
Open Week 2016

— Introduction et présentation —

Objectifs de la semaine

- Changer la manière d'apprendre le Java en BAC 1
- Maîtriser l'outil INGINious
- Faire un pas dans l'univers de l'Open Source
- S'impliquer dans un projet de long terme
- Apprendre
- Passer un bon moment

OpenWeek : Mission

- Permettre aux débutants de s'exercer fréquemment
- Donner aux étudiants le goût de la contribution

OpenWeek : Vision

- Un projet Open Source vivant, auquel plusieurs étudiants prennent part
- Un grand nombre d'exercices différents pour s'adapter totalement au niveau de l'étudiant

Déroulement de la semaine

- Petit déjeuner le matin
- Contribution au projet durant l'avant et l'après midi
- Présentations ou activités chaque jour
- Repas au RestoU à midi
- Activités de détente lorsque vous le désirez

*Les salles informatiques viennent d'être refaites, s'il vous plaît **ne mangez/buvez pas en salle** (il y a un local sur le côté disponible pour cela)*

INGInious

“INGInious is an intelligent grader that allows secured and automated testing of code made by students. ”

*“INGInious is a FOSS, feel completely free to help us improving it!
We accept pull requests, correct issues and are glad to talk about
implementation details onGitter!”*

G. Derval, A. Géo

Tests unitaires

“Program testing can be used to show the presence of bugs, but never to show their absence.”

Dijkstra

“Aucune question relative aux programmes, vue sous l’angle de la fonction qu’ils calculent, ne peut être décidée par l’application d’un algorithme.”

Yves Deville (découle du théorème de Rice)

Tests unitaires

Les tests unitaires ont cependant de vrais avantages en pratique :

- Programmes légers à exécuter
- Faciles à écrire
- Peuvent donner un bon résultat quant à la justesse d'un code
- Ne demandent pas l'utilisation d'invariants

Framework

Installer :

- JUnit
- Python
- Java

Ou alors :

Télécharger la vm, décompresser, lancer

“vagrant box add package.box --name openweek”, puis
“vagrant up” pour initialiser et démarrer la vm.

Un test de base : Java

- Un fichier qui contient la méthode main et les tests, par exemple : Exercice1.java
- Un fichier qui contient la réponse à l'exercice. Il **doit** se nommer Exercice1Stu.java (la classe s'appelle donc *Exercice1Stu*). Fichier inutile pour INGINious mais utile pour lancer les tests rapidement sur votre ordinateur
- Une copie de ce fichier, mais la partie que l'étudiant doit écrire a été remplacée par "@@q1@@". Le fichier **doit** se nommer Exercice1Vide.java (la classe du fichier est donc quand même *Exercice1Stu*)
- Allez voir sur le [Github](#) du projet, tout y est expliqué.

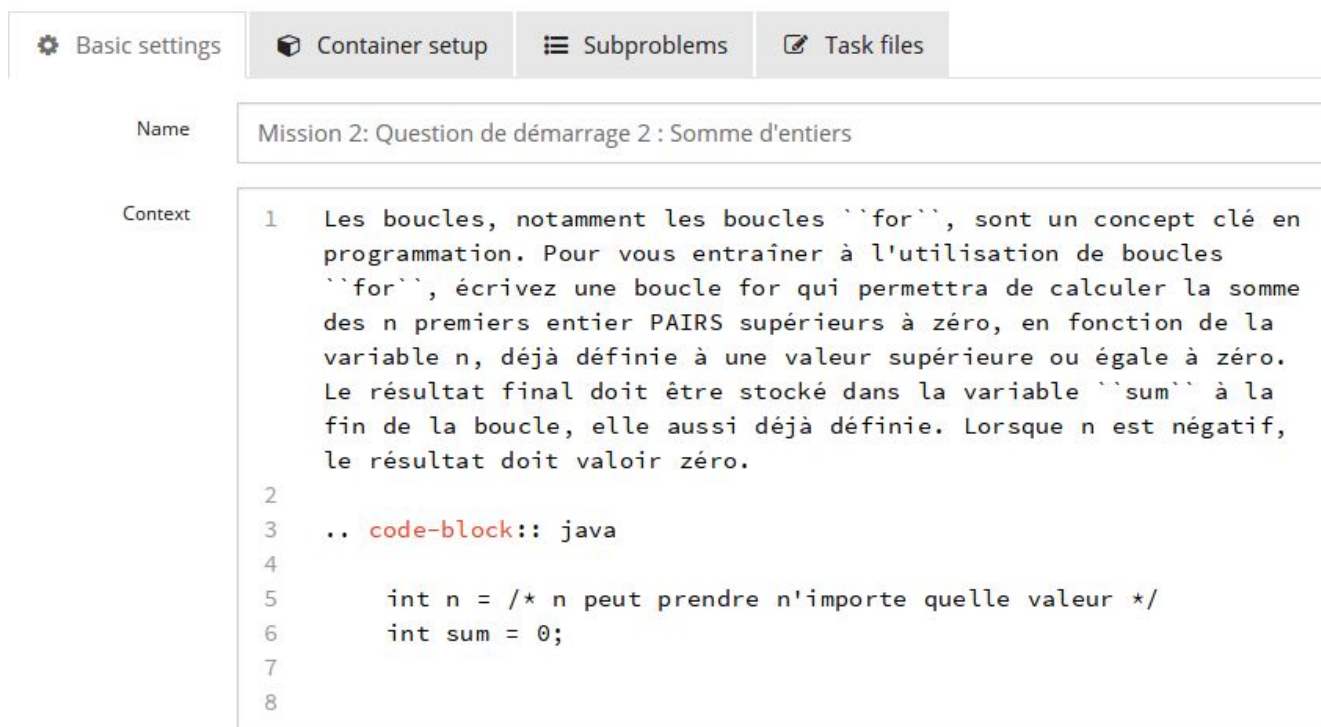
Le fichier run d'INGInious

C'est le fichier qui *exécute* les tests sur le serveur d'INGInious. Un template est disponible sur le Github du projet, il n'y a que quelques variables à remplacer :

- EXERCICE = "Exercice1" (on le remplace par le nom de la classe de la méthode main et les tests)
- EXECCUSTOM = 0 (vaut 1 si vous voulez exécuter un script que vous appellerez "custom.sh" avant de lancer les tests. On conseille de la mettre à 0 en général)
- NEXERCICES = 1 (vaut le nombre d'exercices de la tâche INGInious (en général un seul))

Le fichier task.yaml

Vous pouvez directement configurer la tâche sur votre instance d'INGInious qui tourne sur votre vm, pour récupérer un task.yaml bien formaté.



The screenshot shows the 'Task files' tab in the INGInious configuration interface. The 'Name' field is 'Mission 2: Question de démarrage 2 : Somme d'entiers'. The 'Context' field contains a description of the task and a code block for a Java solution.

Basic settings Container setup Subproblems Task files

Name: Mission 2: Question de démarrage 2 : Somme d'entiers

Context:

```
1 Les boucles, notamment les boucles ``for``, sont un concept clé en
  programmation. Pour vous entraîner à l'utilisation de boucles
  ``for``, écrivez une boucle for qui permettra de calculer la somme
  des n premiers entier PAIRS supérieurs à zéro, en fonction de la
  variable n, déjà définie à une valeur supérieure ou égale à zéro.
  Le résultat final doit être stocké dans la variable ``sum`` à la
  fin de la boucle, elle aussi déjà définie. Lorsque n est négatif,
  le résultat doit valoir zéro.
2
3 .. code-block:: java
4
5     int n = /* n peut prendre n'importe quelle valeur */
6     int sum = 0;
7
8
```

Le fichier task.yaml

Vous pouvez directement configurer la tâche sur votre instance d'INGInious qui tourne sur votre vm, pour récupérer un task.yaml bien formaté.

⚙ Basic settings 📦 Container setup ☰ Subproblems 📄 Task files

Problem id: q1

Name

Somme des n premiers entiers pairs

Context

```
1  Écrivez ici le bout de programme permettant de calculer la somme
    des n premiers entiers pairs :
```

Language

Java

Type

Multiline code ▼

Optional?

☐

Le fichier task.yaml

Les points importants du fichier :

- environment: java7
- limits:
 - memory: '200'
 - time: '30'
 - output: '2'
- problems:
 - q1:
 - language: Java
 - type: code

Le dossier de la tâche

Voici l'architecture des dossiers de la tâche INGINIOUS :

dossier_de_la_tache

| student

| | Exercice1.java

| | Exercice1Vide.java

| run

| task.yaml

Bonnes pratiques

Pour les exercices du début du cours, récupérez toutes les exceptions et transmettez les messages dans un langage compréhensible par les étudiants (ex:ArithmeticException)

There are some errors in your answer. Your score is 0.0%

Vous n'avez pas réussi tous les exercices



Plus grand diviseur

Il semble que vous ayez fait des erreurs dans votre code...



testBiggestDivisor(student.M1Q10): Le code est incorrect : il est interdit de diviser par zéro.

Écrivez ici le bout de programme permettant de calculer le plus grand diviseur entier de n :

```
1  n = 42/0;
```

Bonnes pratiques

Le projet est Open Source, donc spécifiez bien toutes vos méthodes pour que les futurs contributeurs puissent comprendre ce que vous faites.


```
/**
 * @pre -
 * @post Vérifie si le code de l'étudiant gère bien le cas d'une IOException et si -1 est bien retourné dans ce cas.
 * Lance une AssertionError lorsqu'une réponse est incorrecte.
 */
@Test
public void testIOException(){
```


Bonnes pratiques

Vous pouvez reprendre la méthode `main()` générique, déjà utilisée pour tous les tests déjà présents et rajouter vos tests dans le fichier avec la balise `@Test` de JUnit pour qu'ils soient exécutés par la méthode.

```
public static void main(String[] args) {  
    Result result = JUnitCore.runClasses(M1Q7.class);  
    for (Failure failure: result.getFailures()) {  
        System.err.println(failure.toString());  
    }  
    if (result.wasSuccessful()) {  
        System.out.println("Tous les tests se sont passés sans encombre");  
        System.exit(127);  
    }  
}
```

Merci d'avoir suivi

Si vous vous sentez un peu  après cette présentation, n'oubliez pas que les slides sont disponibles sur le GitHub du projet et que je suis toujours présent si vous avez une question ou si vous avez besoin d'un coup de main.

Licenses Open Source

Obligations:

Can I distribute licensed software (modified or unmodified) that has been combined or linked with code covered by another licensing model?

Maybe

Any software that contains GPL code or is derived from GPL code must be licensed as a whole under the GPL terms. What this means is that in order to distribute software that has combined or linked GPL code with non-GPL code, the licenses must be compatible. For example, GPL v.2 is not compatible with GPL v.3. (See <http://www.fsf.org/licenses/licenses>)

The GPL does not explicitly state that linked files create a work derived from the GPL code. However, it is generally understood that static linking, which modifies the code of one program, creates a derivative work and therefore must be licensed under the GPL. It is less clear whether or not dynamic linking creates a derivative work. Dynamic linking does not necessarily modify any code. As this issue has not been litigated, it might be prudent to assume that under the GPL, statically or dynamically linked files both create derivative works.

GPL v.2

(Can I use the licensed software as part of a technological measure - continued)

The GPL v.3 does not stipulate what you can and cannot program. However, it does state that the licensed software shall not be deemed part of an effective technological measure. When you distribute the licensed work, you waive any legal power to forbid circumvention of the technological measures.

Maybe:

Please see the above explanation for GPL v.2 and refer to <http://www.fsf.org/licenses/licenses> for a more in depth look at license compatibility.

GPL v.3

Maybe

The LGPL has an exemption that allows for the linking of LGPL code to non-LGPL code, without violating the license and without requiring source code disclosure of non-LGPL files.

The license describes a library as a collection of software functions intended to be conveniently linked with application programs to form executables.

A program that is designed to work with the LGPL licensed library by being compiled or linked with it, and does not contain a portion of the licensed library, is a work that uses the library, and is not a derivative work and therefore outside the scope of the LGPL. Any modifications to the licensed library itself or any work that contains portions of the licensed library is considered a derivative work and therefore covered by the LGPL.

LGPL v.2.1

Yes:

No specific restriction.

New BSD License

Yes:

Unlike strong copyleft licenses, code under the MPL may be combined with code not licensed under the MPL. When such a larger work has been created, MPL source code and any modifications thereof must remain under the terms of the MPL, however non-MPL code remains non-MPL.

Mozilla Public License (MPL) 1.1

Yes:

No restrictions.

Apache License 2.0

Obligations:

Do I have to release the source code of my modifications?

Yes:

If the modified code will be conveyed externally, there is an obligation to make the source code for all original and modified portions of the licensed code available to all downstream users.

You must prominently notify users what files have been modified and the date of change.

Include all original copyright notices and warranty disclaimers.

You may distribute the GPL licensed software for a fee, but purchasers have the freedom to release it to the public without a fee.

Yes:

No specific restriction

Yes:

If the modified code will be conveyed externally, there is an obligation to make the source code for all original and modified portions of the licensed code available to all downstream users.

You must prominently notify users what files have been modified and the date of change.

Include all original copyright notices and warranty disclaimers. Prominently notify users that the work is released under this license and of any additional permissions.

Yes:

The US Digital Millennium Copyright Act and similar non-US laws prohibit the intentional circumvention of technological measures designed to prevent unauthorized use/access to copyrighted works. (Note: Canada does not currently have any anti circumvention laws)

(continued)

Yes:

If the modified code will be conveyed externally, there is an obligation to make the source code for all original and modified portions of the licensed code available to all downstream users.

You must prominently notify users what files have been modified and the date of change.

Include all original copyright notices and warranty disclaimers

Yes:

No specific restriction. However, with LGPL v.3 licensed software you waive any legal power to forbid circumvention of the technological measures when you distribute the licensed work.

No:

No obligation to disclose the source code of your modifications.

Yes:

No specific restriction.

Yes:

You must make the source code of your modifications available.

Maintain a file documenting modifications, date of the change, and a prominent statement that the modification is derived from the original code, and include the name of the initial developer in the source code.

Yes:

No specific restriction.

No:

No obligation to disclose the source code of your modifications.

Prominently notify users of any modified files.

You may add your own copyright statement and license terms to your modifications so long as you do not remove any of the original license requirements.

Yes:

No specific restriction.

GPL v.2

GPL v.3

LGPL v.2.1

New BSD License

Mozilla Public License (MPL) 1.1

Apache License 2.0

Can I use the licensed software as part of a technological measure?

Obligations:

I plan on using the licensed software internally only

Can I distribute the licensed software unmodified?

There are no restrictions on use if the GPL licensed software is used internally and is not distributed outside the organization. You may even combine GPL licensed software with proprietary licensed software.

Yes:

If the unmodified licensed software will be conveyed outside the organization, there is an obligation to make the source code available to downstream users and publish the original copyright notices and warranty disclaimers.

You may not impose any further restrictions on the recipient's rights.

No restrictions on internal use as long as your license otherwise remains in force.

Yes:

If the unmodified licensed software will be conveyed outside the organization, there is an obligation to make the source code available to downstream users and conspicuously publish on each copy the original copyright notices, warranty disclaimers, and give all recipients a copy of the license.

You may not impose any further restrictions on the recipient's rights. However, you may remove additional permissions and place additional permissions on material added by you.

No restrictions on internal use.

Yes:

If the unmodified licensed software will be conveyed outside the organization, there is an obligation to make the source code available to downstream users and publish the original copyright notices and warranty disclaimers.

You may not impose any further restrictions on the recipient's rights.

No restrictions on internal use.

Yes:

No obligation to disclose source code.

Redistribution of source and binary code must retain the copyright notices, and you must not use the name of the licensor to endorse or promote products derived from the software.

No restrictions on internal use.

Yes:

Include a copy of the license with every copy of the source code you distribute.

Duplicate the notice contained in Exhibit A in each file of the source code.

No restrictions on internal use.

Yes:

No obligation to disclose source code. You may redistribute the original or modified code as open source or proprietary.

You may copy and distribute the software so long as you provide a copy of the license and retain the copyright, patent, trademark and attribution notices from the originating file.

GPL v.2

GPL v.3

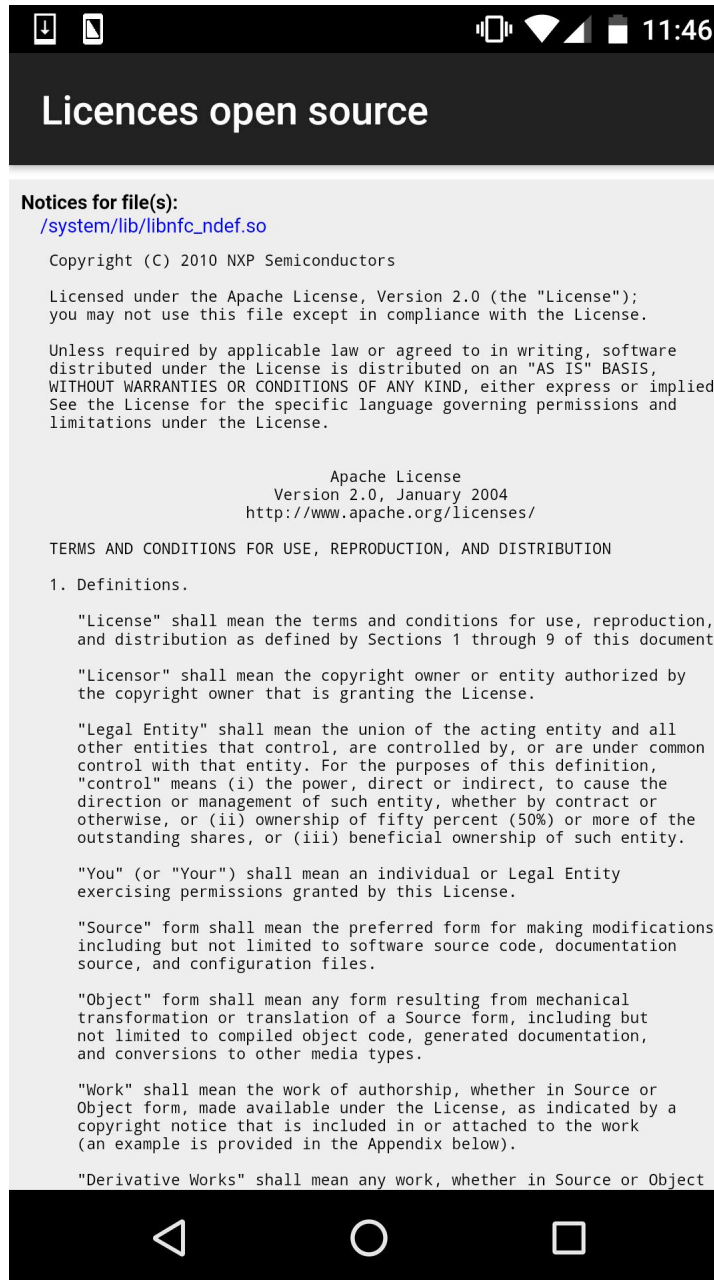
LGPL v.2.1

New BSD License

Mozilla Public License (MPL) 1.1

Apache License 2.0

Exemple : Android



License Open Source : GNU AGPL

Pour ce projet, si cela vous convient, nous utiliserons la license GNU AGPL, qui repose sur la GNU GPL.

GPL:

"You may copy, distribute and modify the software as long as you track changes/dates in source files. Any modifications to or software including (via compiler) GPL-licensed code must also be made available under the GPL along with build & install instructions."

[https://tldrlegal.com/license/gnu-general-public-license-v3-\(gpl-3\)](https://tldrlegal.com/license/gnu-general-public-license-v3-(gpl-3))

License Open Source : GNU AGPL

Pour ce projet, si cela vous convient, nous utiliserons la license GNU AGPL, qui repose sur la GNU GPL.

AGPL:

“You can distribute modified versions if you keep track of the changes and the date you made them. As per usual with GNU licenses, you must license derivatives under AGPL. It provides the same restrictions and freedoms as the GPLv3 but with an additional clause which makes it so that source code must be distributed along with web publication. Since web sites and services are never distributed in the traditional sense, the AGPL is the GPL of the web.”





[https://tldrlegal.com/license/gnu-affero-general-public-license-v3-\(agpl-3.0\)](https://tldrlegal.com/license/gnu-affero-general-public-license-v3-(agpl-3.0))

Comment appliquer la license ?

- Appliquer un header au-dessus de vos fichiers protégés
- Fournir le fichier contenant la license avec votre projet

Comment appliquer la license ?


- Fournir le fichier contenant la license avec votre projet

 francoismichel commit m2dem2		
 M1Q7		commit m2dem2
 .gitignore		Initial commit
 LICENSE		Initial commit

Comment appliquer la license ?

- Appliquer un header au-dessus de vos fichiers protégés

Branch: master ▾ Informatique-1 / Mission2 / demarrage / m2dem2 / student / M1Q7.java

 **francoismichel** update license + refresh missions from inginius

1 contributor


108 lines (99 sloc) | 3.86 KB

```
1  /**
2   * Copyright (c) 2015, 2016 François Michel, Clémentine Munyabarenzi
3   * This program is free software: you can redistribute it and/or modify
4   * it under the terms of the GNU Affero General Public License as published by
5   * the Free Software Foundation, either version 3 of the License, or
6   * (at your option) any later version.
7   * This program is distributed in the hope that it will be useful,
8   * but WITHOUT ANY WARRANTY; without even the implied warranty of
9   * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
10  * GNU Affero General Public License for more details.
11  *
12  * You should have received a copy of the GNU Affero General Public License
13  * along with this program. If not, see <http://www.gnu.org/licenses/>.
14  */
15
16
17
18  package student;
19  import static org.junit.Assert.*;
```

INGInious vs GitHub

Nous avons créé un cours OpenWeek sur INGInious. Il vous servira à tester vos tâches ainsi qu'à tester celles des autres. Cependant, pour rester propre, on utilisera GitHub pour sauvegarder les tâches créées lors de l'OpenWeek. Ce sont les tâches présentes sur le GitHub qui seront intégrées sur le cours INGInious de Java BAC1.

Merci d'avoir suivi

Si vous vous sentez un peu  après cette présentation, n'oubliez pas que les slides sont disponibles sur le GitHub du projet et que je suis toujours présent si vous avez une question ou si vous avez besoin d'un coup de main.