

Git, Gestion de branches sous IntelliJ

Sébastien Jean

IUT de Valence
Département Informatique

v1.2, 6 septembre 2024

Point de départ

- Dépôt distant initial (*fork*) :
 - une branche unique **main** et **2 commits**

The screenshot shows a GitLab repository page for 'branches' (Project ID: 5141). The repository is a dummy repository for git branching training using IntelliJ. It has 2 commits and 1 branch (main). The main branch is selected. The repository contains a README.md file and a src directory. The commit history shows two commits: 'Initial commit' (24 minutes ago) and 'Added Hello World application' (24 minutes ago). The repository is owned by Sébastien Jean.

Name	Last commit	Last update
idea	Initial commit	24 minutes ago
src	Added Hello World application	24 minutes ago
README.md	Initial commit	24 minutes ago

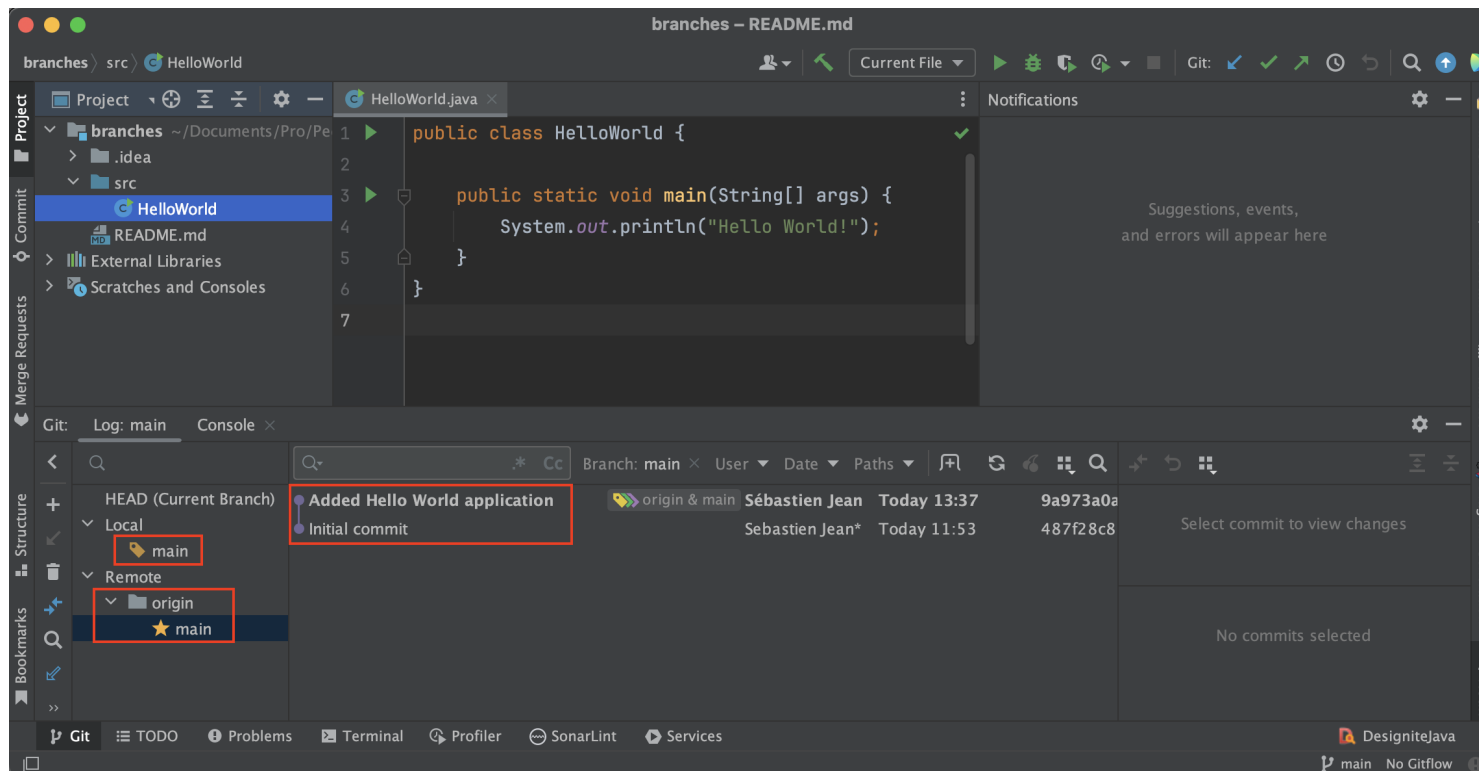
README.md

Git-Branches

Dummy repository for git branching training using IntelliJ.

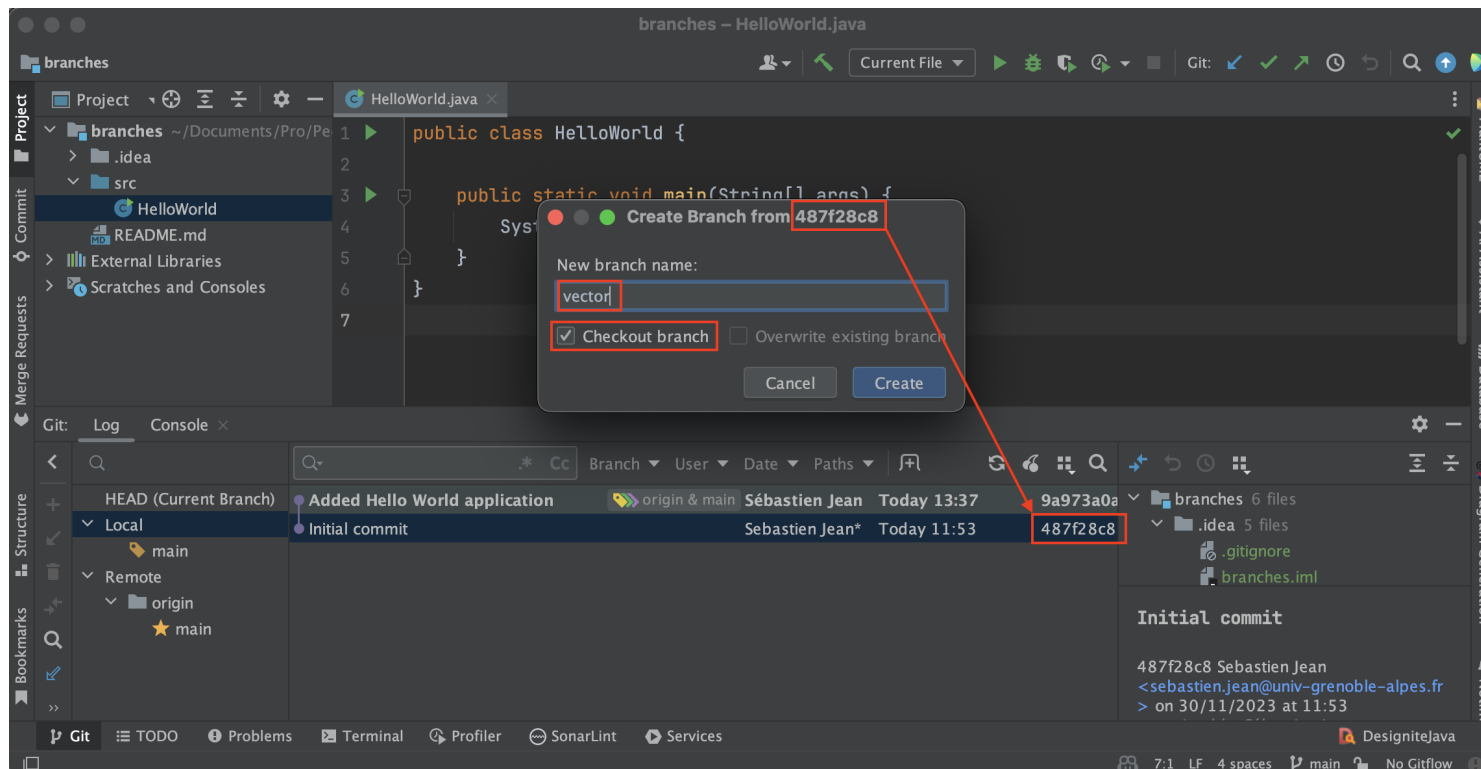
Point de départ

- Dépôt local initial (*clone*) :
 - une branche unique **main** (synchronisée) et **2 commits**



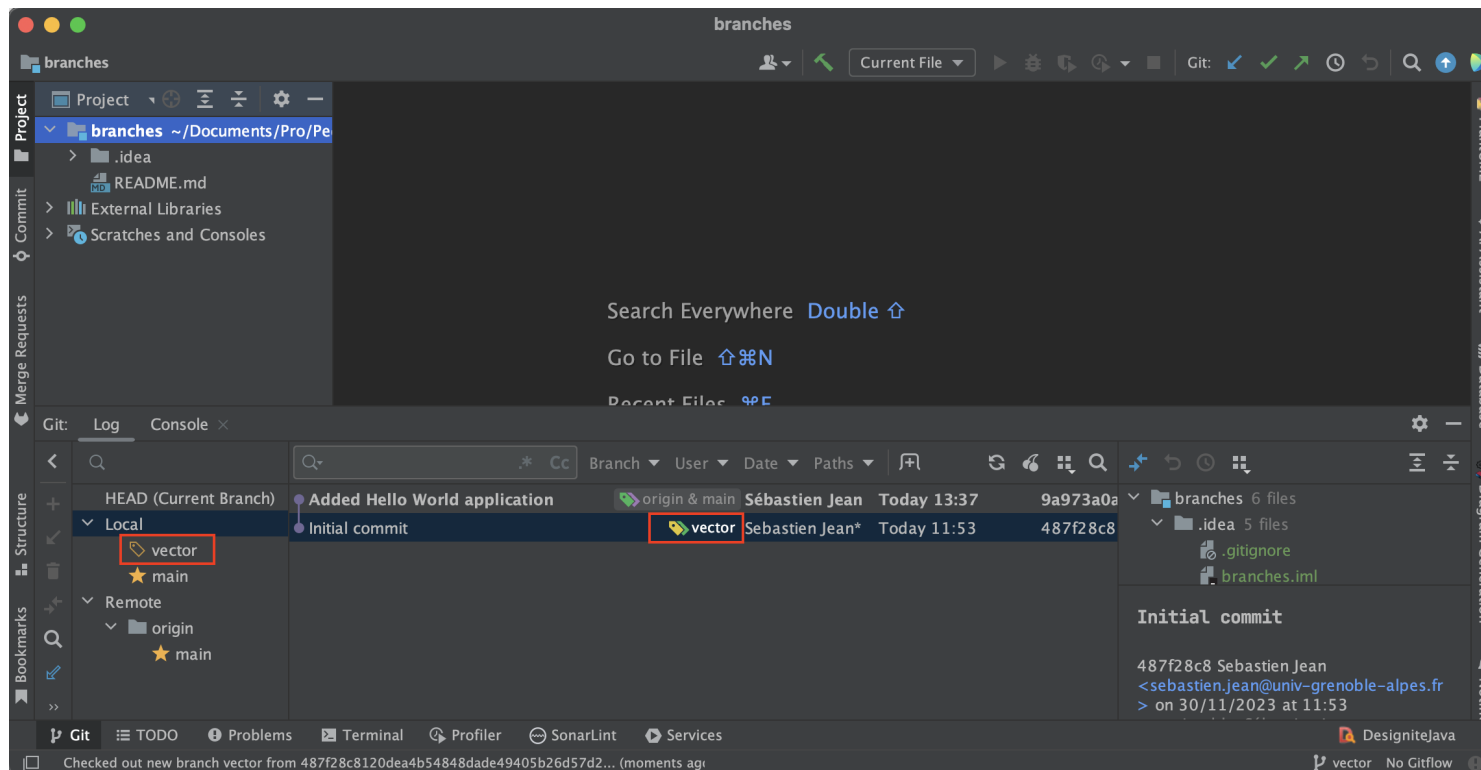
Créer une nouvelle branche

- En se plaçant sur un **commit** (ici 487f28c8)
- En indiquant un **nom** de branche
- Avec en **option** la possibilité de se **déplacer immédiatement** sur la nouvelle branche



Créer une nouvelle branche, suite

- La branche est créée **localement**
- Le **répertoire de travail** reflète le **dernier commit** sur la branche



Créer une nouvelle branche, suite

- La branche est créée **localement uniquement**

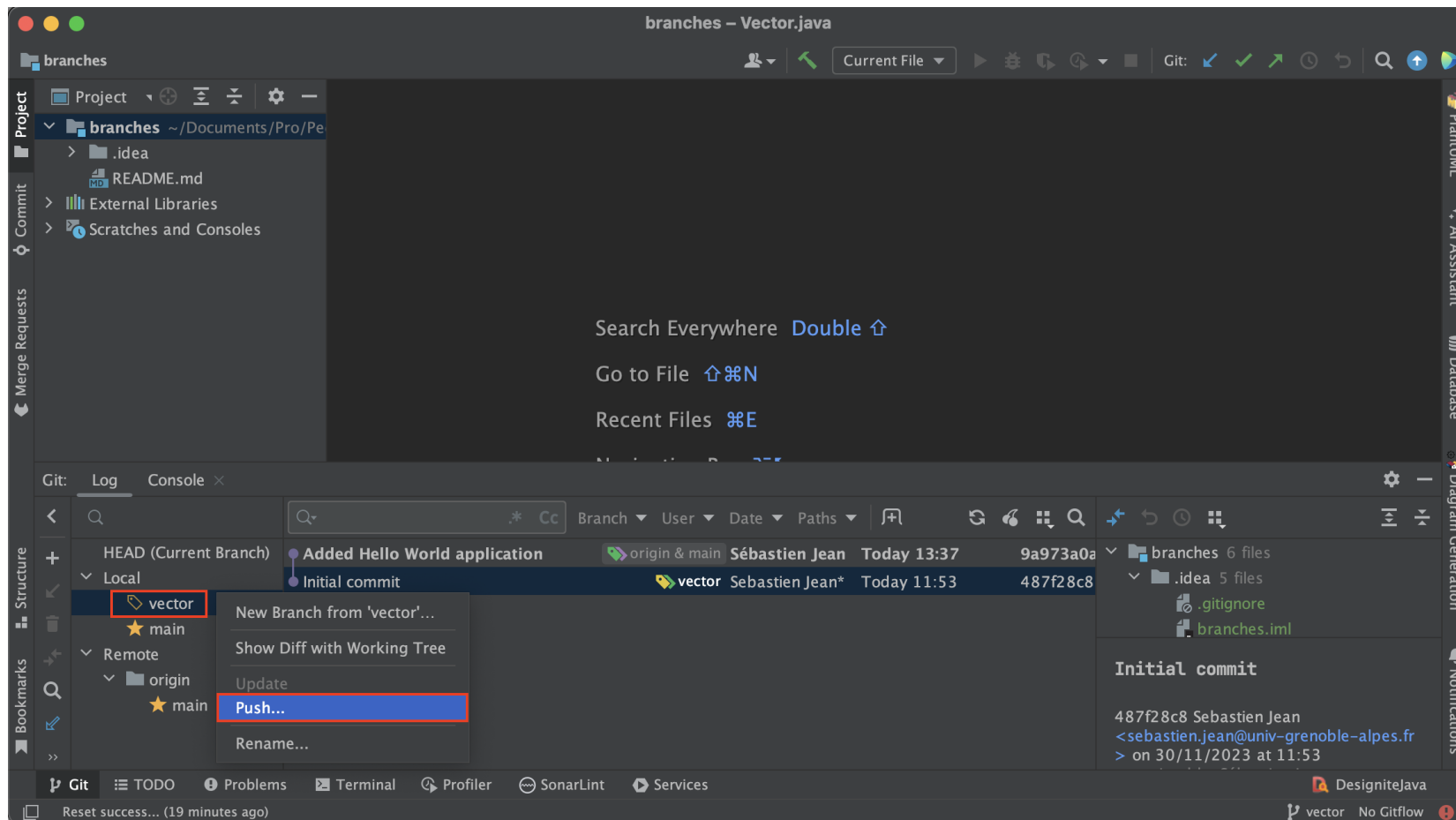
The screenshot shows the GitLab web interface for a repository named 'branches' (Project ID: 5141). The interface includes a sidebar with navigation options like 'Project', 'Pinned', 'Manage', 'Code', and 'Settings'. The main content area displays the repository's commit history, showing a single commit titled 'Added Hello World application' by Sébastien Jean, authored 1 hour ago. Below the commit history, there is a table listing the files in the repository:

Name	Last commit	Last update
.idea	Initial commit	24 minutes ago
src	Added Hello World application	24 minutes ago
README.md	Initial commit	24 minutes ago

Below the table, the content of the README.md file is displayed, showing the title 'Git-Branches' and the description 'Dummy repository for git branching training using IntelliJ.' The 'main' branch is selected in the branch dropdown menu.

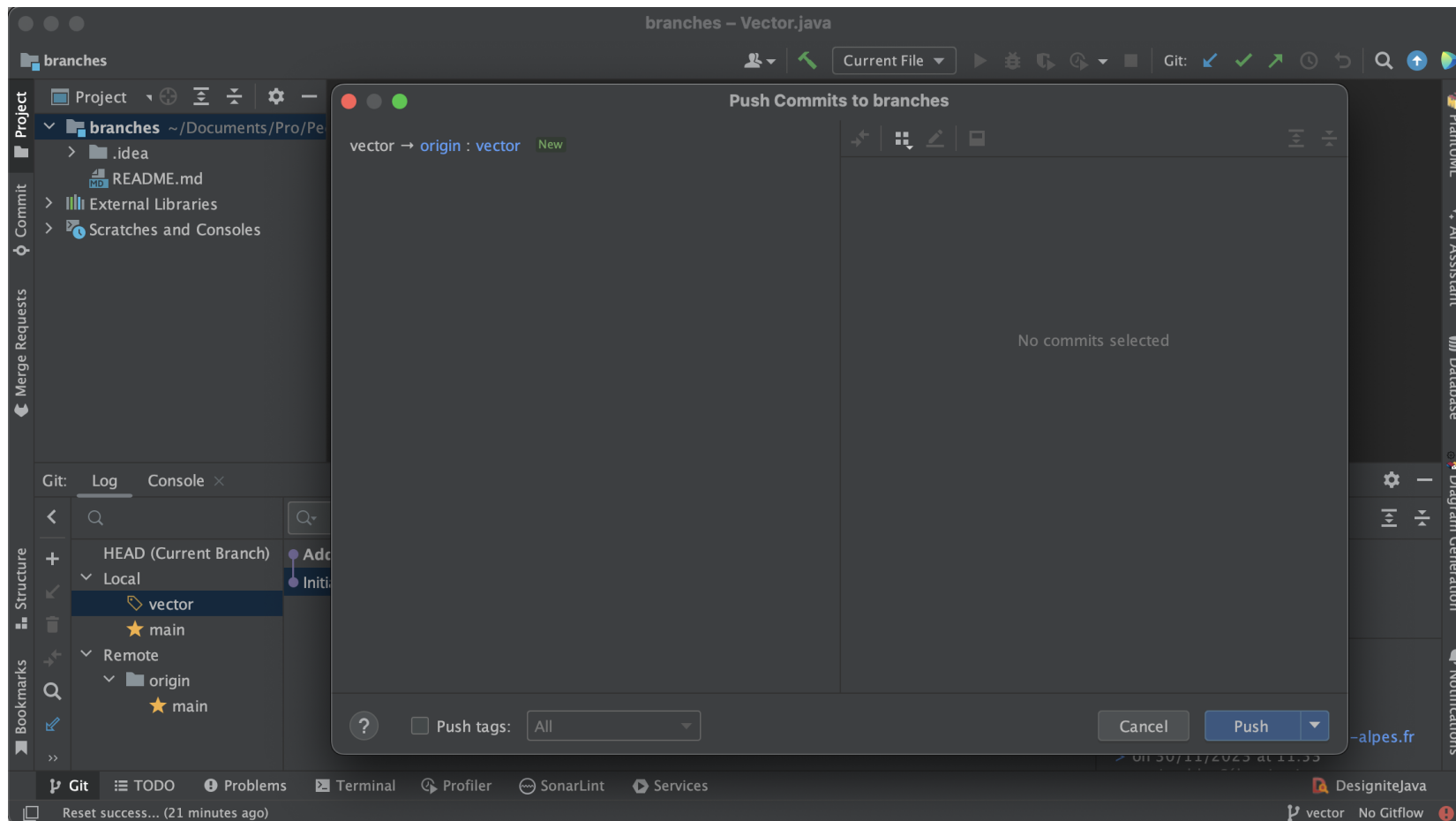
Publier une branche

- La branche doit être **publiée explicitement** sur le serveur



Publier une branche

- (La branche doit être **publiée explicitement** sur le serveur)



Publier une branche, suite

- La branche distante est créée

The screenshot shows a GitLab repository page for 'branches' (Project ID: 5141). The repository is a dummy repository for git branching training using IntelliJ. The page displays 2 commits and 2 branches. A dropdown menu is open, showing the 'vector' branch being selected. The commit history table is as follows:

	Last commit	Last update
	Initial commit	1 hour ago
	Added Hello World application	1 hour ago
+++ README.md	Initial commit	1 hour ago

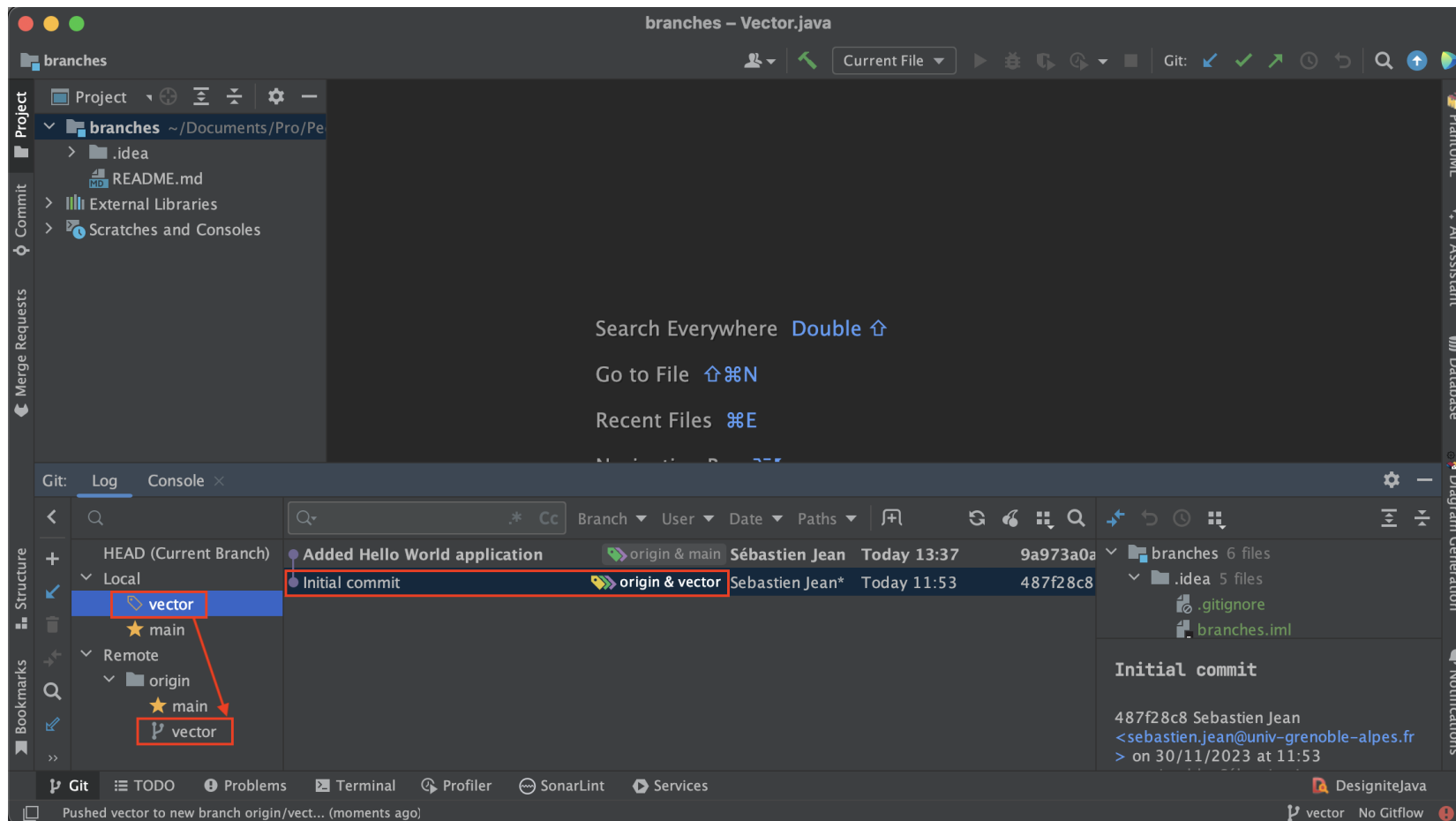
The README.md file content is:

```
Git-Branches

Dummy repository for git branching training using IntelliJ.
```

Publier une branche, fin

- La branche **locale** est **associée** à la branche **distante** (*remote-tracking*)



Effectuer un commit sur une branche

- Un **commit** sur la branche `vector` fait avancer la branche `vector` **indépendamment** de `main`

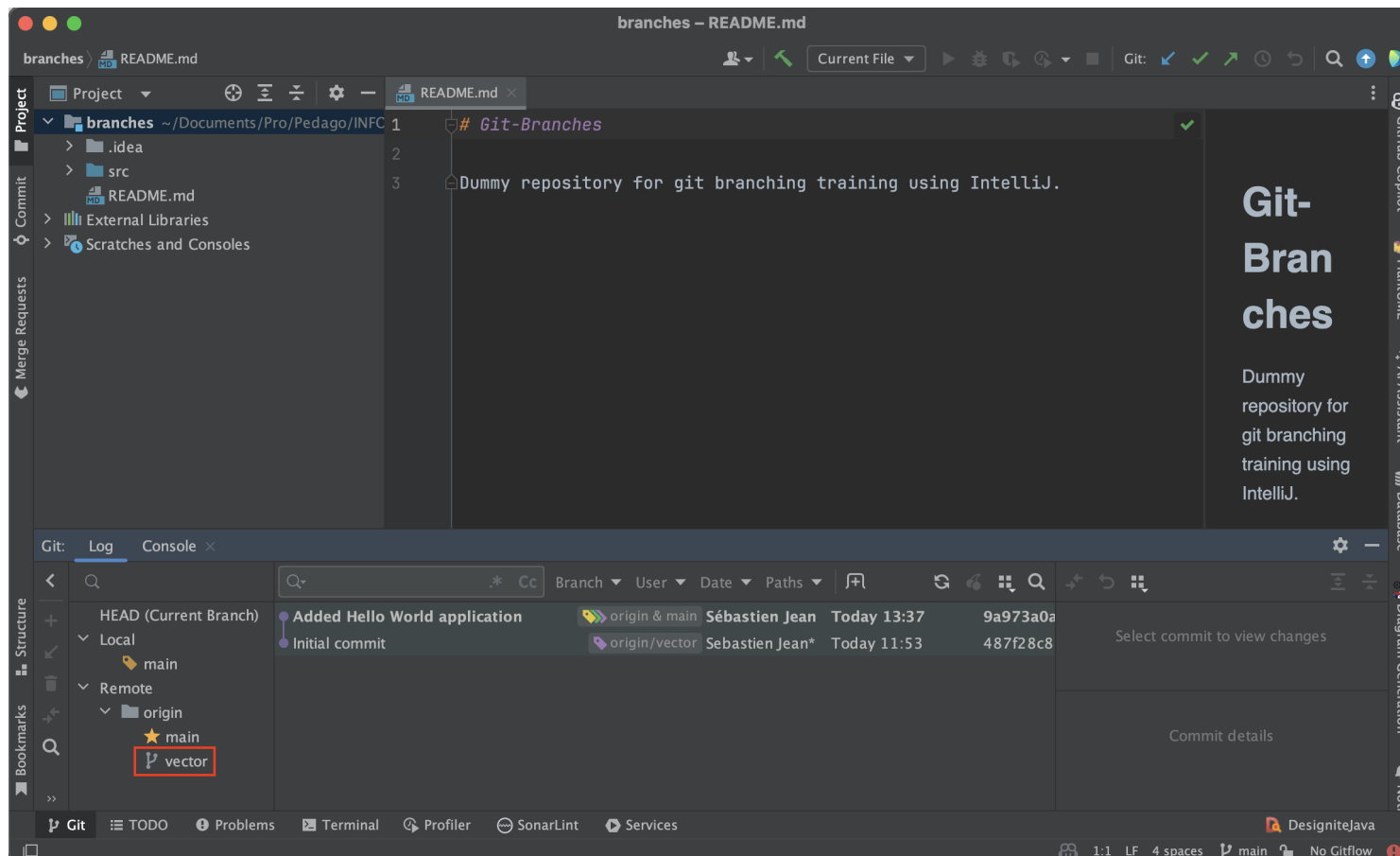
The screenshot shows an IDE window titled "branches - Vector.java". The main editor displays the code for a `Vector` class with two private final double attributes (`cx` and `cy`) and two public getter methods (`getCx()` and `getCy()`). The bottom panel shows the Git log for the `vector` branch. The log entries are:

Commit Hash	Message	Author	Date	SHA-1
97dfb9b9	Added Vector class	Sébastien Jean	Moments ago	97dfb9b9
9a973a0a	Added Hello World application	Sébastien Jean	Today 13:37	9a973a0a
487f28c8	Initial commit	Sebastien Jean*	Today 11:53	487f28c8

The `Initial commit`

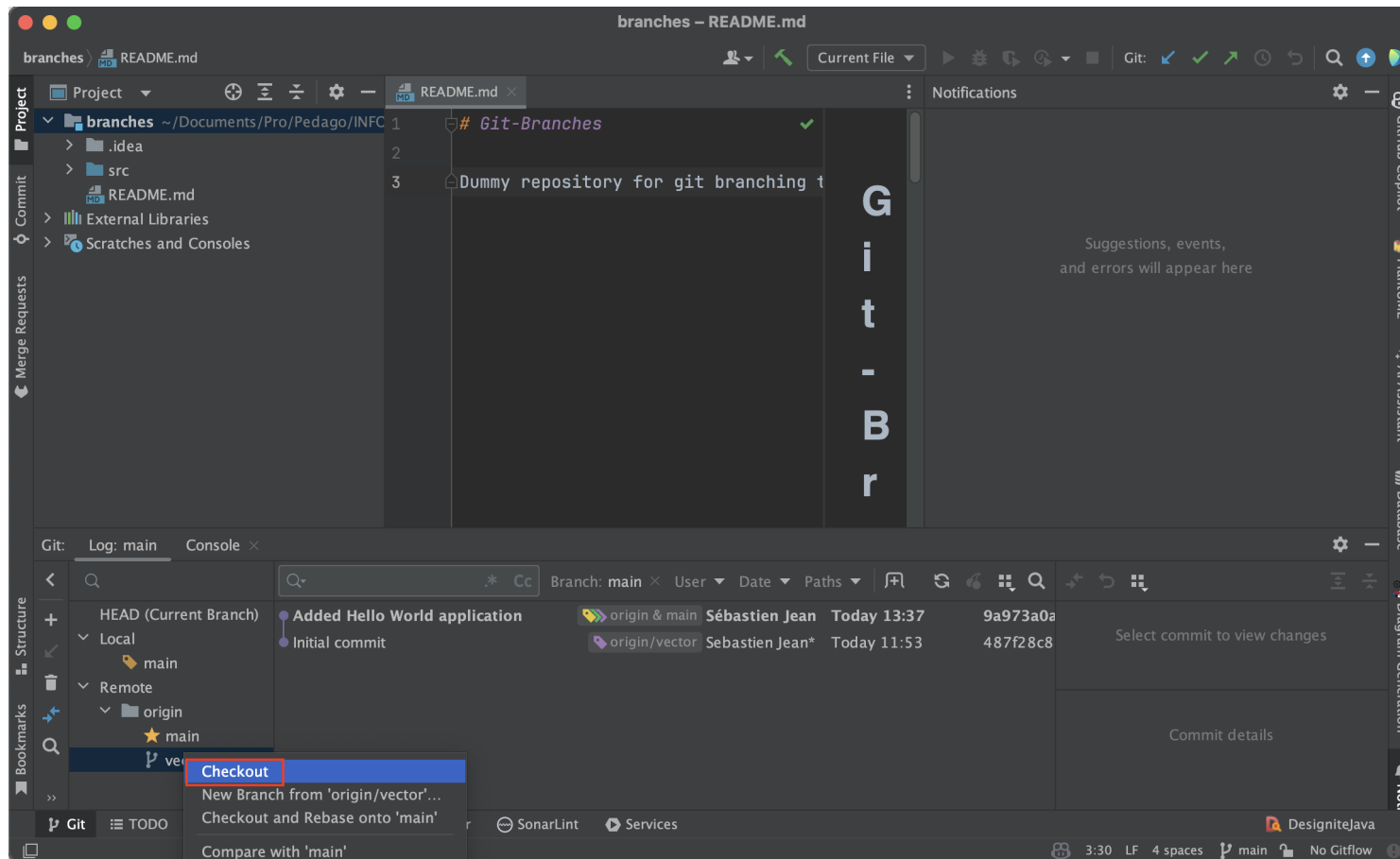
Cloner et reconstruire les branches

- (au préalable) suppression du clone local
- Le **clonage d'un dépôt** contenant **plusieurs branches** :
 - **rapatrie** toutes les branches distantes (*fetch*)
 - **crée uniquement** la branche principale `main`

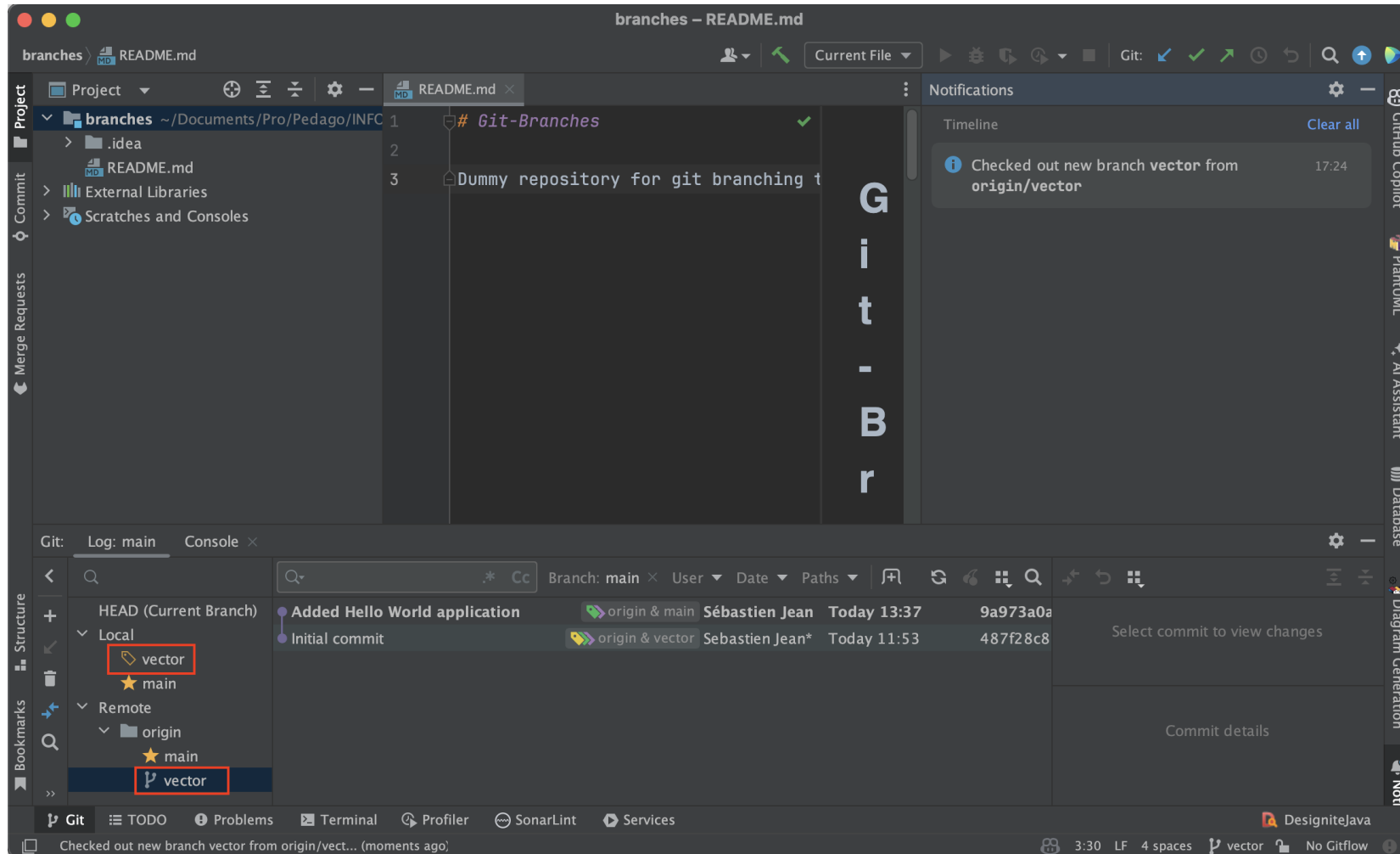


Cloner et reconstruire les branches, suite

- La création d'une branche locale à partir d'une branche distante est explicite



Cloner et reconstruire les branches, fin



Création d'une branche, sans publication, et commit

- (au préalable) rajout du commit perdu sur la branche vector
- création de la branche point et ajout d'un commit

The screenshot displays an IDE window titled "branches - Point.java". The code editor shows the following Java code:

```
1 public class Point {
2
3     private final double x;
4     private final double y;
5
6     public Point(double x, double y) {...}
7
8
9
10
11     public double getX() { return this.x; }
12
13
14
15     public double getY() { return this.y; }
16
17
18 }
```

The Git log shows the following commits:

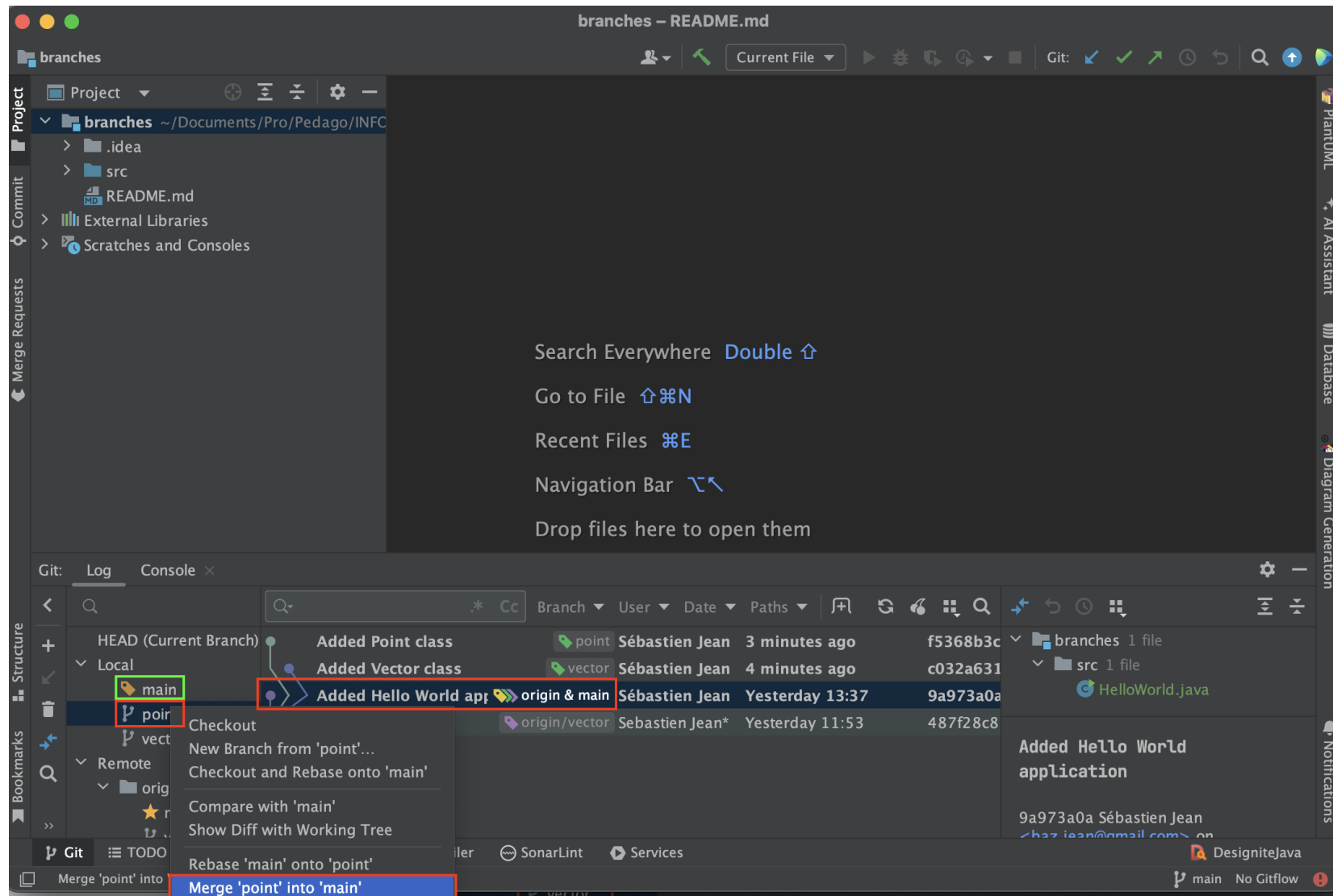
Commit	Author	Date	Hash
Added Point class	Sébastien Jean	Moments ago	662be57c
Added Vector class	Sébastien Jean	31 minutes ago	97dfb9b9
Added Hello World application	Sébastien Jean	Today 13:37	9a973a0a
Initial commit	Sebastien Jean*	Today 11:53	487f28c8

The Git structure shows the following branches:

- HEAD (Current Branch)
- Local
 - point
 - main
 - vector
- Remote
 - origin
 - main
 - vector

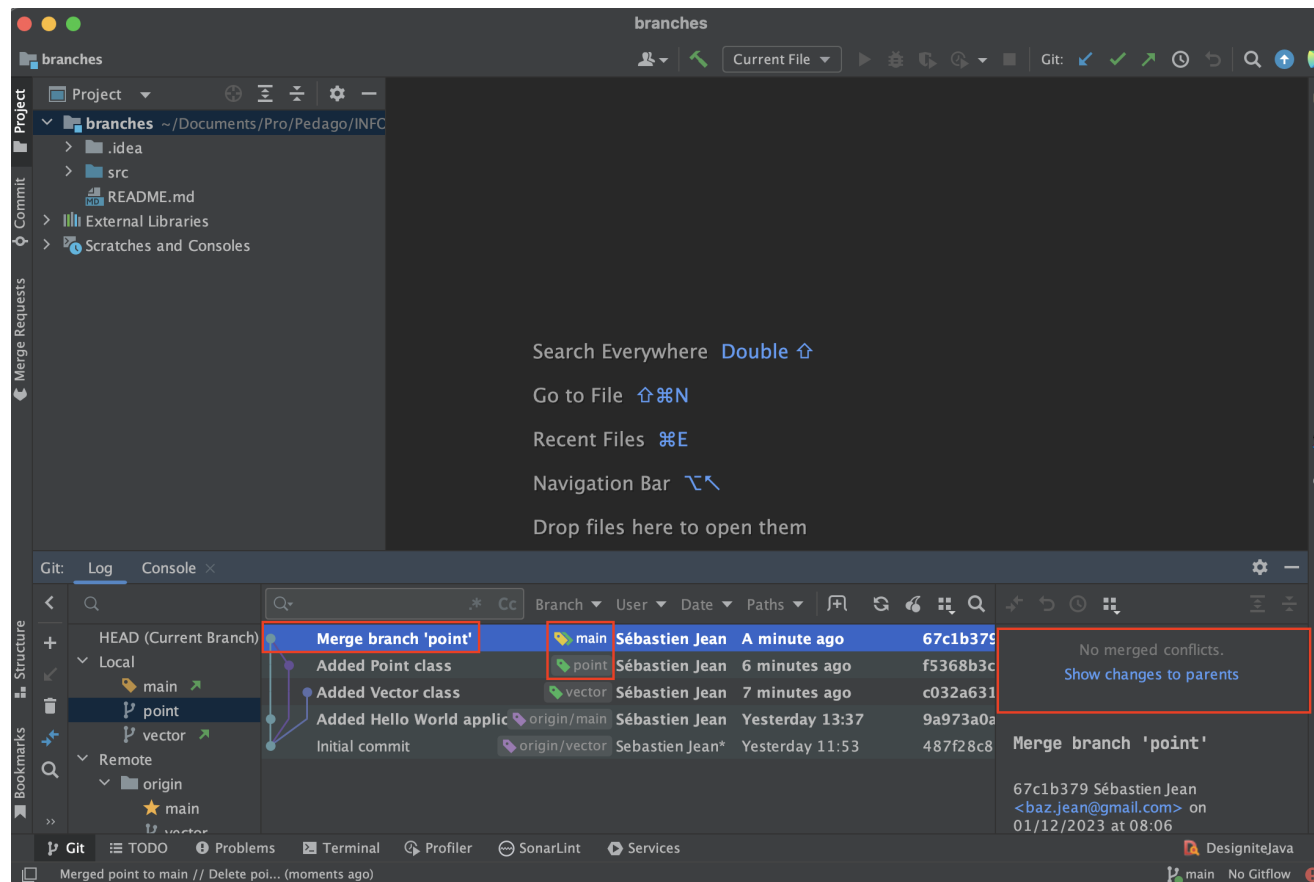
Fusionner 2 branches locales (*merge*)

- La fusion s'opère d'une **branche sélectionnée** vers la **branche courante**



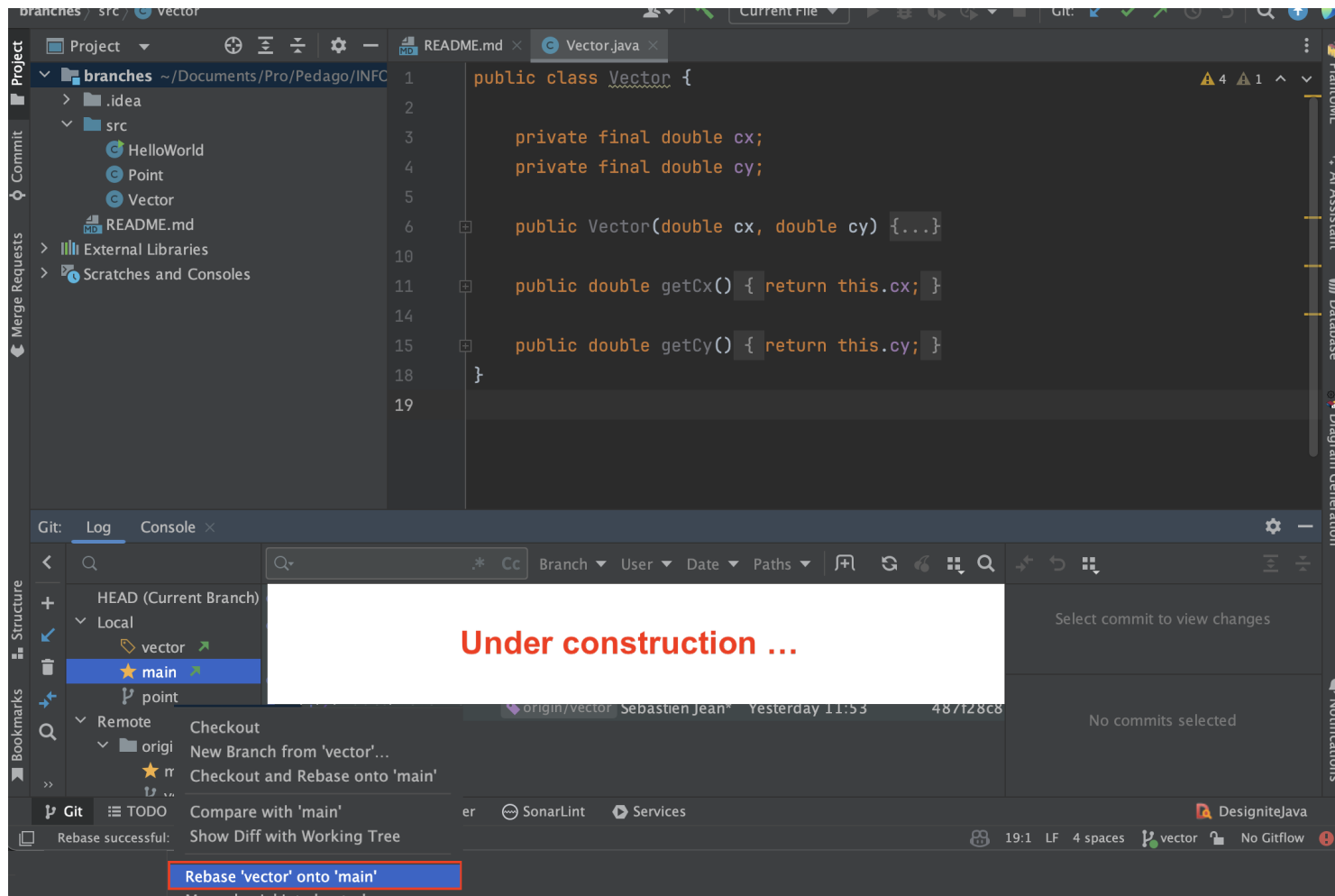
Fusionner 2 branches locales (*merge*), fin

- La **branche courante intègre** les commits de la branche sélectionnée, **pas l'inverse**
 - La fusion est marquée par un **commit (de fusion) ayant 2 parents**
 - Quand il n'y a **pas de conflit**, le commit n'ajoute pas de modification



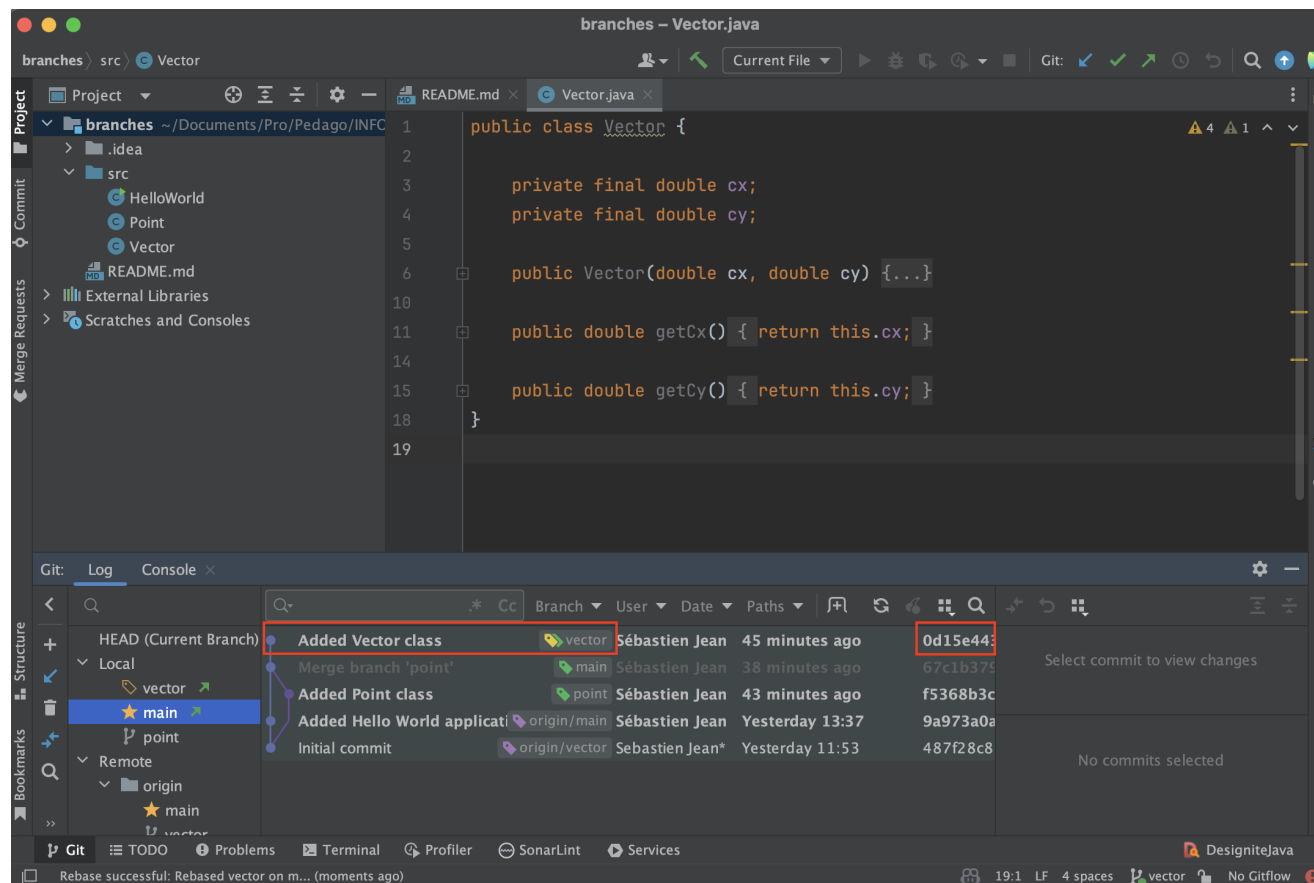
Rebaser une branche locale (*rebase*)

- Les commits de la **branche courante** sont **recalculés** pour se **replacer** au **dessus** du **dernier commit** de la **branche sélectionnée**



Rebaser une branche locale (*rebase*), fin

- La **branche courante intègre** les commits de la branche sélectionnée, **pas l'inverse**
 - Les commits sont bien recalculés (hash différent)
 - En cas de **conflit**, chaque **commit recalculé** inclura également les **modifications de résolution manuelle**



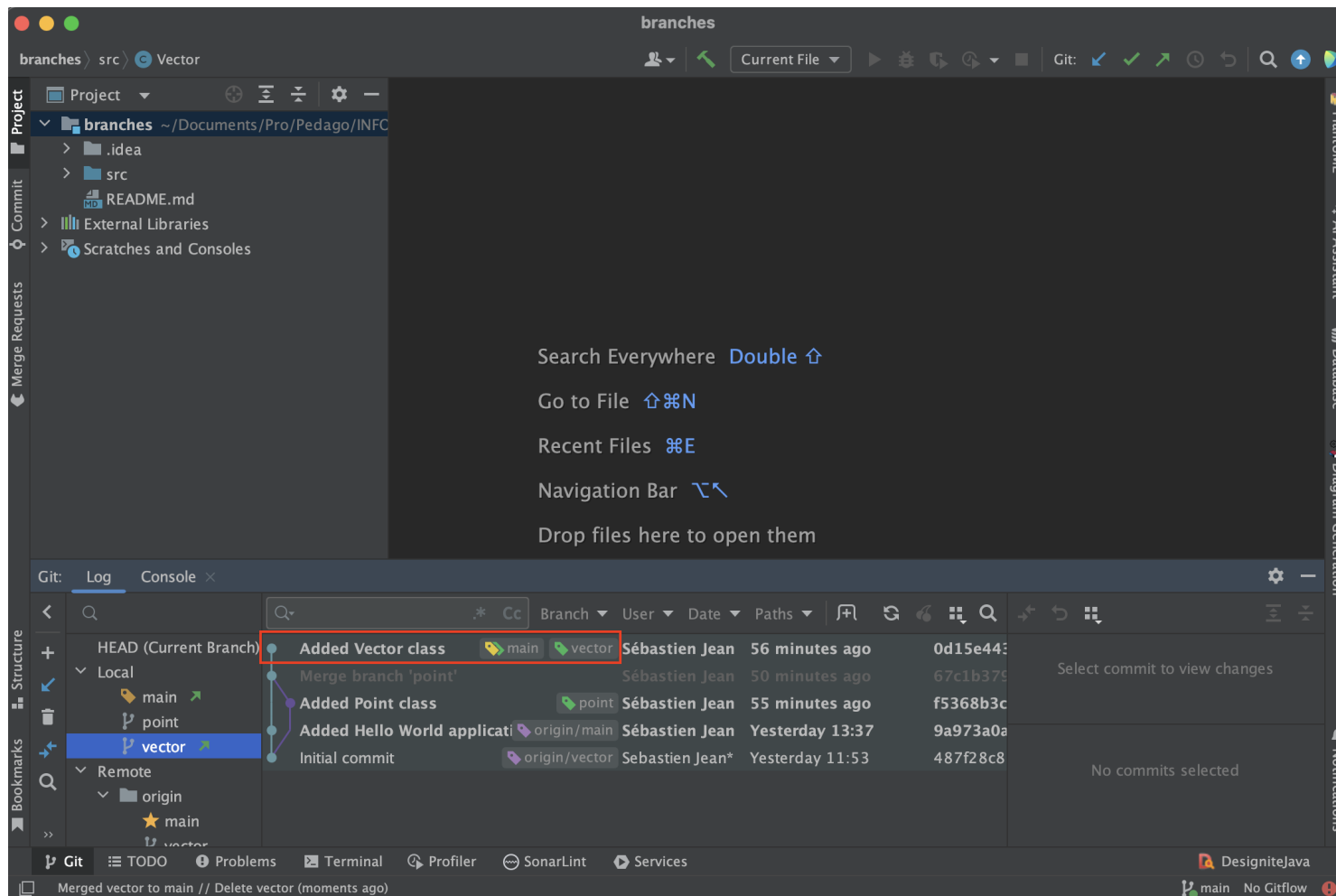
Fusionner 2 branches locales (*merge*) sans divergence

The screenshot shows the IntelliJ IDEA interface with the Git tool window open. The 'Log' tab is active, displaying a commit history. A context menu is open over the 'main' branch, with the option 'Merge 'vector' into 'main'' highlighted. The commit history shows a merge of 'point' into 'main'.

Commit	Message	Author	Date	SHA-1
0d15e443	Added Vector class	Sébastien Jean	54 minutes ago	0d15e443
67c1b379	Merge branch 'point'	Sébastien Jean	48 minutes ago	67c1b379
f5368b3c	Added Point class	Sébastien Jean	53 minutes ago	f5368b3c
9a973a0a	Added Hello World applicati	Sébastien Jean	Yesterday 13:37	9a973a0a
487f28c8		Sebastien Jean*	Yesterday 11:53	487f28c8

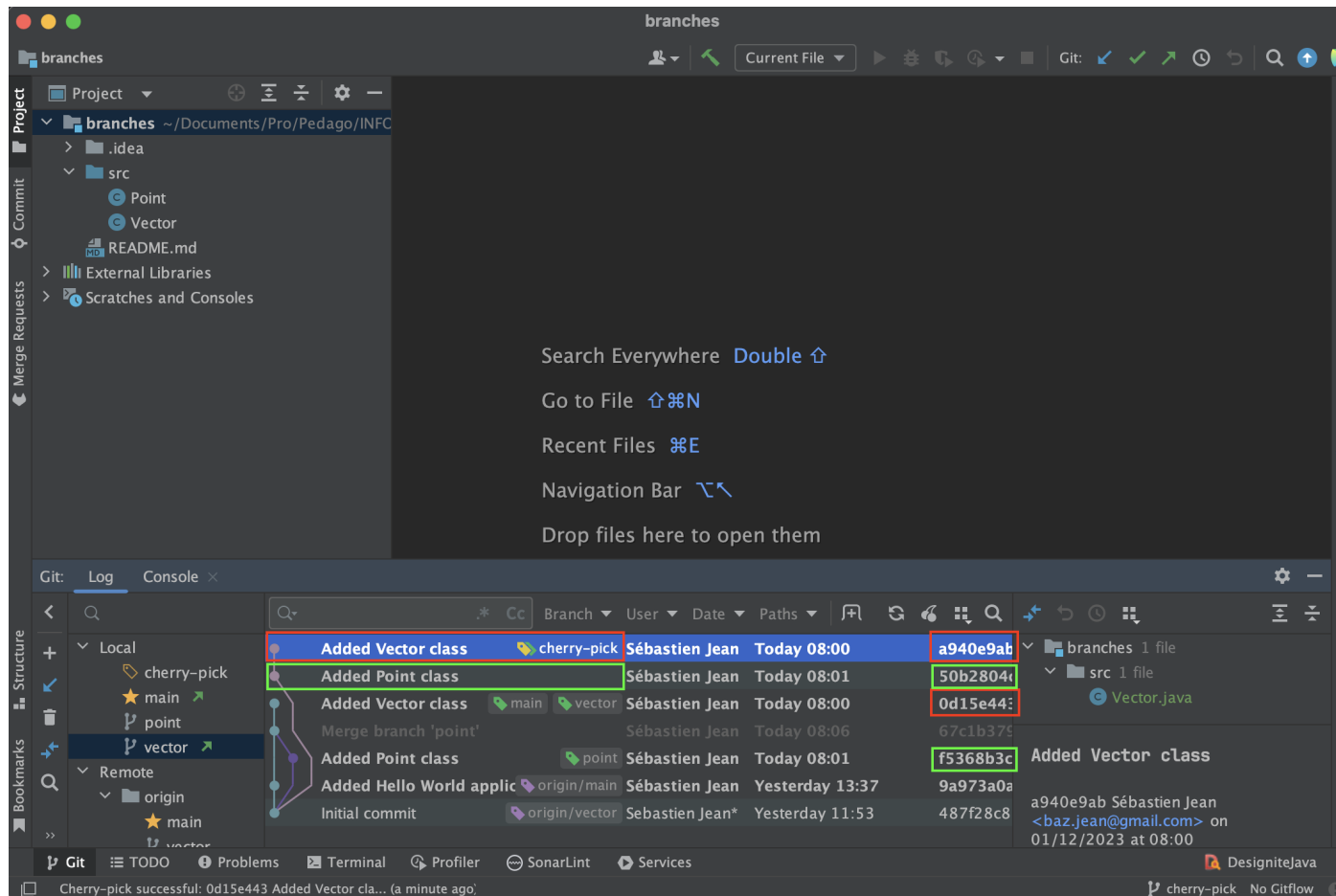
Fusionner 2 branches locales (*merge*) sans divergence, fin

- Le commit de fusion n'est **pas nécessaire**, seul le **pointeur de branche se déplace**



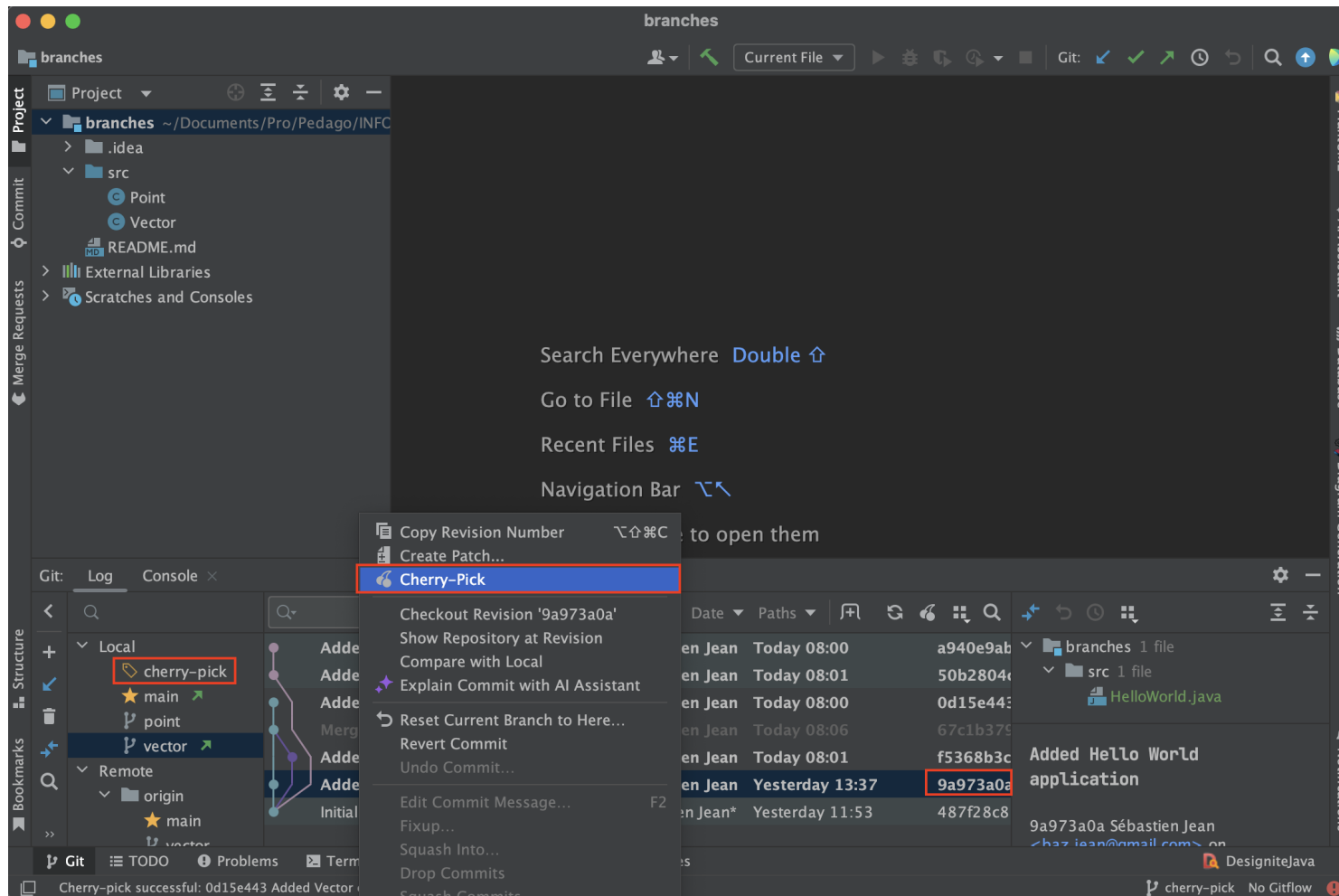
Picorer des commits (*cherry-pick*)

- Intégrer individuellement des commits d'autres branches
 - Ici, on a créé une branche `cherry-pick` à partir du **commit initial**
 - On y a **intégré deux commits** provenant des branches `point` et `vector`, recalculés



Picorer des commits (*cherry-pick*), suite

- On intègre un autre commit plus ancien ... ()



Picorer des commits (*cherry-pick*), fin

- (On intègre un autre commit plus ancien) ... **recalculé** également

The screenshot shows the IntelliJ IDEA interface with the Git Log window open. The log displays a list of commits, with the following entries highlighted:

Commit Hash	Message	Author	Date
9b3cd024	Added Hello World application	Sébastien Jean	Yesterday 13:37
9a973a0a	Added Hello World applic	Sébastien Jean	Yesterday 13:37

The commit 'Added Hello World applic' with hash 9a973a0a is highlighted in red, indicating it is the commit being cherry-picked. The commit 'Added Hello World application' with hash 9b3cd024 is highlighted in blue, indicating it is the commit being integrated. The commit 'Added Hello World applic' with hash 9a973a0a is also highlighted in red, indicating it is the commit being integrated.

Nouveau point de départ

The screenshot shows an IDE window titled "branches - Main.java". The main editor displays the following Java code:

```

1 public class Main {
2
3     public static void main(String[] args) {
4         System.out.println(new Point( x: 0, y: 0));
5     }
6 }
7

```

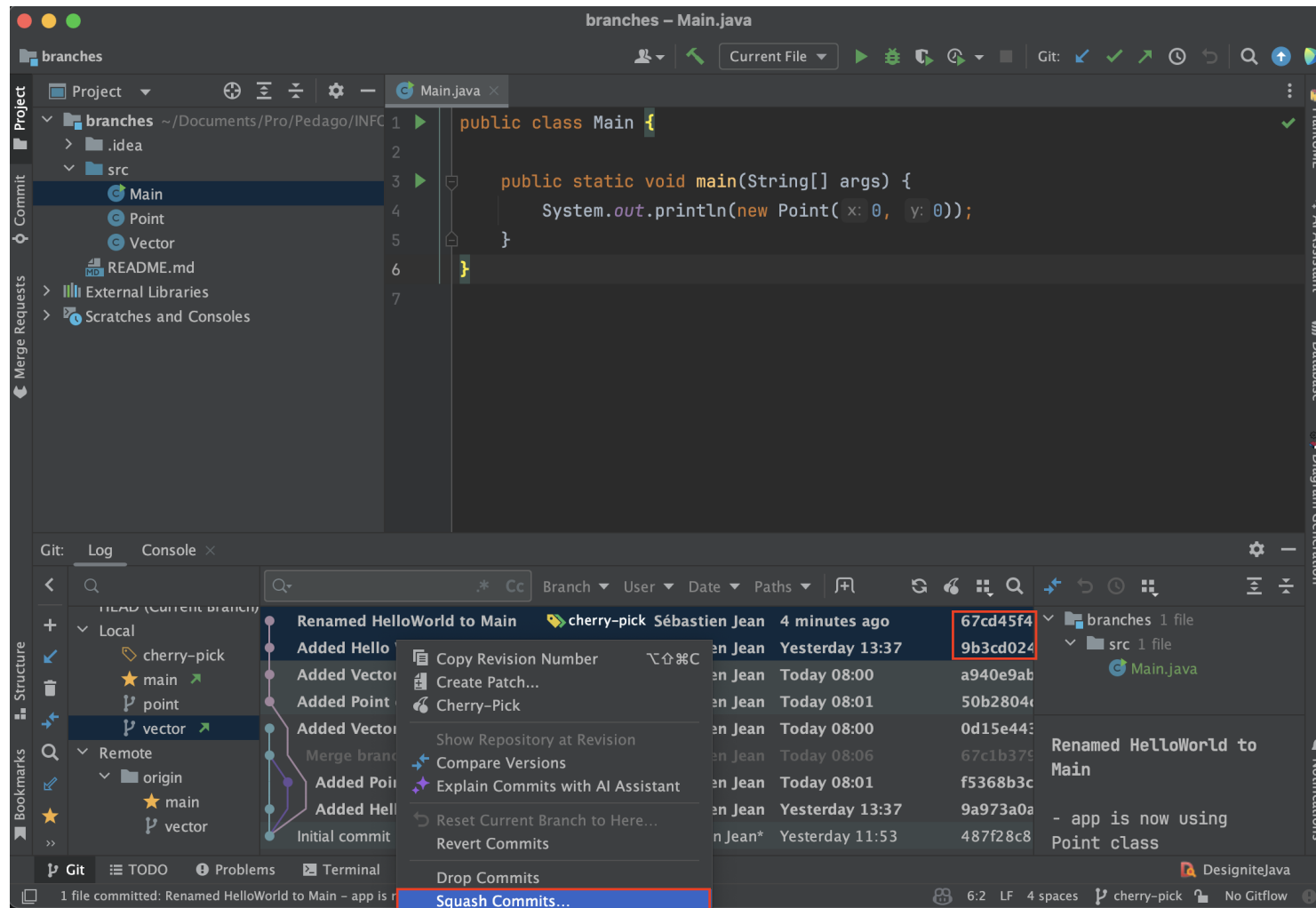
The left sidebar shows the project structure with "Main" selected under the "src" directory. The bottom panel shows the Git Log with the following entries:

Commit Hash	Message	User	Date	SHA-1
67cd45f4	Renamed HelloWorld to Main	Sébastien Jean	3 minutes ago	67cd45f4
9b3cd024	Added Hello World application	Sébastien Jean	Yesterday 13:37	9b3cd024
a940e9ab	Added Vector class	Sébastien Jean	Today 08:00	a940e9ab
50b2804c	Added Point class	Sébastien Jean	Today 08:01	50b2804c
0d15e443	Added Vector class	Sébastien Jean	Today 08:00	0d15e443
67c1b379	Merge branch 'point'	Sébastien Jean	Today 08:06	67c1b379
f5368b3c	Added Point class	Sébastien Jean	Today 08:01	f5368b3c
9a973a0a	Added Hello World application	Sébastien Jean	Yesterday 13:37	9a973a0a
487f28c8	Initial commit	Sebastien Jean*	Yesterday 11:53	487f28c8

The commit log also shows a "cherry-pick" icon next to the "Renamed HelloWorld to Main" commit. The bottom status bar indicates "1 file committed: Renamed HelloWorld to Main - app is now using Point cla... (2 minutes ago)".

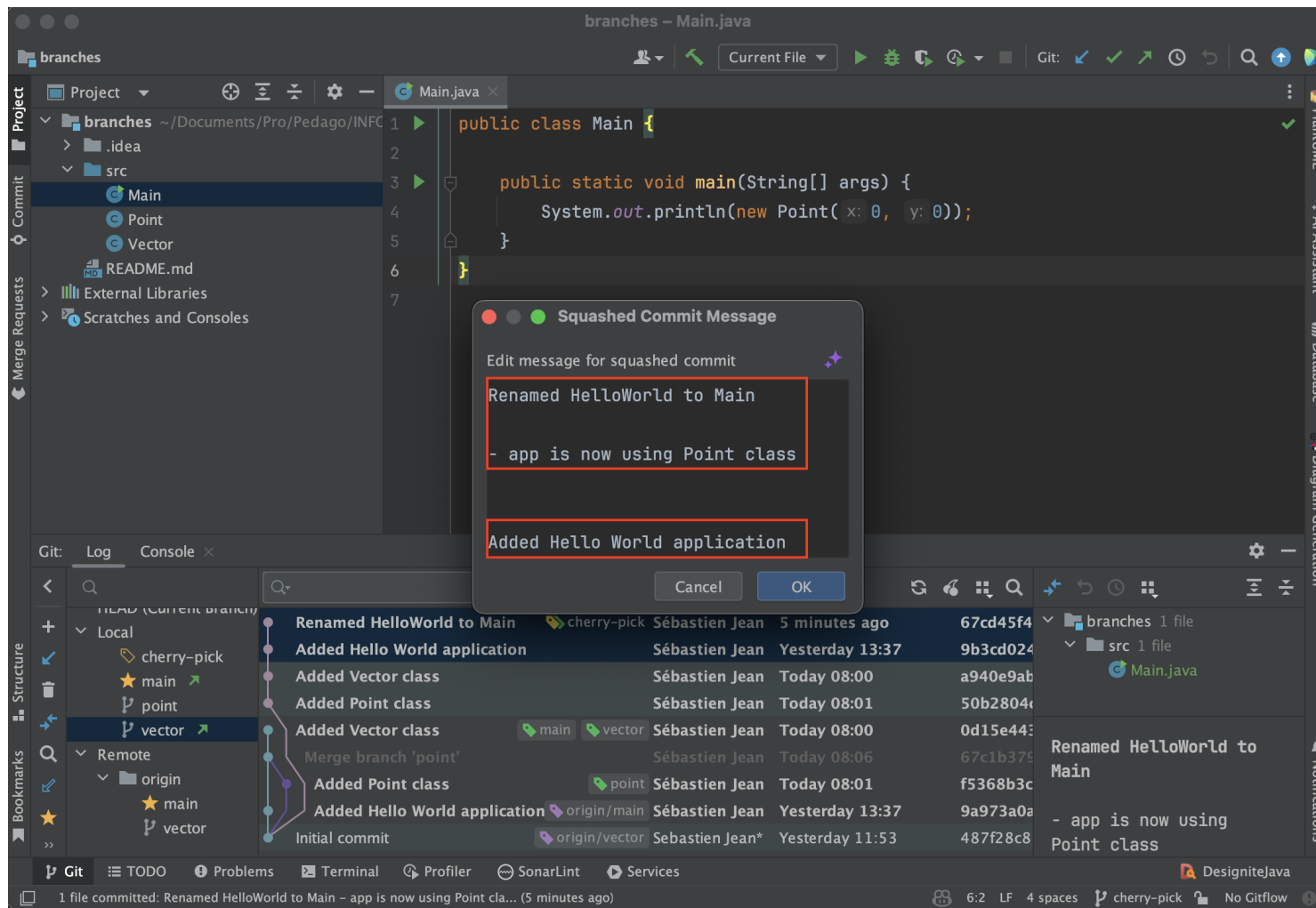
Regrouper des commits (*squash*)

- **Compacter localement 2 commits** de la **même branche**
- Attention : « interdit » si les commits ont déjà été poussés



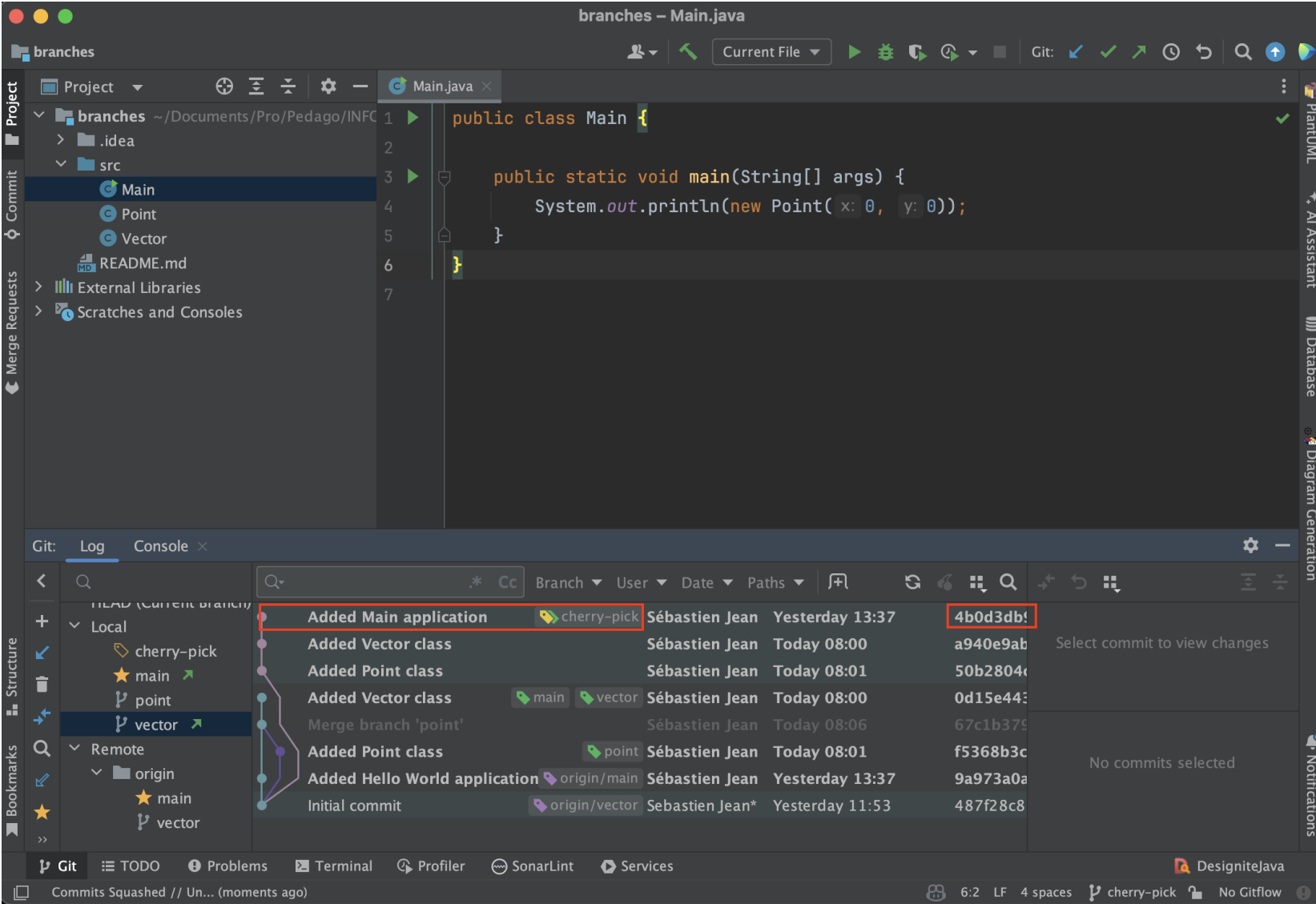
Regrouper des commits (*squash*), suite

- Le message peut être re-rédigé à partir des 2 messages initiaux



Regrouper des commits (*squash*), suite

- Le nouveau commit est recalculé

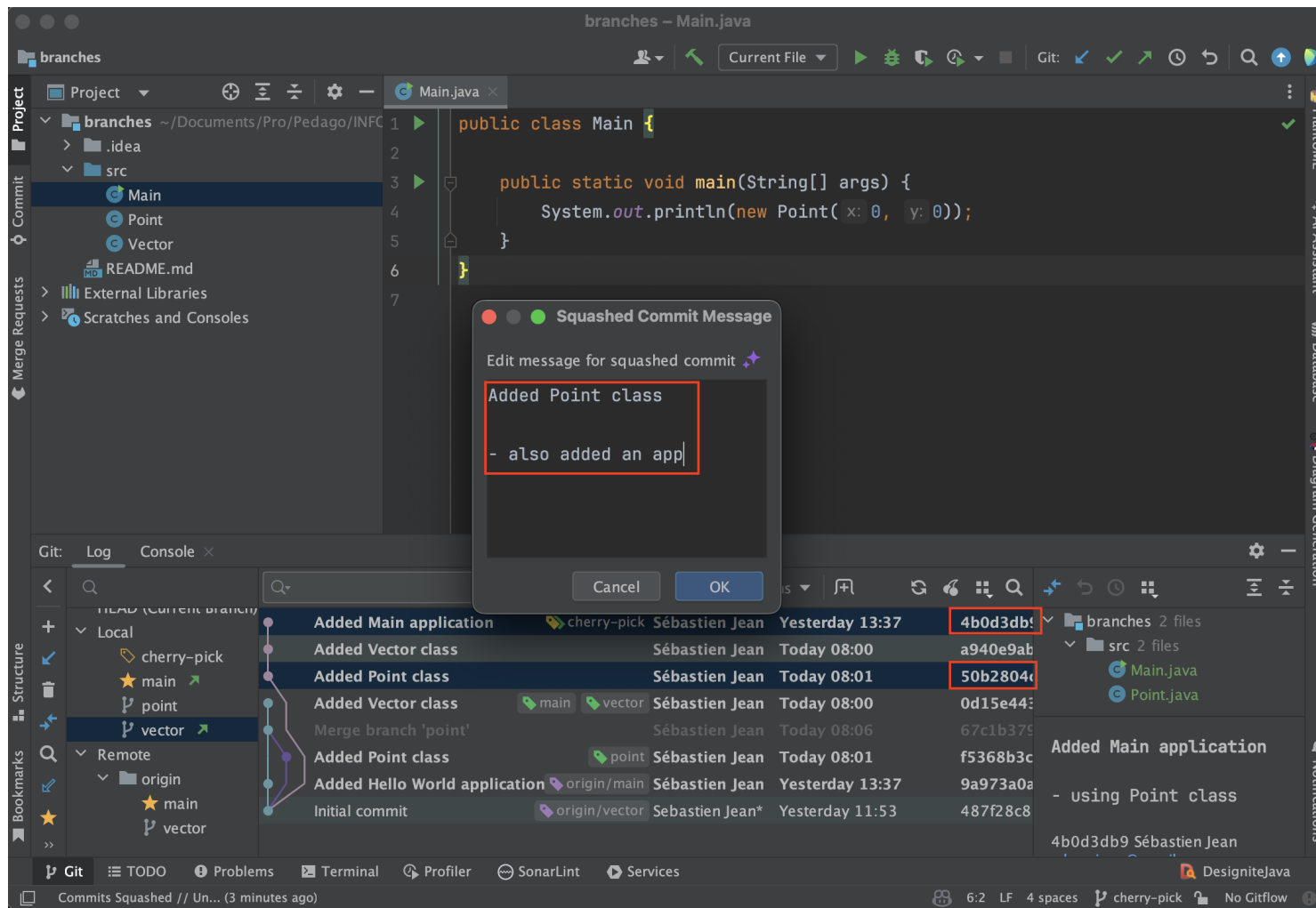


The screenshot shows the IntelliJ IDEA interface with a Git commit log window open. The log displays a series of commits, with the top commit highlighted in red, indicating it is the selected commit. The highlighted commit is titled "Added Main application" and has a commit hash of 4b0d3db1. The log also shows other commits, including "Added Vector class", "Added Point class", and "Merge branch 'point'". The commit log is filtered by the "cherry-pick" branch.

Commit Message	Author	Date	Commit Hash
Added Main application	Sébastien Jean	Yesterday 13:37	4b0d3db1
Added Vector class	Sébastien Jean	Today 08:00	a940e9ab
Added Point class	Sébastien Jean	Today 08:01	50b2804c
Added Vector class	Sébastien Jean	Today 08:00	0d15e443
Merge branch 'point'	Sébastien Jean	Today 08:06	67c1b379
Added Point class	Sébastien Jean	Today 08:01	f5368b3c
Added Hello World application	Sébastien Jean	Yesterday 13:37	9a973a0a
Initial commit	Sebastien Jean*	Yesterday 11:53	487f28c8

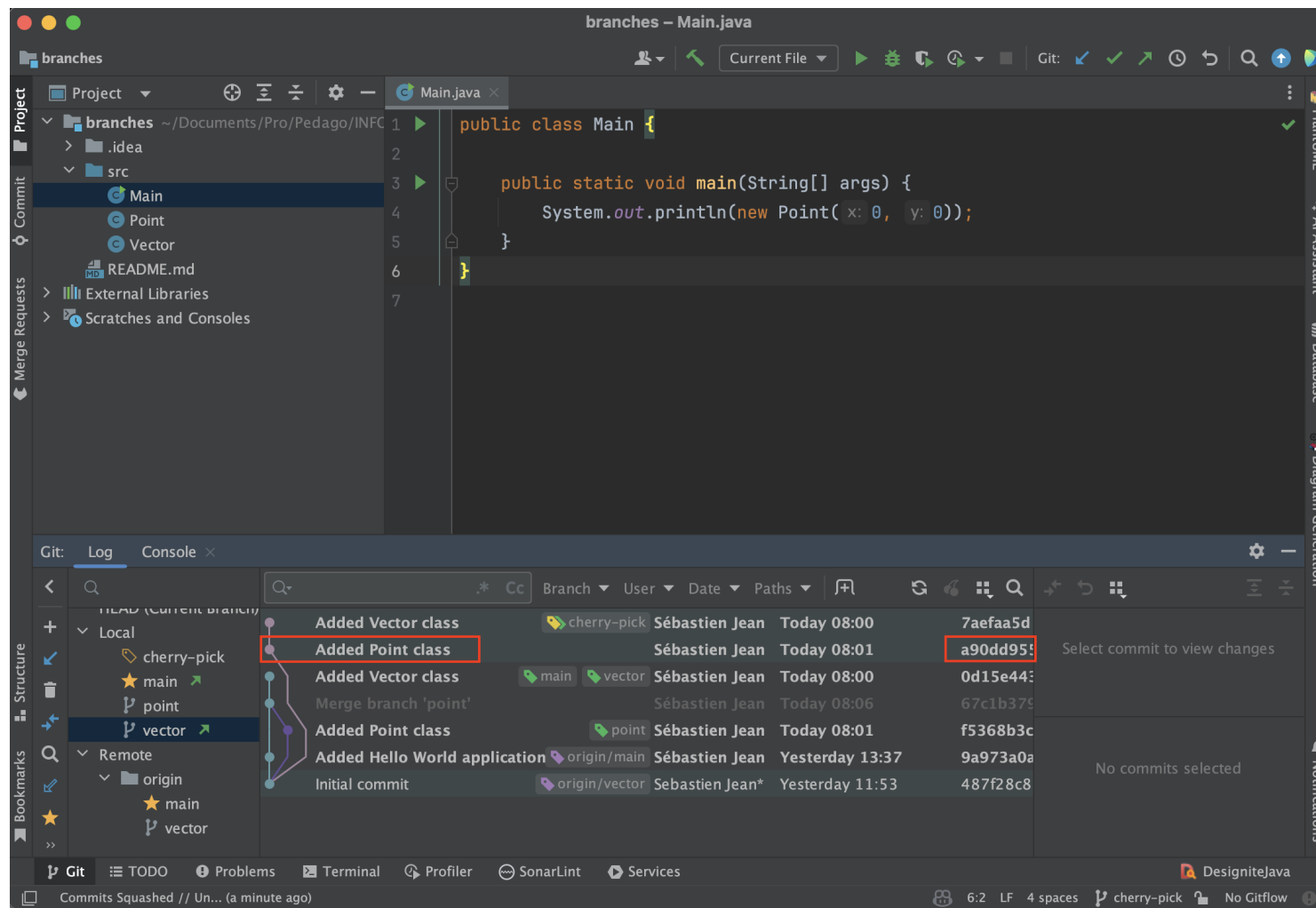
Regrouper des commits (*squash*), suite

- Les 2 commits ne sont pas forcément successifs



Regrouper des commits (*squash*), fin

- Le nouveau commit est recalculé
- Attention : il peut y avoir une **résolution de conflit à opérer**



Fin !

