



**POLYTECHNIQUE  
MONTRÉAL**

UNIVERSITÉ  
D'INGÉNIERIE

LOG1000

Ingénierie logicielle Processus de développement  
d'un projet logiciel open source

Travail pratique #4

Polytechnique Montréal Remis le 1 Avril 2019

Trimestre : Hiver 2019

équipe= 32

Équipier1: Sami Belaidi - 1900787

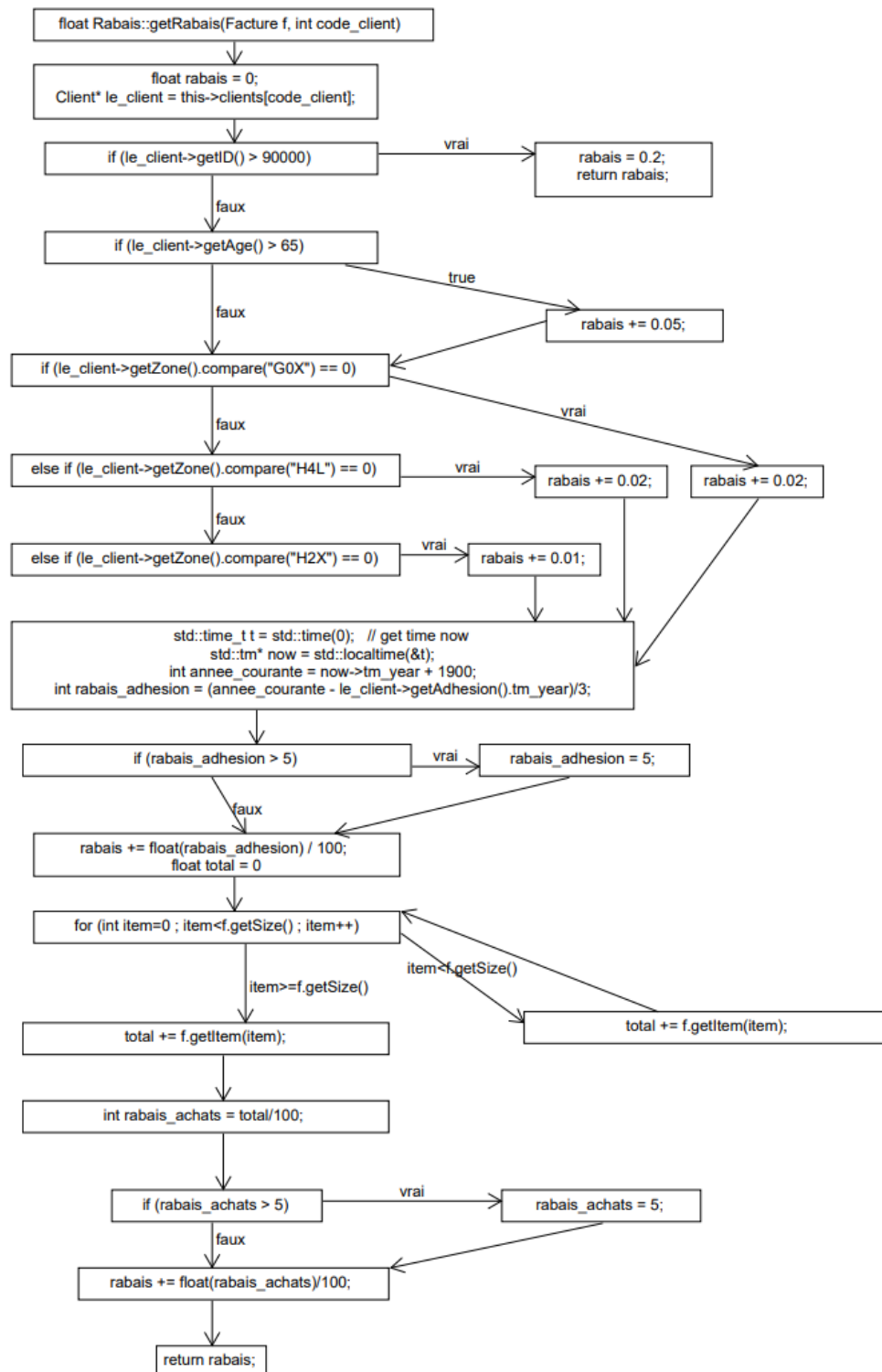
Équipier2: Lucas Simoneau – 1885195

#### 4.1. Exécutez un test unitaire

```
[gigl@fedora-gigl log1000-tp4]$ make test
Makefile:77: warning: overriding recipe for target 'bin/rabais.o'
Makefile:46: warning: ignoring old recipe for target 'bin/rabais.o'
mkdir -p test/bin
g++ -o test/bin/main.o -c test/src/main.cpp
g++ -o test/bin/rabais test.o -c test/src/rabais test.cpp
g++ -o test/bin/typerabais test test/bin/main.o test/bin/rabais test.o bin/
n/rabais.o bin/client.o bin/facture.o bin/afficheur.o -lcppunit
./test/bin/typerabais test
.

OK (1 tests)
```

## 4.2. Rédigez le graphe de flot de contrôle du code



### 4.3. Rédigez les cas de tests pour la couverture des branches

estClient = <{facture = [120], code\_client= 90034, age =25, zone=H1K, tm\_day=15, tm\_mon= 01, tm\_year =2019}, {0.2}>

age = <{facture = [550], code\_client= 14770, age =78, zone=H2H, tm\_day =30, tm\_mon = 01, tm\_year =2010}, {0.13}>

zone\_G0X = <{facture = [100], code\_client= 10456, age =45, zone=G0X, tm\_day = 25, tm\_mon =10, tm\_year =2016}, {0.04}>

zone\_H4L = <{facture = [200], code\_client= 12345, age =60, zone=H4L, tm\_day = 03, tm\_year =01, tm\_year =1998}, {0.08}>

zone\_H2X = <{facture = [300], code\_client= 88876, age =20, zone=H2X, tm\_day = 03, tm\_year =01, tm\_year =2019}, {0.04}>

### 4.4. Codez les cas de tests dans le projet CppUnit

```
[gigl@fedora-gigl log1000-tp4]$ make test
Makefile:77: warning: overriding recipe for target 'bin/rabais.o'
Makefile:46: warning: ignoring old recipe for target 'bin/rabais.o'
mkdir -p test/bin
g++ -o test/bin/main.o -c test/src/main.cpp
g++ -o test/bin/rabais test.o -c test/src/rabais test.cpp
g++ -o test/bin/typerabais test test/bin/main.o test/bin/rabais test.o bin/rabais.o bin/client.o bin/facture.o bin/afficheur.o -lcppunit
./test/bin/typerabais test
.....F.

!!!FAILURES!!!
Test Results:
Run: 6 Failures: 1 Errors: 0

1) test: RabaisTest::test H4L (F) line: 73 test/src/rabais test.cpp
equality assertion failed
- Expected: 0.08
- Actual : 0.09
```

#### **4.5. Trouvez un défaut**

D'après les tests effectués ci-haut, nous voyons qu'il y a un test qui a échoué. Il s'agit du test pour la zone H4L, qui correspond à l'exigence SRS05. Le rabais additionnel pour la zone H4L est de 1%. Or, d'après nos tests, nous obtenons 0.09 au lieu de 0.08, soit une différence de 0.01. Celle-ci vient de la ligne de code suivante :

```
else if (le_client->getZone().compare("H4L") == 0) rabais += 0.02;
```

Afin de respecter les exigences, nous devrions plutôt ajouter 0.01 au rabais. La ligne devrait être :

```
else if (le_client->getZone().compare("H4L") == 0) rabais += 0.02;
```

#### **5. Rétroaction**

1) Nous avons passé 10 heures-personnes. Oui, l'effort est adéquat car nous devons faire des tests et vérifier si nos tests parcourent toutes les branches.

2) Non, nous n'avons pas de recommandations à faire pour améliorer ce travail pratique.