

POLYTECHNIQUE
MONTRÉAL

LE GÉNIE
EN PREMIÈRE CLASSE



Travail pratique #3: Tests

École Polytechnique de Montréal

Trimestre : Automne 2020

Équipier1: Sami Bourai #2041659

Équipier2: Fanilotiana Guy Randriamanalinarivo #2040625

Équipe:312

Présenté à : William Harvey et Nicolas Verbaere

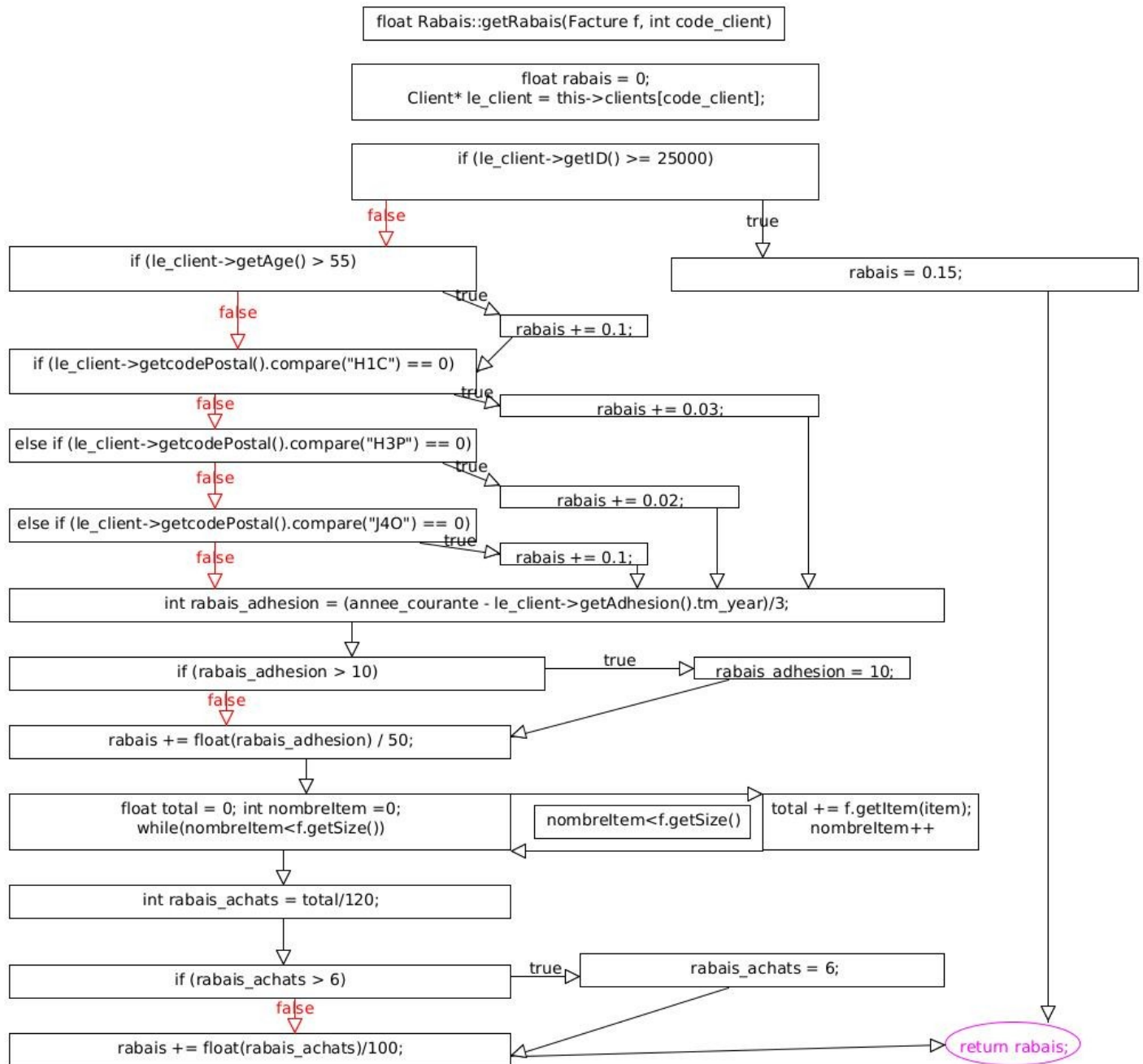
Polytechnique Montréal
Remis le 3 novembre 2020

4.1 – Exécution du test unitaire :

```
[gigl@localhost code]$ make test
Makefile:76: avertissement : surchargement de la recette pour la cible « bin/rabais.o »
Makefile:45: avertissement : ancienne recette ignorée pour la cible « bin/rabais.o »
./test/bin/typerabais_test
.

OK (1 tests)
```

4.2 – Diagramme de flot de contrôle de la fonction getRabais()



4.3- Rédigez les cas de tests pour la couverture des branches :

calcul de la complexité cyclomatique : = 9 → donc on a 9 tests à effectuer minimalement, normalement.

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9
Employée ou Pos	T	F	F	F	F	F	F	F	F
Age > 55	*	T	F	T	F	T	F	T	F
Zone H1C	x	T	T	F	F	F	F	T	F
Zone HEP	*	F	F	T	T	F	F	F	F
Zone J40	*	F	F	F	F	T	T	F	F
Rabais Adhésion > 10	*	T	F	T	F	T	F	T	F
Rabais achat > 6	*	F	T	F	T	F	T	F	T

branche vrai de SRS01/SRS02/SRS03 :

d1 = <{Facture=120.0,client =[27000, ``Tremblay``, ``Joseph``, 60, ``H1C`` 2020] }, 0.15}>

Branche fausse de SRS01/SRS02/SRS03 et donc le reste des exigences:

SRS04-SRS05-SRS06-SR07-SRS08

d2 = <{Facture=120.0,client =[20 000, ``Lionel``, ``Messi``, 57, ``H1C`` 1987] }, 0.35}>

SRS05-SRS08-SRS09

d3 = <{Facture=1200.0,client =[20 000, ``Fani``, ``Ioliana``, 25, ``H1C`` 2020] }, 0.10}>

SRS04-SRS05-SRS06-SRS07-SRS08

d4 = <{Facture=120.0,client =[20 000, ``Cristiano``, ``Ronaldo``, 60, ``H3P`` 1990] }, 0.24}>

SRS05-SRS08-SRS09

d5 = <{Facture=1200.0,client =[20 000, ``Riyad``, ``Mahrez``, 20, ``H3P`` 2020] }, 0.09}>

SRS04-SRS05-SRS06-SRS07-SRS08

d6 = <{Facture=120.0,client =[20 000, ``Antoine``, ``Greizman``, 57, ``J40`` 1986] }, 0.33}>

SRS05-SRS08-SRS09

d7 = <{[Facture=1200.0,client =[20 000, ``Diego``, ``Maradona``, 20, ``J4O`` 2020], 0.08}>

SRS04-SRS06-SRS07-SRS08

d8 = <{[Facture=120.0,client =[20 000, ``Antoine``, ``Greizman``, 60, ``H1G`` 1987]], 0.31}>

SRS06-SRS08-SRS09

d9 = <{[Facture=1200.0,client =[20 000, ``Jack``, ``Graelish `` , 20, ``H1H`` 2020], 0.06}>

Tous les tests passent à part le d4.

4.4 - Codez les cas de tests dans le projet CppUnit :

4.5 -Trouvez un défaut :

```
int annee_courante = now().tm_year + 1900;

// 2% de rabais par trois années, maximum 10%
int rabais_adhesion = (annee_courante - le_client->getAdhesion().tm_year)/3;
if (rabais_adhesion > 10) rabais_adhesion = 10;
rabais += float(rabais_adhesion) / 50;

// Rabais basé sur le total de la facture
```

Pour d4 nous avons testé avec 1990 comme années d'adhésion. Comme le client a 30 ans de fidélité, il devrait au maximum avoir 10% de rabais automatiquement. Donc on fait juste additionner 0.1 à rabais. Or, ce n'est pas le résultat obtenu.

$30/3 = 10$

$10! > 10 \rightarrow \text{rabais} += 10/50 = 0.2.$

or 0.2 représente 20% de rabais et non le 10% que c'est supposé.

Voici la correction :

```
afficheur.h      108 // 2% de rabais par trois années, maximum 10%
client.cpp       109 int rabais_adhesion = (annee_courante - le_client->getAdhesion().tm_year)/3;
client.h         110 if ((float(rabais_adhesion) / 50)>0.1)
facture.cpp      111 {
facture.h        112     rabais+=0.1;
main.cpp         113 }else{
rabais.cpp       114     rabais += float(rabais_adhesion) / 50;
rabais.cpp       115 }
```

4.6 – Rétroaction :

On a passer 12 h dessus par personne, mais ou moins, malgré toutes les difficulté qu'on a eu par rapport au Makefile, on a réussi à faire le 100% du rapport. Je pense qu'après ce TP on est prêt a rédiger nos propres tests lorsque nous programmons.

Quant aux recommandations, on aurait aimé trouver moins de difficulté par rapport au Makefile. Je pense que vu que le TP vise les tests on devrait pas investir de temps sur le Makefile malgré que cela nous permet d'apprendre.