sais utiliser certains composants simples que me propose JavaFX.

Je sais utiliser certains layout que me propose JavaFX.

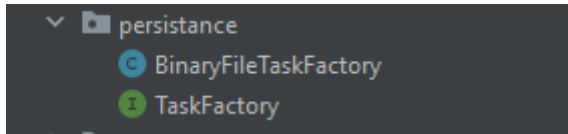
Je sais utiliser les différents composants complexes (listes, combo...) que me propose JavaFX.

```
12 <BorderPane xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1" fx:controller="view.Calendar">
13     <top>
14         <HBox>
15             <ComboBox fx:id="monthChoice">
16                 <items>
17                     <FXCollections fx:factory="observableArrayList">
18
19                     </FXCollections>
20                 </items>
21             </ComboBox>
22             <ComboBox fx:id="yearsComboBox">
23                 <items>
24                     <FXCollections fx:factory="observableArrayList">
25
26                     </FXCollections>
27                 </items>
28             </ComboBox>
29         </HBox>
30     </top>
31     <left>
32         <GridPane fx:id="dayGrid">
33
34         </GridPane>
35     </left>
36     <bottom>
37         <VBox>
38             <TextField minHeight="40" promptText="Titre de la tâche" fx:id="title">
39
40             </TextField>
41             <TextArea minHeight="200" promptText="Description de la tâche" fx:id="description">
42
43             </TextArea>
44             <Button fx:id="submit" text="Valider">
```

Je sais hiérarchiser mes classes pour spécialiser leur comportement.

Je sais gérer la persistance de mon modèle.

Je sais utiliser à mon avantage le polymorphisme.



Je sais intercepter des événements en provenance de la fenêtre JavaFX. (Cf : MainWindow.java)

```
109 del.setOnAction(new EventHandler<ActionEvent>() {
110     @Override
111     public void handle(ActionEvent actionEvent) {
112         Task toDelTask = (Task) todolist.getSelectionModel().selectedItemProperty().get();
113         TaskModel.delTask(tm,toDelTask);
114         todolist.getSelectionModel().selectFirst();
115     }
116 });
```

Je sais surveiller l'élément sélectionné dans un composant affichant un ensemble de données. (Cf : MainWindow.java)

```
89 todolist.getSelectionModel().selectedItemProperty().addListener(new ChangeListener() {
90     @Override
91     public void changed(ObservableValue observableValue, Object o, Object t1) {
92         Task tsk = (Task) t1;
93         try{
94             taskTitle.setText(tsk.getTitle());
95             taskDescription.setText(tsk.getDescription());
96             del.setVisible(true);
97             done.setVisible(true);
98         }
```

Je sais utiliser les lambda-expression. (Cf : BinaryFileTaskFactory.java)

Je sais utiliser les listes observables de JavaFX.

```
41 allTasks.forEach((localDate, task) -> netTasks.put(localDate,new SimpleListProperty<>(FXCollections.observableArrayList(task))));
```

Je sais utiliser un fichier CSS pour styler mon application JavaFX.

```
21 <?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet href="style.css" type="text/css" ?>
<fx:root xmlns:fx="http://javafx.com/javafx/11.0.2" xmlns:fx:skin="http://javafx.com/javafx/11.0.2" style="background-color: #f0f0f0; min-height: -Infinity; min-width: -Infinity; pref-height: 600.0; pref-width: 900.0; stylesheets="@../style/style.css" xmlns="http://javafx.com/javafx/11.0.2" />
```

Je sais utiliser GIT pour travailler avec mon binôme sur le projet.

#### Latest revisions

#		Date	Author	Comment
271d10ed	●	19/01/2021 09:04 PM	Clément JUVENTIN	[FIX] Modification des imports
1c20d213	○ ●	19/01/2021 09:02 PM	Clément JUVENTIN	[DEV] Ajout de la cellfactory
6fbca5ee	○ ○	19/01/2021 07:44 PM	Clément JUVENTIN	[FIX] Style.css et commentaires
dda77aa9	○ ○	19/01/2021 07:05 PM	Clément JUVENTIN	[FIX] Restructuration des fichiers, ajouts d'une factory, simplifi code (lamda expression, idea warnings, ect.)
e7549363	○ ○	18/01/2021 09:37 PM	Clément JUVENTIN	[DEV] Binding des propriétés de temps sur des labels
deedba79	○ ○	18/01/2021 08:51 PM	Clément JUVENTIN	[FIX] Fix sur les dates (Février, année bisextile), fix de bindings réorganisation de l'architecture
c44a56e2	○ ○	18/01/2021 08:47 PM	Axel CROZET	[DEV]: ->Ajout Timer.java (ajout fuseaux horaires)
236652cf	○ ○	18/01/2021 03:26 PM	Clément JUVENTIN	[DEV] Ajout de la persistance en mémoire
8765f261	○ ○	08/01/2021 09:34 AM	Clément JUVENTIN	[DEV] Modification de la vue MainWindow avec restructuration éléments et des styles
d7dee656	○	07/01/2021 03:38 PM	Axel CROZET	Merge remote-tracking branch 'origin/master'

~~Je sais utiliser un convertisseur lors d'un bind entre deux propriétés JavaFX.~~

~~Je sais utiliser un formateur lors d'un bind entre deux propriétés JavaFX.~~