



PROG-TP1 – Le test

Utilisation de l’outil JUnit

UV1 – INF301

2^e semestre 2015

M.T. SEGARRA, I. REBAÏ, F. DAGNAT, J. MALLET

Objectifs

À la fin de la séance les élèves devront être capables de :

- expliquer l’intérêt de l’utilisation d’un outil d’automatisation de tests ;
- produire pour une classe Java un jeu de tests unitaires ayant une couverture (semi-)complète (cohérence de la classe : comportement normal des méthodes de la classe ; le retour d’erreur : exception ; et le comportement limite des méthodes : valeur extrême, temps de réponse ; etc.) ;
- écrire, en code JUnit, un jeu de tests unitaires à couverture (semi-)complète pour une classe Java.

Pré-requis

Avant de réaliser ce TP vous devez avoir compris le principe de fonctionnement des exceptions en Java. Sur le site du zéro (<http://openclassrooms.com/courses/apprenez-a-programmer-en-java/les-exceptions>) vous trouverez les informations nécessaires.

Ressources

Voici quelques ressources qui vous aideront à comprendre JUnit et son utilisation :

- Le blog de Lars Vogel : <http://www.vogella.com/articles/JUnit/article.html>
- Sur le site développez.com, vous trouverez une bonne adresse pour comprendre JUnit 4 : <http://rpouiller.developpez.com/tutoriels/java/tests-unitaires-junit4>

- Et une présentation très succincte de JUnit pour ceux qui sont pressés : <http://www.cavdar.net/2008/07/21/junit-4-in-60-seconds/>

Découverte de JUnit

JUnit¹ est un cadriciel (« *framework* »), c'est-à-dire un ensemble de classes, visant l'automatisation des tests unitaires en Java. Le principe de base de l'utilisation de cet outil est très simple : on définit un ensemble de tests, l'outil enchaîne automatiquement tous ces tests et met en évidence les tests qui ont échoué. Le test d'une méthode consiste typiquement à comparer son résultat à la valeur attendue (constante ou calculée).

L'usage veut que les tests ne soient pas mélangés avec le code et qu'ils soient définis dans des classes spécifiques. Des conventions de nommage peuvent être adoptées pour bien distinguer les tests du code. Par exemple, pour chaque classe à tester, on définit une classe de test correspondante, dont le nom est celui de la classe à tester suffixé par « **Test** »². Ainsi, la classe de test d'une classe **A** se nommera **ATest**.

Cette classe de test pourra contenir autant de tests que nécessaire sous la forme de méthodes publiques, non statiques, sans paramètre et qui sont annotées par **@Test**. Vous trouverez des exemples de classes de test dans les sources du projet de fil rouge qui vous ont été fournis dans la première séance dédiée au projet. Lancez Eclipse.

Exercice 1 (*Prise de contact*)

JUnit est intégré dans Eclipse. Son principe d'utilisation est le suivant :

- Pour créer une classe de test, on sélectionne la classe à tester et, à l'aide du menu contextuel (bouton droite de la souris), on sélectionne *New -> JUnit Test Case*. Il vous propose alors de générer la classe de test selon la version de JUnit souhaitée, ici nous utiliserons la v4. Ensuite, il vous demande de sélectionner les méthodes que vous voulez tester³. Eclipse ne peut générer qu'un squelette de la classe de test. À vous de le compléter avec votre jeu de test.
- Pour lancer un test, il suffit de sélectionner la classe de test et de lancer (par le menu contextuel) l'exécution (menu *Run as -> JUnit test*). Une fenêtre s'affiche alors vous présentant les résultats de votre test (voir figure 1).
- Pour relancer un test, il suffit d'utiliser le bouton de lancement (rond vert) de cette fenêtre de résultat.

Vous trouverez sur Moodle une archive contenant un projet Eclipse qui représente une version très simplifiée d'un logiciel de gestion d'une bibliothèque (abonnement/désabonnement, emprunt de livres, retour de livres).

- ▷ **Exécutez la classe de test de la classe **Book** qui représente un ouvrage de la bibliothèque. Il s'agit de la classe **BookTest** qui se trouve dans le paquetage **fr.tb.uv1.tests.unit**.**

1. Le code de JUnit est téléchargeable sur le site www.junit.org sous forme d'une archive zip qui contient entre autre **.junit.jar**. Ce TP a été testé avec la version 4.8.1 de JUnit.

2. Dans les versions 3.x de JUnit cette convention était obligatoire mais elle ne l'est plus. De plus, les méthodes de test devaient avoir un nom qui commençait par **test**.

3. Selon la configuration d'Eclipse, une fenêtre peut s'afficher ensuite et vous demande d'ajouter la bibliothèque JUnit v4 dans votre projet. Répondez oui à la question.

Remarque :

*Il est également possible de lancer l'exécution des tests indépendamment d'Eclipse. Pour cela, il faut que les classes de la bibliothèque JUnit soient accessibles au compilateur et à la machine virtuelle (par le **CLASSPATH**). Une fois compilée, une classe de test peut être exécutée en version non graphique :*

```
java org.junit.runner.JUnitCore ClasseTest
```

N'hésitez pas à visiter le site www.junit.org qui contient de nombreux documents pédagogiques sur le test en général et sur l'utilisation de JUnit en particulier.

Exercice 2 (*Le test des ouvrages*)

Nous allons étudier la classe `BookTest` pour une première prise en main de JUnit. Cette classe teste le fonctionnement des méthodes de la classe `Book`.

▷ **Question 2.1 :**

La méthode `testEqualsObject` teste le fonctionnement de la méthode `equals` de la classe `Book`. En analysant le code de cette méthode, donnez la notion d'égalité entre ouvrages.

▷ **Question 2.2 :**

Changez le paramètre `authors` dans la première ligne de la méthode `testOKBook` de la classe `BookTest` par `null`. Quel est le résultat affiché par JUnit ?

▷ **Question 2.3 :**

Rétablissez le paramètre `authors` de la question précédente. Modifiez maintenant la ligne `assertTrue` de la méthode `testLendBook` de manière à indiquer que le livre n'a pas été emprunté. Quel est le résultat affiché par JUnit ?

▷ **Question 2.4 :**

Pourquoi le test de la création d'ouvrages est réalisé dans deux méthodes différentes ? Quelles sont ces méthodes ?

Une classe de test est une classe comme les autres et peut donc définir et utiliser des attributs. On peut ainsi définir des attributs pour des données souvent utilisées dans les différentes méthodes de tests. Par exemple, si un jeu de test est partagé, ces valeurs seront des attributs. Avant chaque exécution d'une méthode de test, JUnit exécute les méthodes annotées par `@Before`. On peut donc utiliser ces méthodes pour réinitialiser des valeurs si cela s'avère nécessaire.

▷ **Question 2.5 :**

Quelle est l'utilité des attributs de la classe `BookTest` ? Où sont-ils initialisés ?

Remarque :

N'hésitez pas à consulter sur <http://kentbeck.github.com/junit/javadoc/latest/> la documentation Javadoc de JUnit.

Tests unitaires d'une application

Maintenant que les principes de base de JUnit sont acquis, vous allez utiliser cet outil pour tester le bon fonctionnement d'un logiciel.

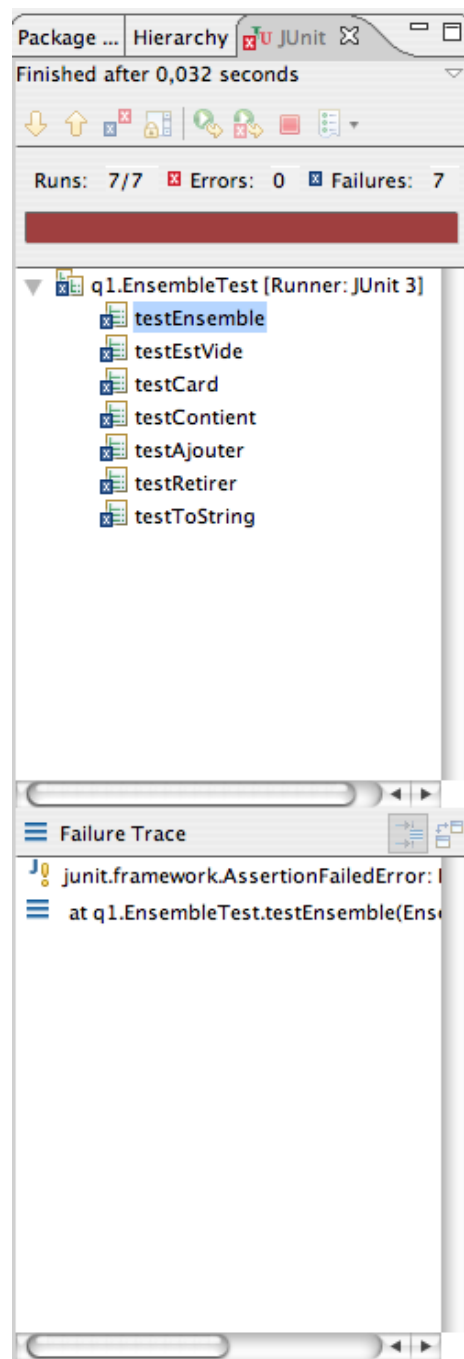


FIGURE 1 – La fenêtre de résultat d'Eclipse

L'objectif est de tester (tests unitaires) une classe Java. À la fin de la séance, si vous avez le temps, vous pourrez vérifier si votre classe passe des tests fournis par les enseignants.

Exercice 3 (*Mise en application sur le calcul de l'IMC*)

L'indice de masse corporelle (IMC ; en anglais, BMI : « *Body Mass Index* ») est une grandeur qui permet d'estimer la corpulence d'une personne et des risques éventuels sur la santé. Cet indice se calcule en fonction de la taille et de la masse de la personne. Selon l'Organisation Mondiale de la Santé (OMS), la formule utilisée pour le calcul de l'IMC est la suivante :

$$\text{IMC} = \frac{\text{masse}}{\text{taille}^2}$$

L'OMS a défini cet indice de masse corporelle comme un standard pour évaluer les risques liés au surpoids chez l'adulte. Il a également défini des intervalles standards (maigreur, indice normal, surpoids, obésité ...) en se basant sur la relation constatée statistiquement entre l'IMC et le taux de mortalité. Le tableau suivant illustre cette classification.

IMC ($kg.m^{-2}$)	Interprétation
moins de 16,5	dénutrition ou famine
16,5 à 18,5	maigreur
18,5 à 25	corpulence normale
25 à 30	surpoids
30 à 35	obésité modérée
35 à 40	obésité sévère
plus de 40	obésité morbide ou massive

Bien qu'il fût conçu initialement pour les adultes de 18 à 65 ans, de nouveaux diagrammes de croissance ont vu le jour au cours des dernières décennies pour les enfants de 0 à 18 ans. Il ne peut pas non plus être utilisé tel quel pour les femmes enceintes ou qui allaitent, les athlètes d'endurance ou les personnes très musclées⁴.

▷ Développez et testez le bon fonctionnement d'une classe **Person** qui doit permettre de :

- calculer l'IMC de la personne ;
- retourner sa corpulence ;
- afficher un avis expert personnalisé (en utilisant le nom, le prénom de la personne et les recommandations du tableau donné en annexe).

Le diagramme de classes disponible figure 2 représente la classe **Person** que vous devez développer et une classe utilitaire **Corpulence**.

Attention, les *getters* et *setters* ne sont pas représentés dans le diagramme.

Pour vous faciliter le travail, vous trouverez sur Moodle un projet Eclipse contenant le code de la classe **Corpulence** ainsi que les deux exceptions qui doivent être levées par la classe **Person**. Téléchargez ce projet et ouvrez-le avec Eclipse. La classe **Person** doit être créée dans le paquetage `junit.bmiPerson`.

Nous considérons que :

⁴. Le texte de cette partie est largement inspiré de la page Wikipédia http://fr.wikipedia.org/wiki/Indice_de_masse_corporelle.

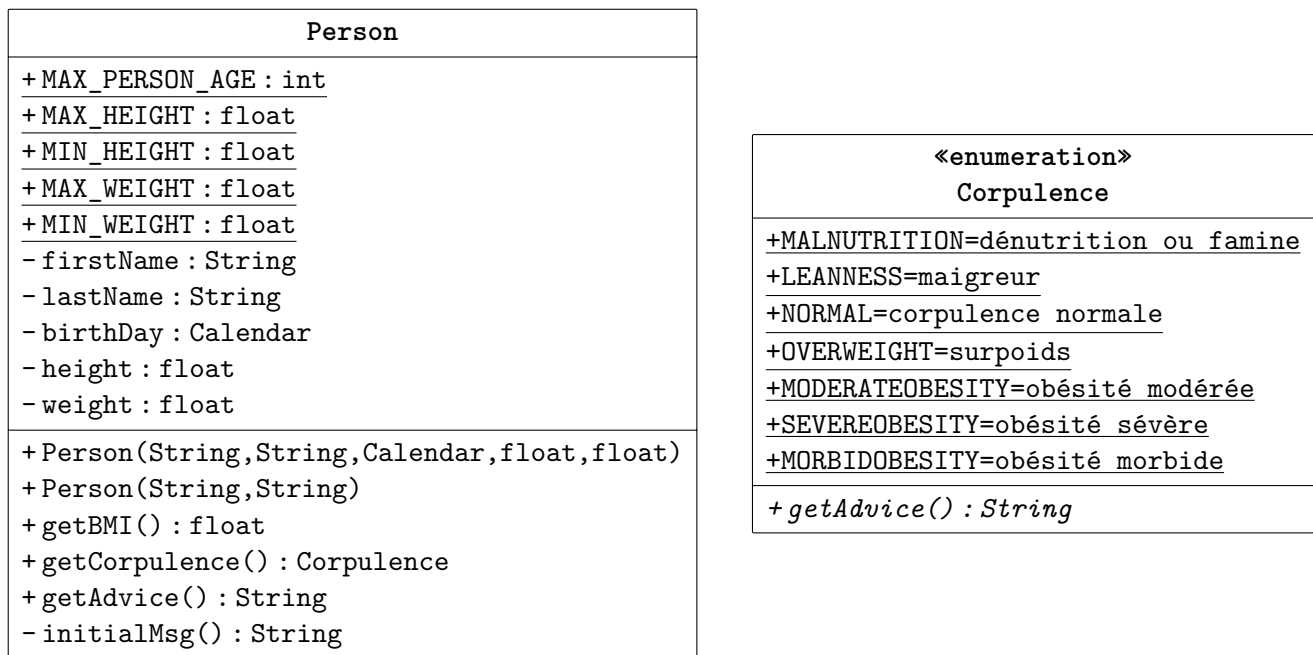


FIGURE 2 – Diagramme de classes de l'application à développer et tester.

- l'âge maximale d'une personne est de 150 ans ;
- il n'y a pas d'âge minimale, mais la personne doit être né avant la date courante ;
- la taille maximale pour une personne est de 200.75 cm et la minimale est de 35 cm ;
- le poids maximal est de 720 kg et le minimum est de 1 kg.

Si ces contraintes ne sont pas respectées, la classe lève une exception `BadParametersException`, la même que lorsque le nom ou le prénom de la personne sont « vides ».

La méthode `getBMI()` lève une exception `MissingParametersException` si la taille ou le poids ne sont pas définis pour la personne. Elle est utilisée par les méthodes `getCorpulence` et `getAdvice`.

Le premier constructeur a 5 paramètres dans l'ordre suivant : prénom, nom, date de naissance, taille et poids. Pour le deuxième constructeur, les paramètres sont prénom et nom.

Enfin, la méthode `initialMsg()` rend la chaîne de caractères existant entre `<poids>` et `<taille>` dans le tableau de recommandations donnés en annexe. Il est donc différent en fonction de la corpulence de la personne.

Annexes

A Tableau de recommandations

Ces recommandations sont disponibles en mode texte à l'adresse <http://www.doctissimo.fr>.

IMC	Message
moins de 18,5	<prénom> <nom>, votre poids <poids> apparaît insuffisant au regard de votre taille <taille>. Cette maigreur excessive peut être la conséquence de maladies, de troubles psychologiques et elle peut aussi être elle-même à l'origine d'autres maladies entraînant ainsi un cercle vicieux. Il est utile que vous consultiez un médecin afin qu'il constate l'ampleur de l'insuffisance pondérale et en recherche la cause éventuelle. Il pourra également vous faire une proposition de traitement.
moins de 25	<prénom> <nom>, votre poids <poids> est idéal compte tenu de votre taille <taille>. Une alimentation équilibrée, sans excès de matières grasses, et une activité physique régulière comme la marche, la natation ou le vélo vous aideront à maintenir votre poids idéal.
moins de 30	<prénom> <nom>, votre poids <poids> apparaît modérément excessif compte tenu de votre taille <taille>. Ce surpoids peut augmenter votre risque de maladies, en particulier votre risque de troubles cardiaques ou vasculaires et votre risque de diabète. Ce risque peut se cumuler avec d'autres risques provoqués par le tabagisme, l'hypertension ou le cholestérol. Si vous présentez déjà un autre facteur de risque, une diminution de poids vous sera bénéfique. Une alimentation moins riche en graisses, avec d'avantage de fruits et de légumes, et une activité physique régulière vous permettront sans doute de retrouver un poids idéal. Votre médecin sera le mieux à même de vous donner les conseils adaptés à cet objectif.
moins de 40	<prénom> <nom>, votre poids <poids> apparaît excessif compte tenu de votre taille <taille>. Ce surpoids peut augmenter votre risque de maladies, en particulier de troubles cardiaques ou vasculaires et de diabète. Ce risque peut se cumuler avec d'autres risques provoqués par le tabagisme, l'hypertension ou le cholestérol. Dans tous les cas, une diminution de poids vous sera bénéfique. Une alimentation moins riche en graisses, avec d'avantage de fruits et de légumes, et une activité physique régulière vous permettront sans doute de retrouver un poids idéal. Votre médecin sera le mieux à même de vous donner les conseils qui vous aideront à atteindre cet objectif.

IMC	Message
plus de 40	<prénom> <nom>, votre poids <poids> est beaucoup trop important compte tenu de votre taille <taille>. Vous souffrez d'obésité et cela signifie que vous êtes exposé à un risque non négligeable de contracter des maladies cardiaques (infarctus), vasculaires (accident vasculaire cérébral, insuffisance veineuse) ou métaboliques (diabète). Cette obésité peut aussi être à l'origine d'essoufflement, de fatigue, de douleurs dorsales ou articulaires et difficultés psychologiques qui perturbent sérieusement vos activités quotidiennes. Heureusement, même une perte de poids modérée (5 à 10%) peut avoir un effet positif sur votre santé et votre mental, à condition bien sûr de ne pas reprendre les kilos perdus. Dans cette optique, il serait judicieux de pratiquer tous les jours un peu plus d'activité physique et de réduire la part des graisses dans votre alimentation. En tout cas, une consultation avec votre médecin s'impose, pour qu'il fasse le bilan de votre maladie et envisage avec vous les méthodes de perte de poids, ainsi que les éventuels traitements possibles. Les associations de patients peuvent aussi vous fournir des conseils et une aide psychologique précieuse.

B Pour passer les tests fournis par les enseignants

À la fin de la séance, vous pouvez passer les tests unitaires préparés par les enseignants pour la classe `Person`. Pour ce faire, vous devrez récupérer sur Moodle le fichier `PersonTest.class` disponible pour la séance. Ce fichier contient des tests unitaires de la classe `Person`.

Pour l'exécuter, il faut le copier dans le répertoire `bin/junit/bmiPerson/tests` de votre projet, exécuter la commande `SETUP JUNIT`, vous placer dans le répertoire `bin` de votre projet puis exécuter la commande :

```
java org.junit.runner.JUnitCore junit.bmiPerson.tests.PersonTest
```

Si aucune exception n'est levée votre classe satisfait les spécifications.