

Compte Rendu

Intégration des compétences

Réalisé par
EL ASRI Ayoub
FERCHICHI Naima
FOURCADET Maxime

Groupe
 4A SAGI TD1

I. Conception UML

A. Diagramme de classe

Le diagramme de classes illustre trois classes, la classe DossierBancaire étant la classe accessible à l'utilisateur, contient deux attributs de type CompteCourant et CompteEpargne (**relation de composition**) et un attribut qui représente le solde global (voir **figure 1**).

Toutes les méthodes (déposer, rémunérer et get_solde) sont implémentées dans la classe DossierBancaire.

Pour les deux classes CompteCourant et Compte Epargne, ils sont constitués d'un attribut chacune qui représente leurs soldes actuels, accompagnés de leurs accesseurs et mutateurs.

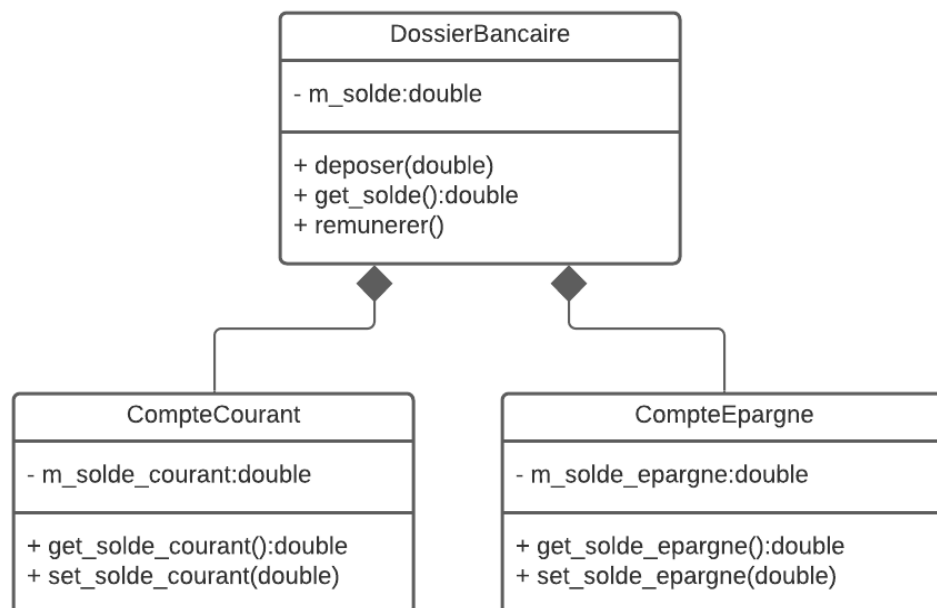


Figure 1: Diagramme de classe

B. Diagramme de séquence

Le diagramme de séquence ci-dessous décrit l'action de l'utilisateur de déposer de l'argent puis de rémunérer son compte épargne et d'afficher son solde également. (voir **figure 2**)

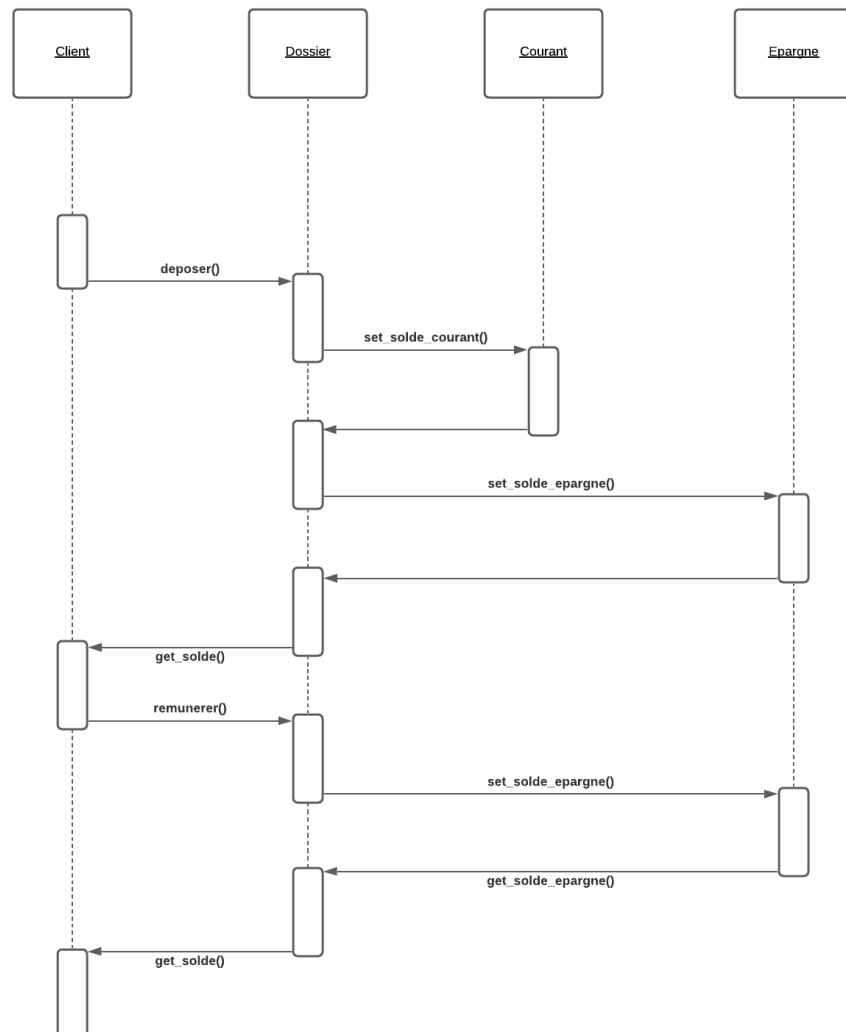


Figure 2: Diagramme de séquence

C. Diagramme de cas d'utilisation

Il s'agit d'un diagramme de cas d'utilisation illustrant les actions qui peuvent être faites par l'utilisateur sur son dossier bancaire (voir **figure 3**).

Un utilisateur peut :

- Déposer de l'argent sur son dossier bancaire (dont 60% sur son compte épargne et 40% sur son compte courant)
- Rémunérer son dossier bancaire (avec un taux de 3.2%)
- Consulter son solde de dossier bancaire.

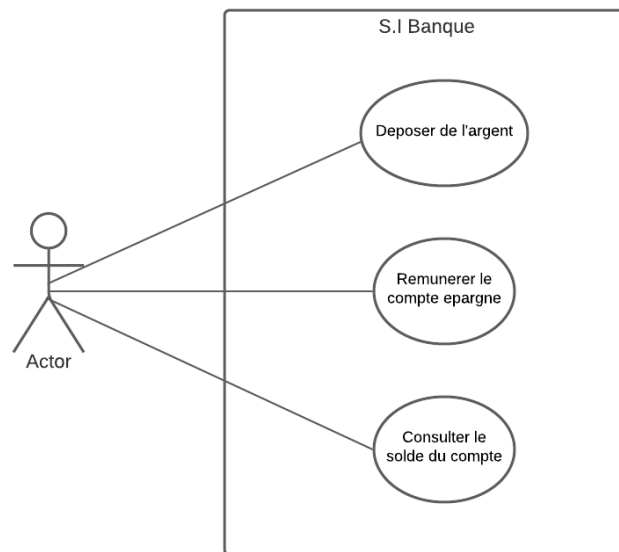


Figure 3: Diagramme de cas d'utilisation

II. Compilation en ligne de commande

Javac est le compilateur de Java, il utilise un fichier source Java fourni en paramètre pour créer un ou plusieurs fichiers contenant le bytecode Java correspondant.

Pour chaque fichier source, un fichier portant le même nom avec l'extension .class est créé si la compilation se déroule bien.

La machine virtuelle Java ou JVM (Java Virtual Machine) est un environnement d'exécution pour applications Java. C'est un des éléments les plus importants de la plate-forme Java. Elle assure l'indépendance du matériel et du système d'exploitation lors de l'exécution des applications Java.

On essaye de faire la compilation sans utiliser un IDE pour mieux comprendre le processus.

Pour la compilation :

javac myPackage/Main.java

Pour l'exécution

java myPackage/Main

```

C:\Users\usrlocal\Desktop\notreprojet\JavaProgSujet\src>javac myPackage/Main.java
C:\Users\usrlocal\Desktop\notreprojet\JavaProgSujet\src>java myPackage/Main
  
```

Figure 4: Compilation et exécution du code

La compilation et l'exécution du code fourni a bien fonctionné, on peut voir notre interface graphique.

```

C:\Users\usrlocal\Desktop\notreprojet\JavaProgSujet\src>javac -cp "C:\Users\usrlocal\...
a tests/MyTest2.java tests/MyTestSuite1.java tests/MyTestSuite1Runner.java

C:\Users\usrlocal\Desktop\notreprojet\JavaProgSujet\src>java -cp "C:\Users\usrlocal\...
Runner
test2_1(tests.MyTest2): Test2_1 Not yet implemented
test2_2(tests.MyTest2): Test2_2 Not yet implemented

C:\Users\usrlocal\Desktop\notreprojet\JavaProgSujet\src>
  
```

Figure 5: Compilation et exécution des tests

La compilation et l'exécution des tests fonctionne aussi.

III. Développement

Création de dépôt git sur Eclipse l'environnement de développement java.
 Eclipse offre la possibilité de faire la gestion des versions à l'aide de Git.
 Un dossier ".git" apparaît dans le répertoire de notre projet

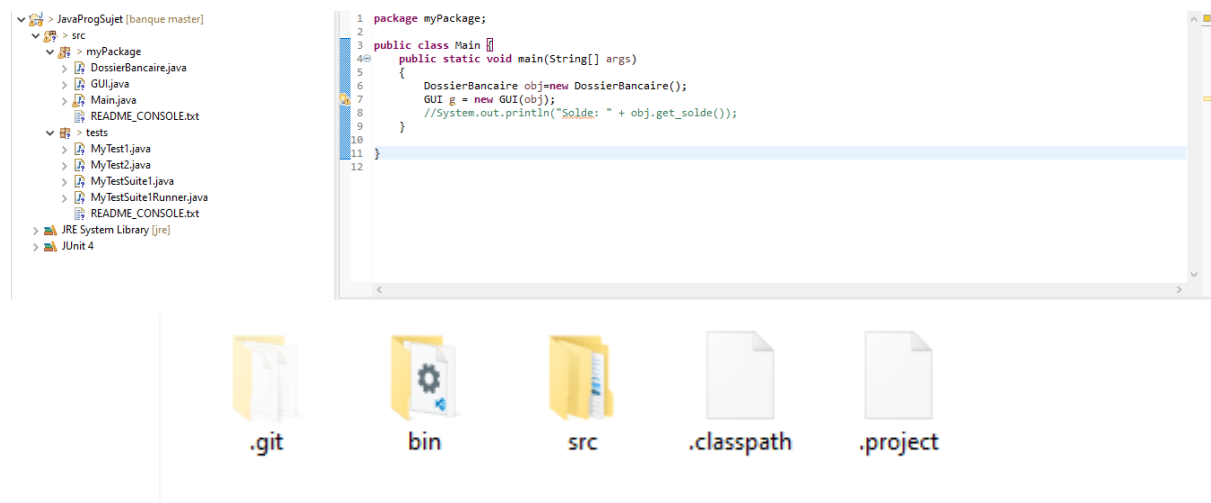


Figure 6: Initialisation du git

On a ajouté les fichiers (incluant les tests) de départ au dépôt git.
 Le premier commit a pour message "premier commit" (voir **Figure 7**)



Figure 7: Premier commit

On a créé une classe de test appelé TestsDossierBancaire qui implemente 3 tests de la classe DossierBancaire.
On a ainsi modifié le nom de la classe test TestSuite1 en TestsSuite, ainsi que son contenu.
Après, on a supprimé toutes les tests inutiles.

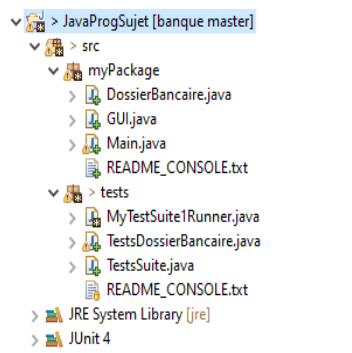


Figure 8: Suppression des tests inutiles

On a intégré les modifications à git et on a créé un tag nommé (V1.0).

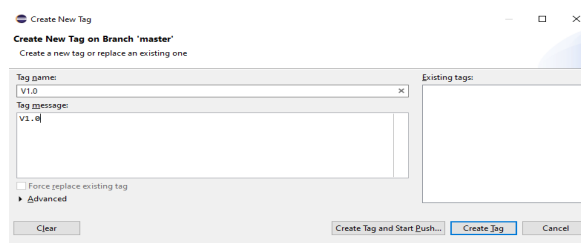


Figure 9: Création du tag

Ajout de la classe CompteCourant et intégration de cette classe au DossierBancaire.
La rémunération ne change pas le solde (méthode n'est pas encore implémenté).

Le dépôt sur le dossier bancaire est intégralement affecté au compte courant. (solde compte courant = solde dossier bancaire).

Ajout des tests unitaires :

On a créé une classe de test appelé TestsCompteCourant qui implémente 2 tests de la classe CompteCourant.

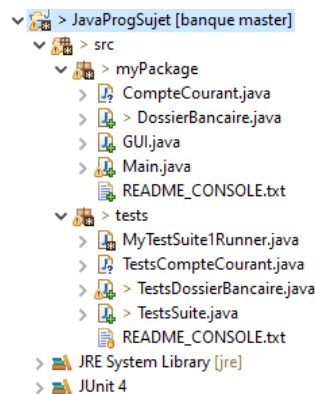


Figure 10: Ajout du TestCompteCourant

Ajout de la classe CompteEpargne et intégration de cette classe au DossierBancaire.

La rémunération doit être implémenté.

Le dépôt sur le DossierBancaire est maintenant répartie sur les 2 classes.

Ajout des tests unitaires :

On a créé une classe de test appelé TestsCompteEpargne qui implémente 2 tests de la classe CompteEpargne.

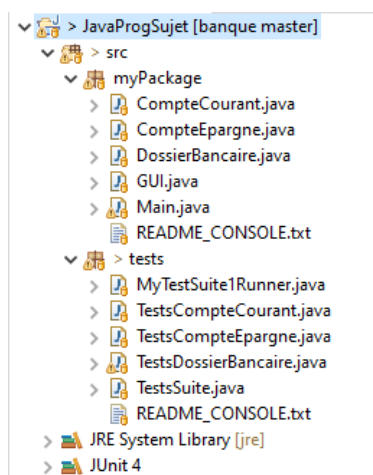


Figure 11: Ajout du TestCompteEpargne

On fait un commit et on tague notre version V2.0

Project: JavaProgSujet [JavaProgSujet]						
Id	Message	Author	Authored Date	Committer	Committed Date	
dfcb331	V2.0 master HEAD Classe Epargne + Tests Epargnes + Correction B	Elas	6 minutes ago	Elas	6 minutes ago	
bd31d47	correction bug TestsSuite	Elas	19 minutes ago	Elas	19 minutes ago	
ea86047	Tests Pour la Classe Compte Courant	Elas	20 minutes ago	Elas	20 minutes ago	
f36582d	Ajout du Classe Courant	Elas	29 minutes ago	Elas	29 minutes ago	
e6b50b5	V1.0 TestsDossierBancaire avec 3 Tests	Elas	32 minutes ago	Elas	32 minutes ago	
41d9ab1	premier commit	Elas	41 minutes ago	Elas	41 minutes ago	

Figure 12: Historique de nos versions

Test du retour la version antérieure 1.0

Project: JavaProgSujet [JavaProgSujet]					
Id	Message	Author	Authored Date	Committer	Committed Date
e6b50b5	V1.0 HEAD TestsDossierBancaire avec 3 Tests	Elas	41 minutes ago	Elas	41 minutes ago
41d9ab1	premier commit	Elas	50 minutes ago	Elas	50 minutes ago

Figure 13: Retour à la version 1.0

Consultation des fichiers après le “checkout “

JavaProgSujet [JavaProgSujet V1.0 e6b50b5]
src
myPackage
DossierBancaire.java
GUI.java
Main.java
README_CONSOLE.txt
tests
TestsDossierBancaire.java
TestsSuite.java
TestsSuiteRunner.java
README_CONSOLE.txt
JRE System Library [jre]
JUnit 4

Figure 14: Consultation des fichiers

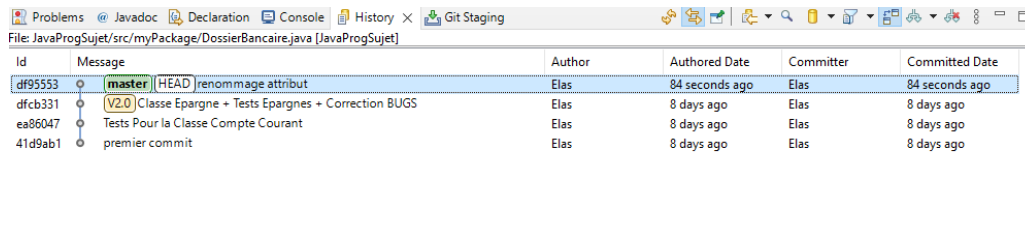
On est revenu à la dernière version (V2.0).

Id	Message	Author	Authored Date	Committer	Committed Date
dfcb331	V2.0 master HEAD Classe Epargne + Tests Epargnes + Correction B	Elas	14 minutes ago	Elas	14 minutes ago
bd31d47	correction bug TestsSuite	Elas	26 minutes ago	Elas	26 minutes ago
ea86047	Tests Pour la Classe Compte Courant	Elas	27 minutes ago	Elas	27 minutes ago
f36582d	Ajout du Classe Courant	Elas	36 minutes ago	Elas	36 minutes ago
e6b50b5	V1.0 TestsDossierBancaire avec 3 Tests	Elas	40 minutes ago	Elas	40 minutes ago
41d9ab1	premier commit	Elas	49 minutes ago	Elas	49 minutes ago

Figure 15: Retour à la version 2.0

IV. Fusion

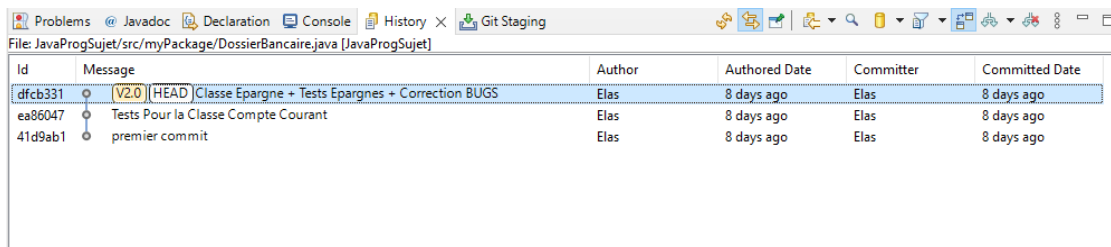
Changement de m_solde vers m_soldeDossierBancaire, puis on effectue un commit.



Id	Message	Author	Authored Date	Committer	Committed Date
df95553	master HEAD renommage attribut	Elas	84 seconds ago	Elas	84 seconds ago
dfcb331	V2.0 Classe Epargne + Tests Epargnes + Correction BUGS	Elas	8 days ago	Elas	8 days ago
ea86047	Tests Pour la Classe Compte Courant	Elas	8 days ago	Elas	8 days ago
41d9ab1	premier commit	Elas	8 days ago	Elas	8 days ago

Figure 16: Modification et Commit

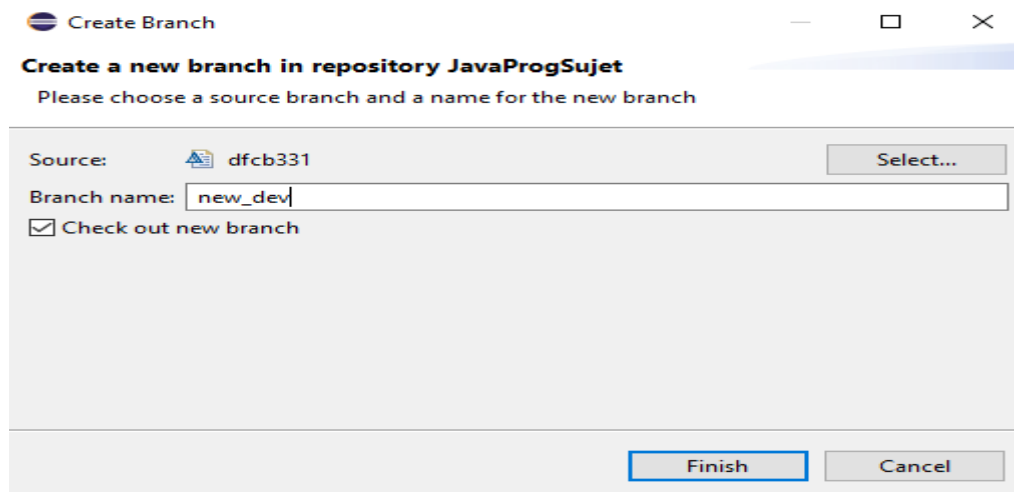
Retour à la version 2.0 puis création d'une nouvelle branche "new_dev"



Id	Message	Author	Authored Date	Committer	Committed Date
dfcb331	V2.0 HEAD Classe Epargne + Tests Epargnes + Correction BUGS	Elas	8 days ago	Elas	8 days ago
ea86047	Tests Pour la Classe Compte Courant	Elas	8 days ago	Elas	8 days ago
41d9ab1	premier commit	Elas	8 days ago	Elas	8 days ago

Figure 17: Retour à la V2.0

Création d'une nouvelle branche appelée "new_dev"



Create Branch

Create a new branch in repository JavaProgSujet

Please choose a source branch and a name for the new branch

Source: dfcb331 Select...

Branch name: new_dev

☒ Check out new branch

Finish Cancel

Figure 18: Nouvelle branch "new_dev"

Maintenant, on a 2 branches, la branche Master et la branche new_dev.

Le diagramme de classe de l'application suite aux modifications (ajout de la classe "Compte" et intégration de l'héritage afin de factoriser).ceci nous a amené bien évidemment à

effectuer des modification sur le code des classes Compte , CompteCourant et CompteEpargne

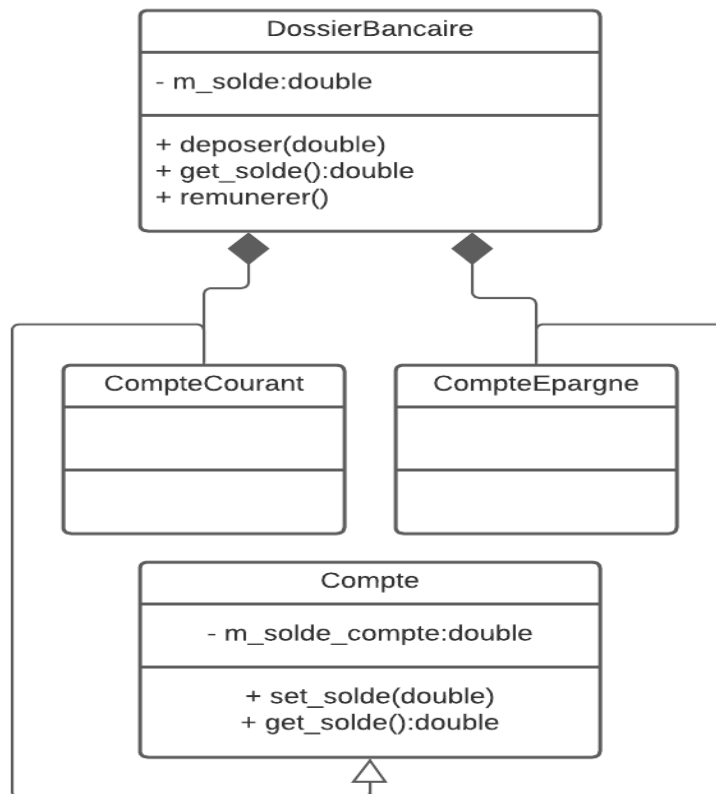


Figure 19: Diagramme de classe amélioré

Après implémentation de ces nouvelles modifications en commitant sur la nouvelle branche. On revient vers la branche master, on fait des petites modifications (ajout de commentaires) et on commit.

Id	Message	Author	Authored Date	Committer	Committed Date
c1dcb18	master (HEAD) Ajout de commentaires	Elas	15 minutes ago	Elas	15 minutes ago
d17da29	new_dev Heritage fini et Tests	Elas	19 minutes ago	Elas	19 minutes ago
4e21332	Classe Compte	Elas	23 minutes ago	Elas	23 minutes ago
df95553	renommage attribut	Elas	84 minutes ago	Elas	84 minutes ago
dfcb331	V2.0 Classe Epargne + Tests Epargnes + Correction BUGS	Elas	8 days ago	Elas	8 days ago
bd31d47	correction bug TestsSuite	Elas	8 days ago	Elas	8 days ago
ea86047	Tests Pour la Classe Compte Courant	Elas	8 days ago	Elas	8 days ago
f36582d	Ajout du Classe Courant	Elas	8 days ago	Elas	8 days ago
e6b50b5	V1.0 TestsDossierBancaire avec 3 Tests	Elas	8 days ago	Elas	8 days ago
41d9ab1	premier commit	Elas	8 days ago	Elas	8 days ago

Figure 20: Historique des versions

Maintenant, on a essayé d'intégrer les modifications de la branche «new_dev», en revenant au préalable sur le «master», on est amené à faire un peu de gestion des conflits.

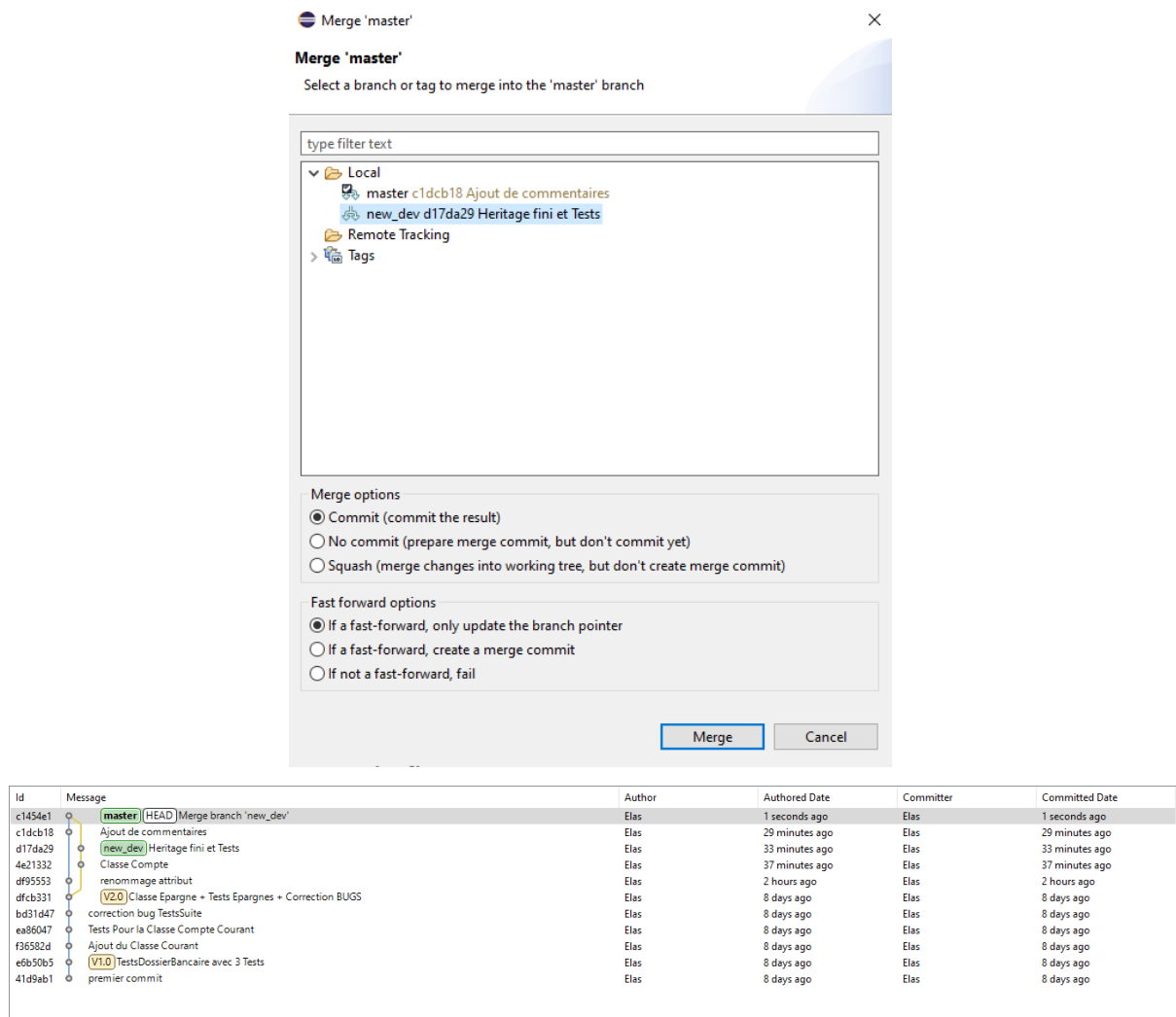


Figure 21: Historique des versions

V. Tests

Après création d'une méthode qui sert à retirer de l'argent du solde, en test si le test unitaire fonctionne bien pour la gestion de l'exception.

```
@Test
public void retirer()
{
    try{
        DossierBancaire dossier=new DossierBancaire();
        dossier.deposer(1000);
        dossier.retirer(500);
    }
    catch(Exception f) {
        assert(f.getMessage().contains("Solde insuffisant"));
    }
}
```

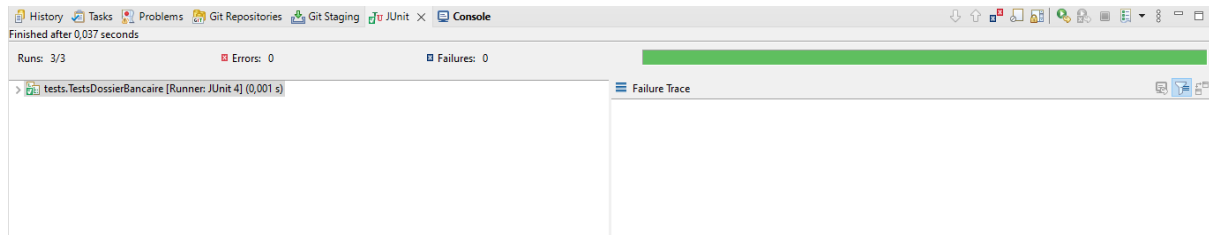


Figure 22: Test unitaire sur TestDossierBancaire

Modification de la classe GUI pour avoir une interface graphique qui donne accès à toutes les fonctionnalités.

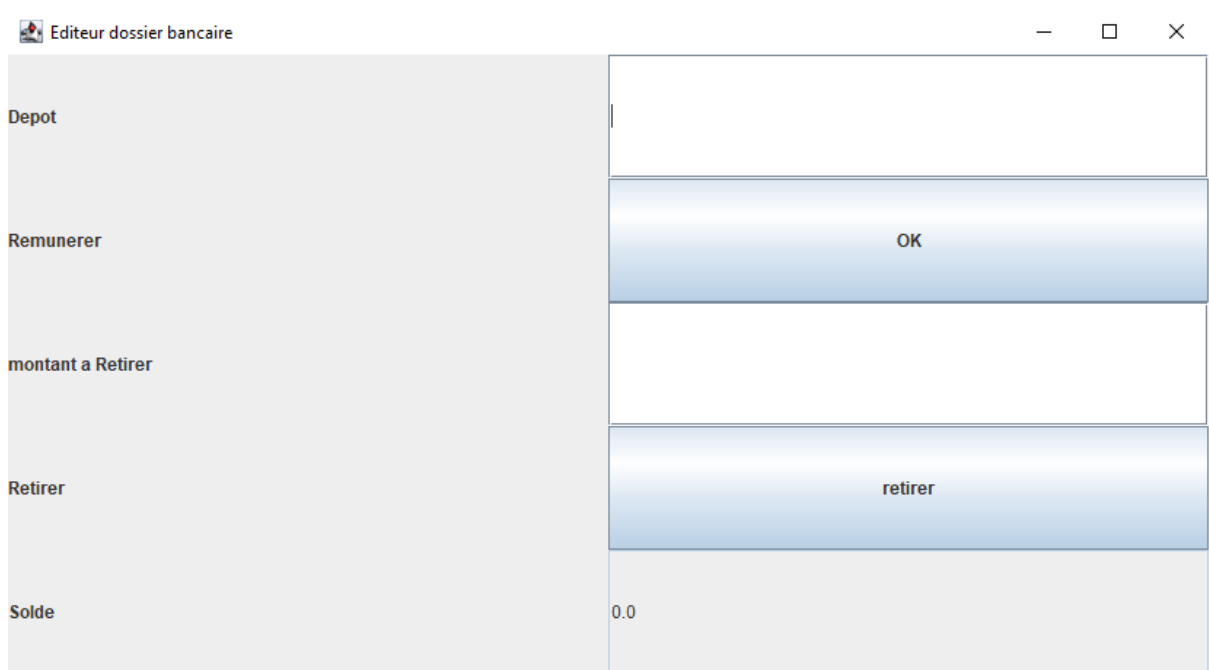


Figure 23: Interface Graphique améliorée

Id	Message	Author	Authored Date	Committer	Committed Date
703b985	V3.0 (master HEAD) Ajout de retirer () et Mise à jour de la GUI	Elas	64 seconds ago	Elas	64 seconds ago
c1454e1	Merge branch 'new_dev'	Elas	44 minutes ago	Elas	44 minutes ago
c1dcb18	Ajout de commentaires	Elas	73 minutes ago	Elas	73 minutes ago
d17da29	new_dev Heritage fini et Tests	Elas	77 minutes ago	Elas	77 minutes ago
4e21332	Classe Compte	Elas	81 minutes ago	Elas	81 minutes ago
df95553	renommage attribut	Elas	2 hours ago	Elas	2 hours ago
dfcb331	V2.0 Classe Epargne + Tests Epargnes + Correction BUGS	Elas	8 days ago	Elas	8 days ago
bd31d47	correction bug TestsSuite	Elas	8 days ago	Elas	8 days ago
ea86047	Tests Pour la Classe Compte Courant	Elas	8 days ago	Elas	8 days ago
f36582d	Ajout du Classe Courant	Elas	8 days ago	Elas	8 days ago
e6b50b5	V1.0 TestsDossierBancaire avec 3 Tests	Elas	8 days ago	Elas	8 days ago
41d9ab1	premier commit	Elas	8 days ago	Elas	8 days ago

Figure 24: Historique des versions

Ce projet nous a permis de mettre en pratique nos compétences acquises en génie logiciel (Test, git sur l'environnement Eclipse) pour mettre en place une application en passant de sa conception sur UML, puis en implémentant sur Eclipse, en gérant les versions et les tests.

Lien vers le projet :

<https://drive.google.com/file/d/1L9Kj0m6RDmYgdS8lpToebBwLWpVCSdq7/view?usp=sharing>