Aix*Marseille université
Initiative d'excellence
Année 2018-19

Domaine Sciences et Technologies
Licence informatique

Programmation 2 : TP 1 (informatique)
Code UE : SIN3U02

Gestion des notes des étudiants

1 Gestion des notes des étudiants

1.1 Introduction

Le but de ce TP est de créer des classes permettant de représenter des étudiants (classe `Student`), des notes (classe `Grade`), des résultats à une unité d'enseignement (classe `TeachingUnitResult`) et des promotions d'étudiants (classe `Cohort`).

L'objectif pour vous est de vous initier :

— à l'utilisation d'un IDE : IntelliJ IDEA de JetBrains est mis en avant mais n'importe quel IDE équivalent pourra être utilisé,
— à la gestion de version grâce à git,
— à l'utilisation des tests (ici déjà programmé) pour valider votre programme.

Vous allez sans doute découvrir un certain nombre de nouveaux outils durant ce TP. Le temps d'apprentissage de ces outils peut être au début frustrant mais ils vont vous faire gagner du temps par la suite.

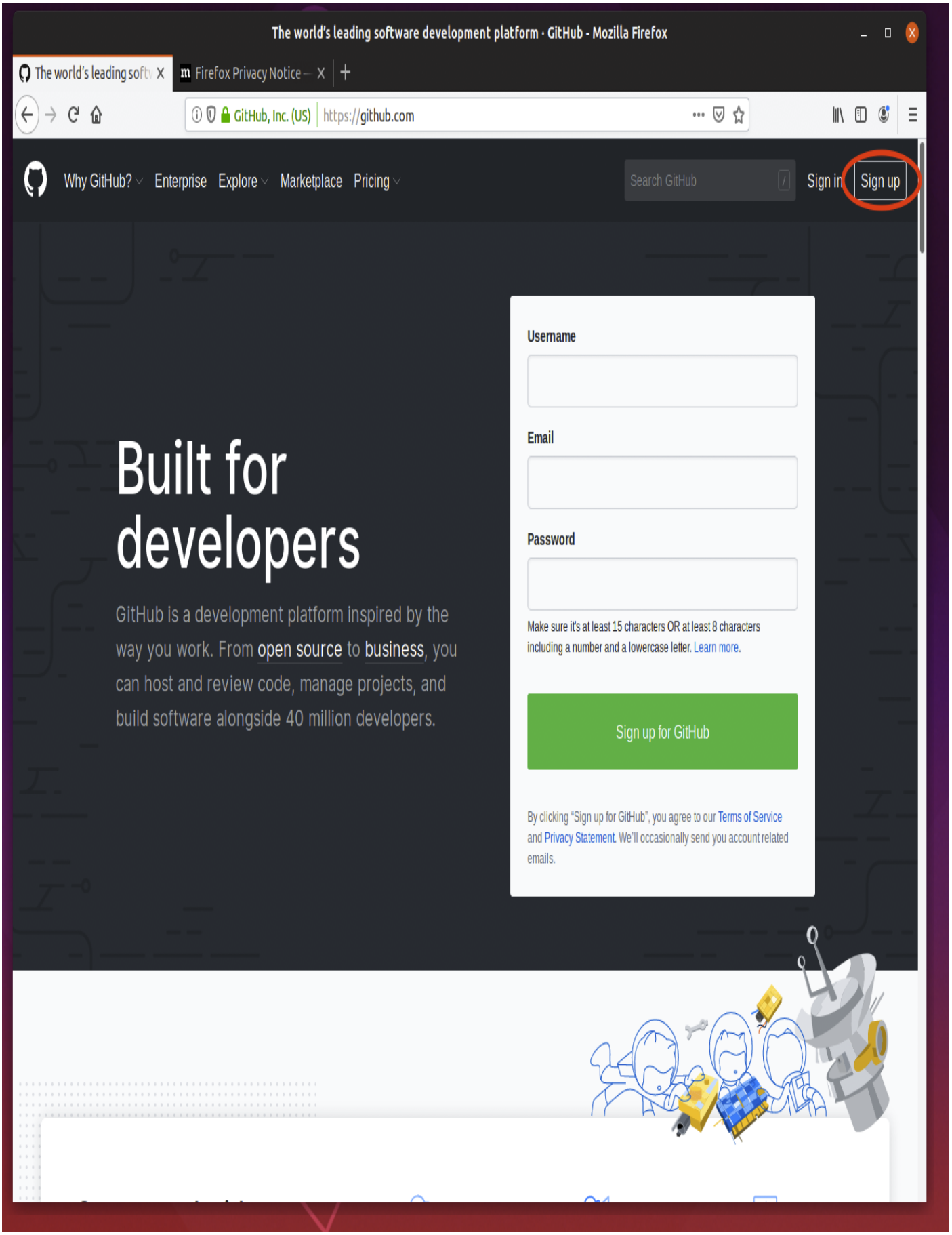
1.2 Mise en place du TP

1.2.1 Créer le dépôt git pour le TP

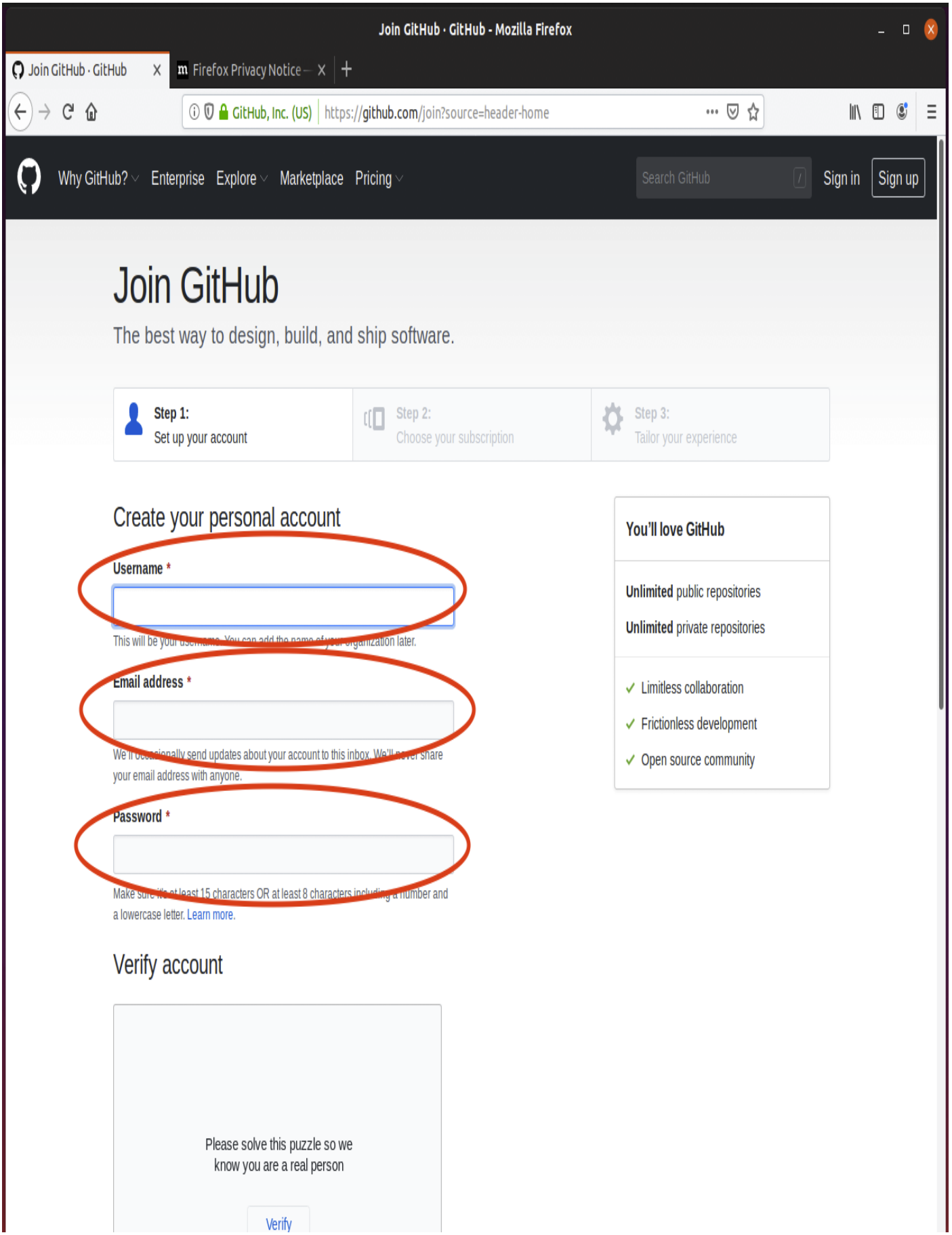
Ce TP est l'occasion d'apprendre à utiliser des outils de développement professionnel. En plus de l'IDE IntelliJ, vous aurez à utiliser un outil de gestion de versions, en l'occurrence `git`. Un tel outil permet d'enregistrer à distance votre projet et de permettre à plusieurs personnes de travailler simultanément, sans devoir échanger les fichiers par courriel, et sans se marcher sur les pieds (par exemple modification sans le savoir du même fichier).

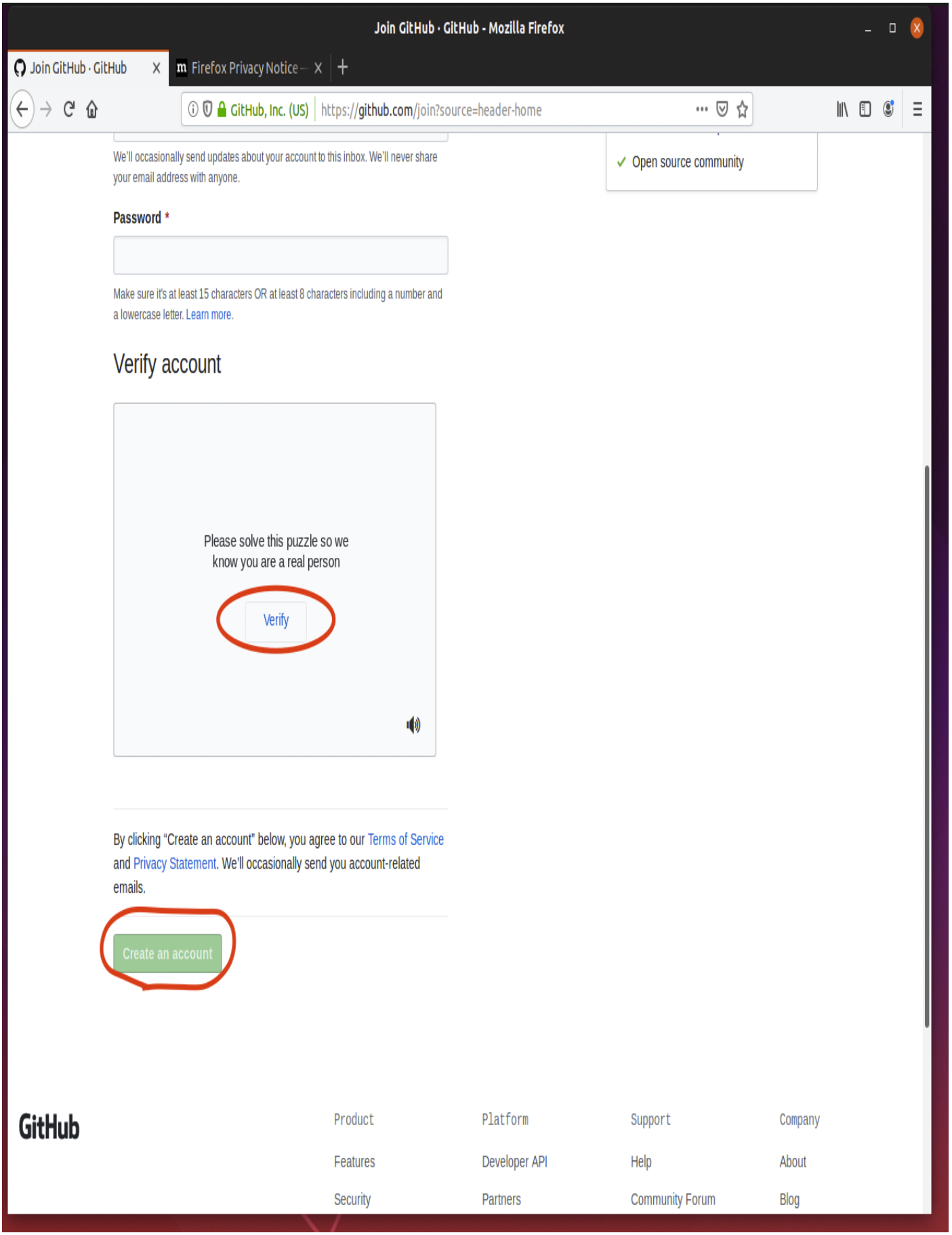
La première étape pour utiliser la gestion de version est de vous inscrire à github et de créer un dépôt à l'aide de github classroom.

1. Si vous ne possédez pas de compte Github, aller sur github et créez-vous un compte gratuit. Pour celà, allez sur `https://github.com` et cliquez sur le bouton `signup`.


The world's leading software development platform · GitHub - Mozilla Firefox
The world's leading soft
Firefox Privacy Notice
GitHub, Inc. (US)
https://github.com
Why GitHub?
Enterprise
Explore
Marketplace
Pricing
Search GitHub
Sign in
Sign up
Built for developers
GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 40 million developers.
Username
Email
Password
Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.
Sign up for GitHub
By clicking "Sign up for GitHub", you agree to our Terms of Service and Privacy Statement. We'll occasionally send you account related emails.

Remplissez le formulaire avec votre login, adresse email et mot de passe. Compléter le test pour vérifier que vous n'êtes pas une machine et finissez en appuyant sur le bouton create account.


Join GitHub · GitHub - Mozilla Firefox
Join GitHub · GitHub
Firefox Privacy Notice
GitHub, Inc. (US)
https://github.com/join?source=header-home
Why GitHub?
Enterprise
Explore
Marketplace
Pricing
Search GitHub
Sign in
Sign up
Join GitHub
The best way to design, build, and ship software.
Step 1:
Set up your account
Step 2:
Choose your subscription
Step 3:
Tailor your experience
Create your personal account
Username *
This will be your username. You can add the name of your organization later.
Email address *
We'll occasionally send updates about your account to this inbox. We'll never share your email address with anyone.
Password *
Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.
You'll love GitHub
Unlimited public repositories
Unlimited private repositories
Limitless collaboration
Frictionless development
Open source community
Verify account
Please solve this puzzle so we know you are a real person
Verify


Join GitHub · GitHub - Mozilla Firefox
Join GitHub · GitHub
Firefox Privacy Notice —
GitHub, Inc. (US)
https://github.com/join?source=header-home
We'll occasionally send updates about your account to this inbox. We'll never share your email address with anyone.
✓ Open source community
Password *
Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.
Verify account
Please solve this puzzle so we know you are a real person
Verify
By clicking "Create an account" below, you agree to our Terms of Service and Privacy Statement. We'll occasionally send you account-related emails.
Create an account
GitHub
Product
Features
Security
Platform
Developer API
Partners
Support
Help
Community Forum
Company
About
Blog

Vous allez recevoir un mail sur l'adresse que vous avez indiquée. Cliquez sur le bouton `verify email address` pour valider votre adresse.

[GitHub] Please verify your email address.

GitHub <noreply@github.com>
Lun 2019-09-09 23:02
Vous

Almost done, [redacted] To complete your GitHub sign up, we just need to verify your email address:
[redacted]

Verify email address

Once verified, you can start using all of GitHub's features to explore, build, and share projects.

Button not working? Paste the following link into your browser: https://github.com/users/serbrek/emails/82800937/confirm_verification/6dccbe24b223047f477750c0ba86d73a339c7ba2

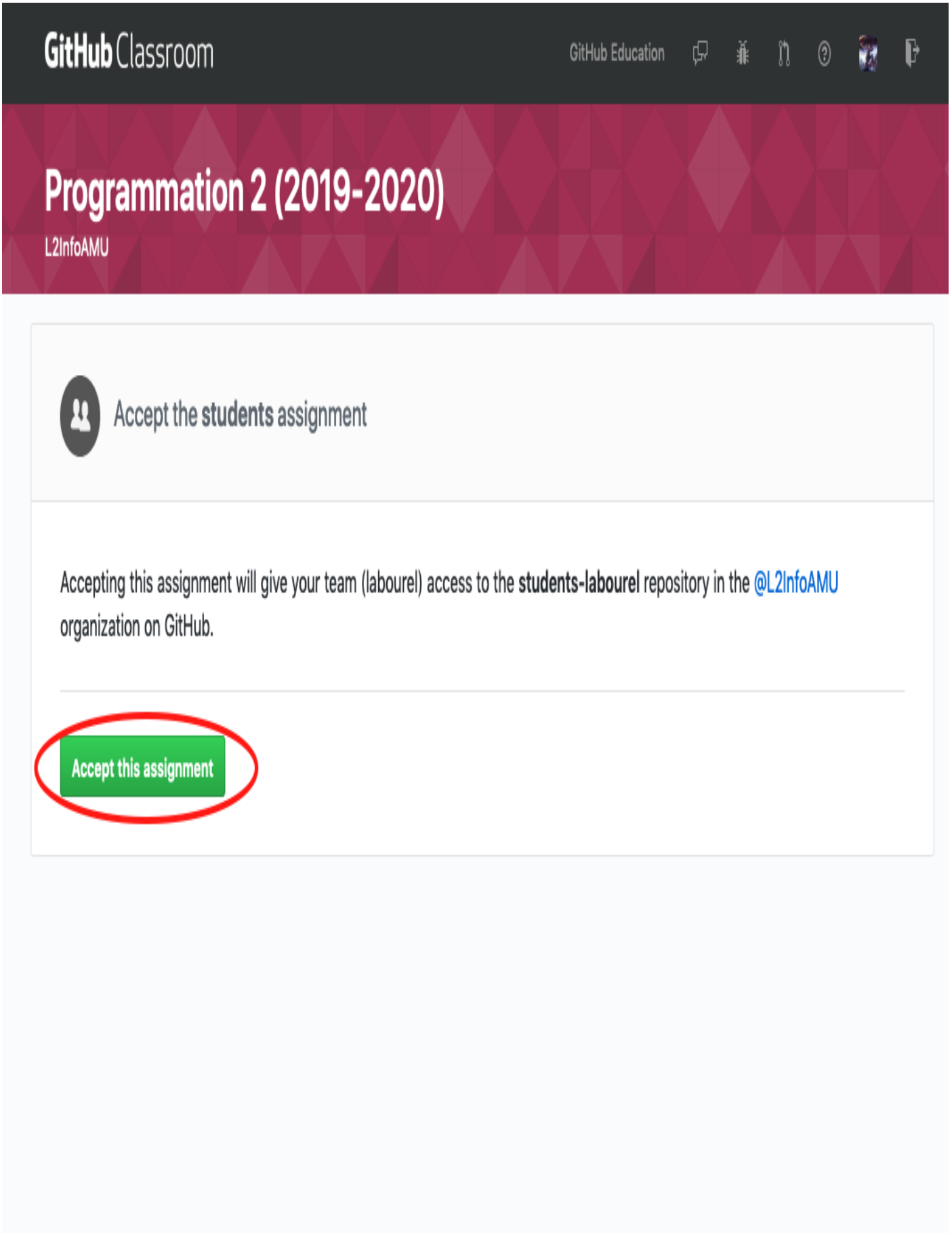
You're receiving this email because you recently created a new GitHub account or added a new email address. If this wasn't you, please ignore this email.

Email preferences · [/articles/github-terms-of-service/*%7CUPR_UTM%7C*]Terms · [/articles/github-privacy-policy/*%7CUPR_UTM%7C]Privacy · Sign into GitHub

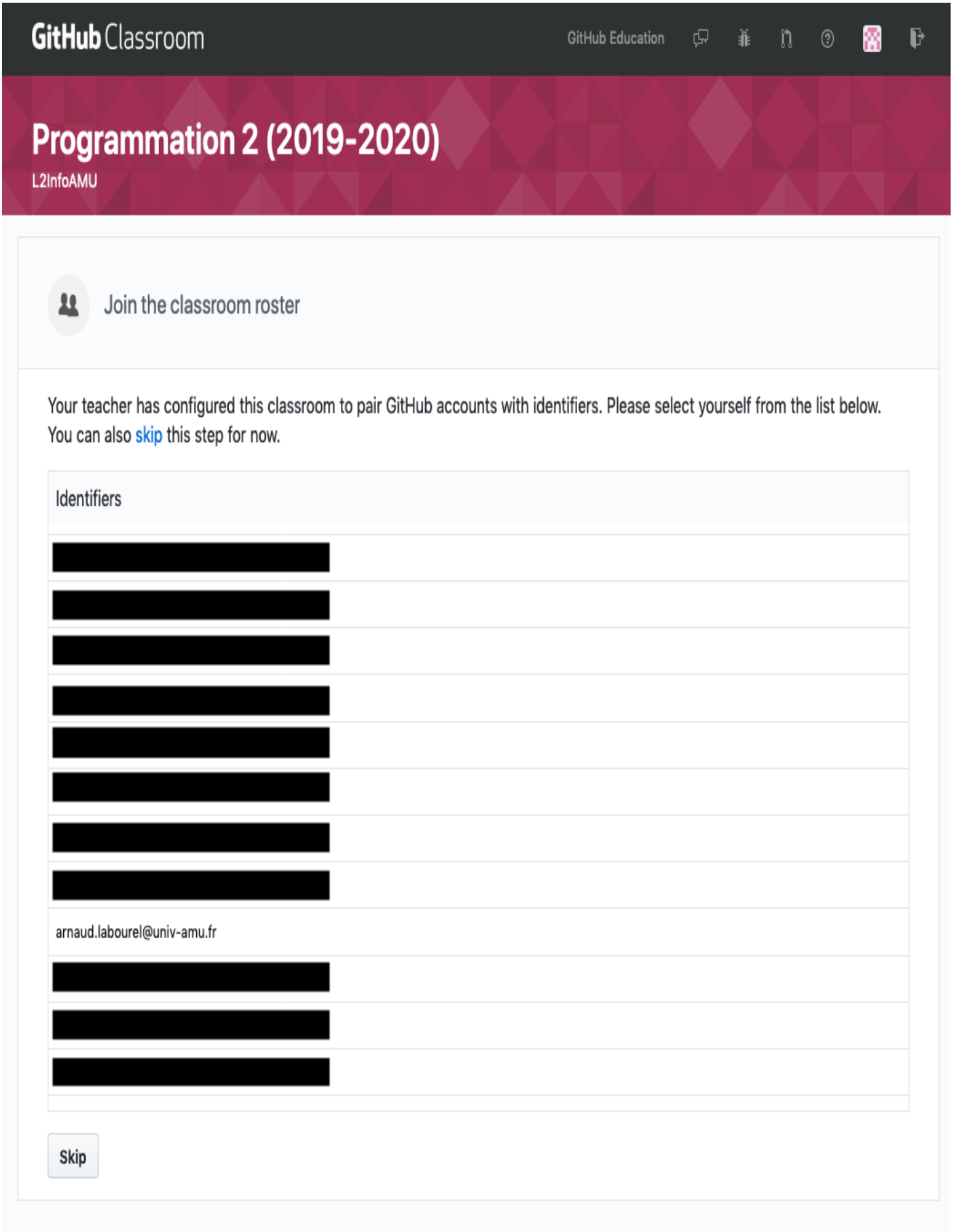
GitHub

Sent with <3 by GitHub.

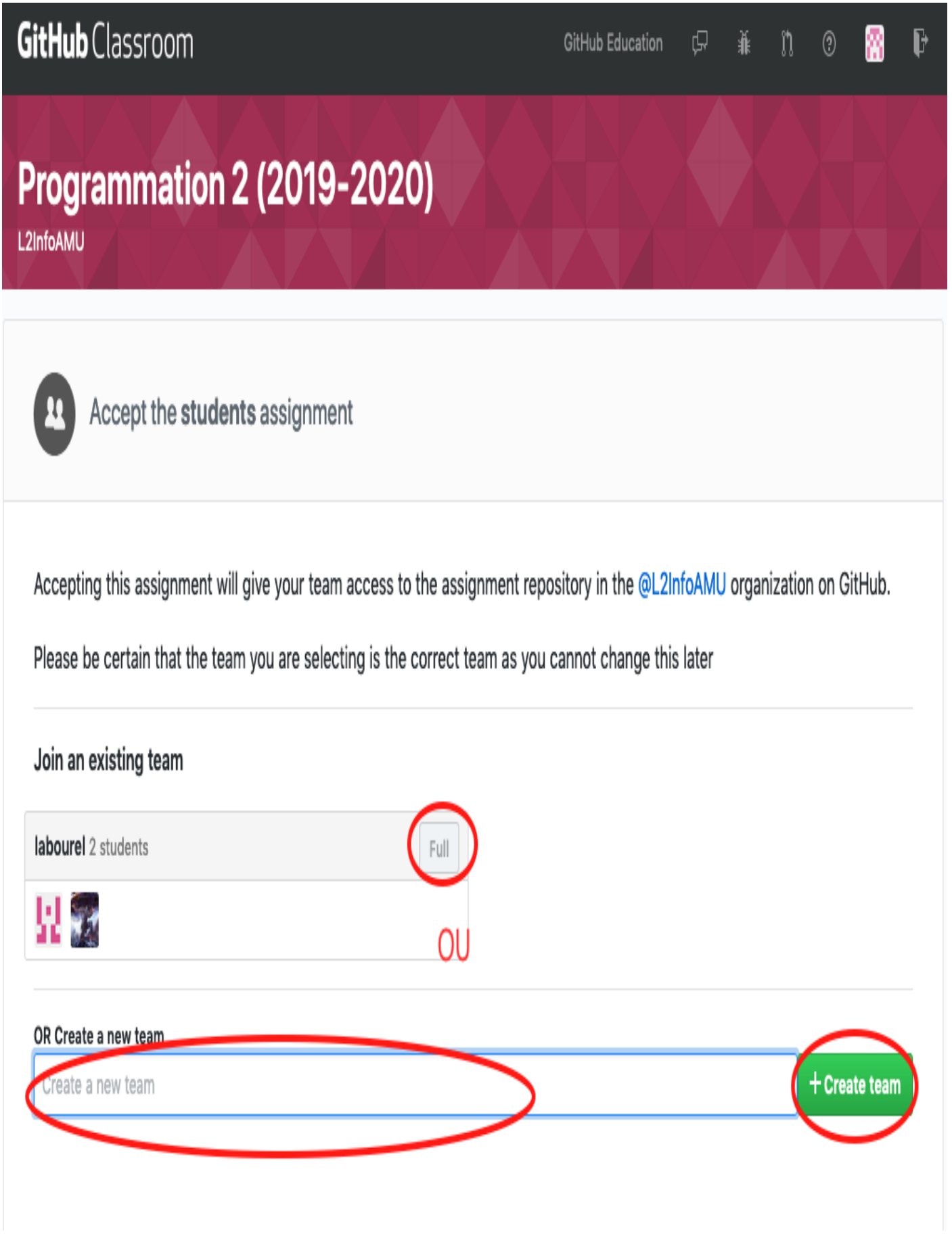
2. Vous allez maintenant créer un dépôt grâce à github classroom. Pour cela, il vous suffit d'aller à l'adresse suivante : `https://classroom.github.com/g/vjEBS5Ee` en étant connecté sur github à votre compte et d'accepter le devoir en cliquant sur le bouton `accept this assignment`.


GitHub Classroom
GitHub Education
Programmation 2 (2019-2020)
L2InfoAMU
Accept the students assignment
Accepting this assignment will give your team (labourel) access to the students-labourel repository in the @L2InfoAMU organization on GitHub.
Accept this assignment

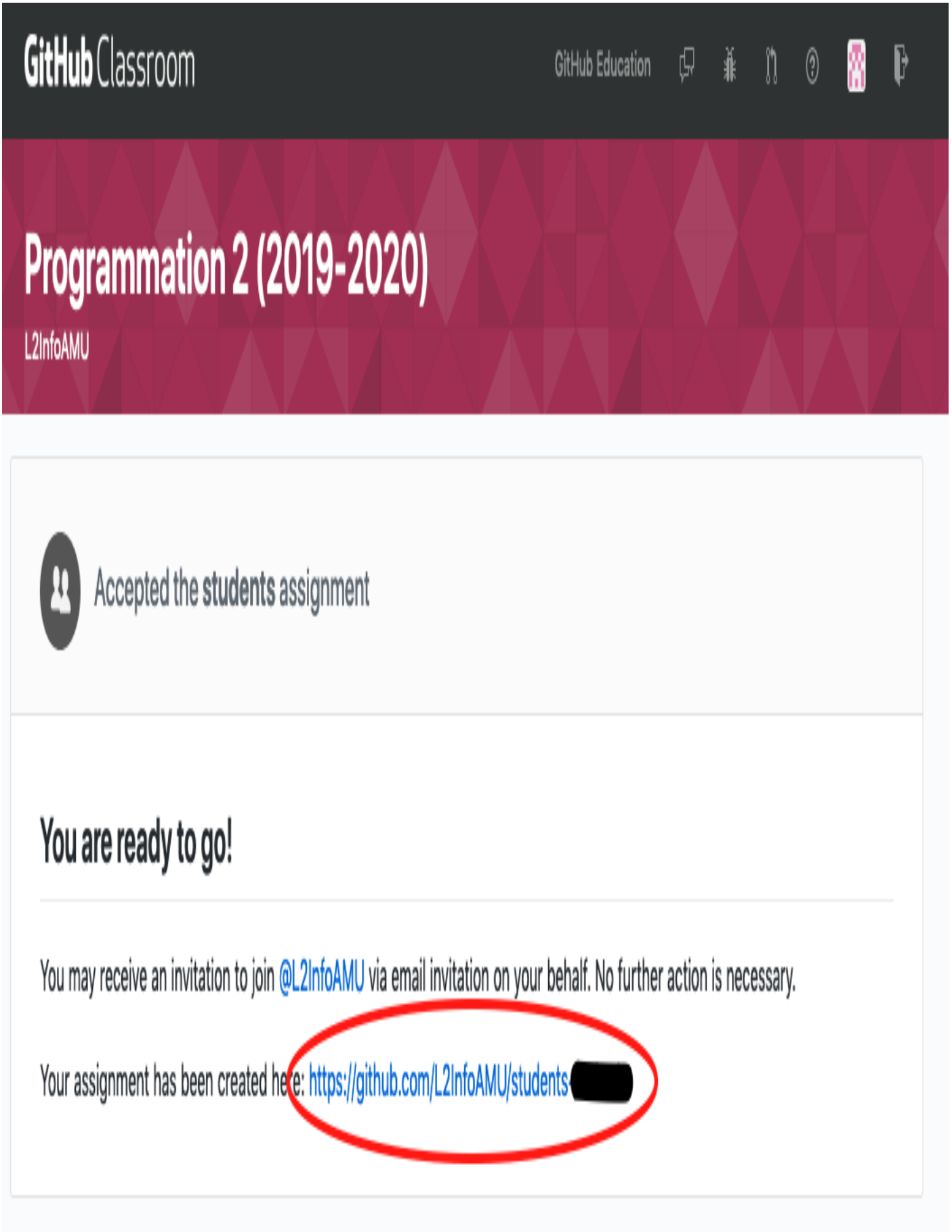
On vous demande maintenant de cliquer sur votre adresse étudiante (adresse `@etu.univ-amu.fr`) afin que vous soyez clairement identifié sur `github`. C'est très important car certains des TP que vous ferez par la suite sur `github` seront notés.


GitHub Classroom
GitHub Education
Programmation 2 (2019-2020)
L2InfoAMU
Join the classroom roster
Your teacher has configured this classroom to pair GitHub accounts with identifiers. Please select yourself from the list below.
You can also skip this step for now.
Identifiers
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]
arnaud.labourel@univ-amu.fr
[redacted]
[redacted]
[redacted]
Skip

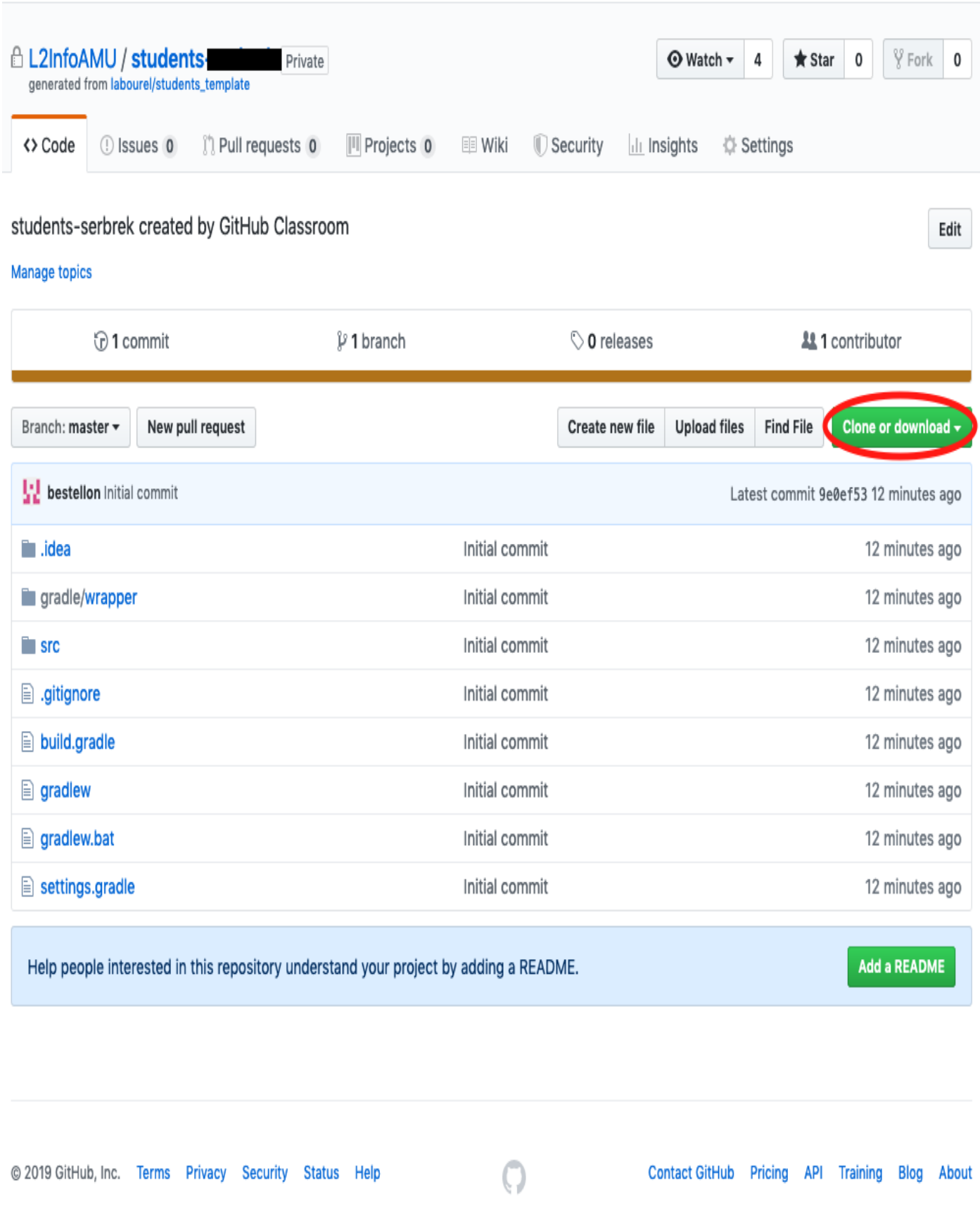
Ensuite, vous devez choisir soit de créer une équipe ou bien de rejoindre une équipe existante. Ce TP peut se faire en binôme ou bien seul. Faites très attention à cette étape car vous ne pourrez plus changer d'équipe par la suite. Pour créer une équipe, il vous suffit de remplir le nom que vous souhaitez pour votre équipe (évitez les noms trop long) en bas et de cliquer sur le bouton `create team`. Sinon vous pouvez rejoindre une équipe déjà créée en cliquant sur `join` (chaque équipe pourra avoir au plus 2 participants).


GitHub Classroom
GitHub Education
Programmation 2 (2019-2020)
L2InfoAMU
Accept the students assignment
Accepting this assignment will give your team access to the assignment repository in the @L2InfoAMU organization on GitHub.
Please be certain that the team you are selecting is the correct team as you cannot change this later
Join an existing team
labourel 2 students
Full
OU
OR Create a new team
Create a new team
+ Create team

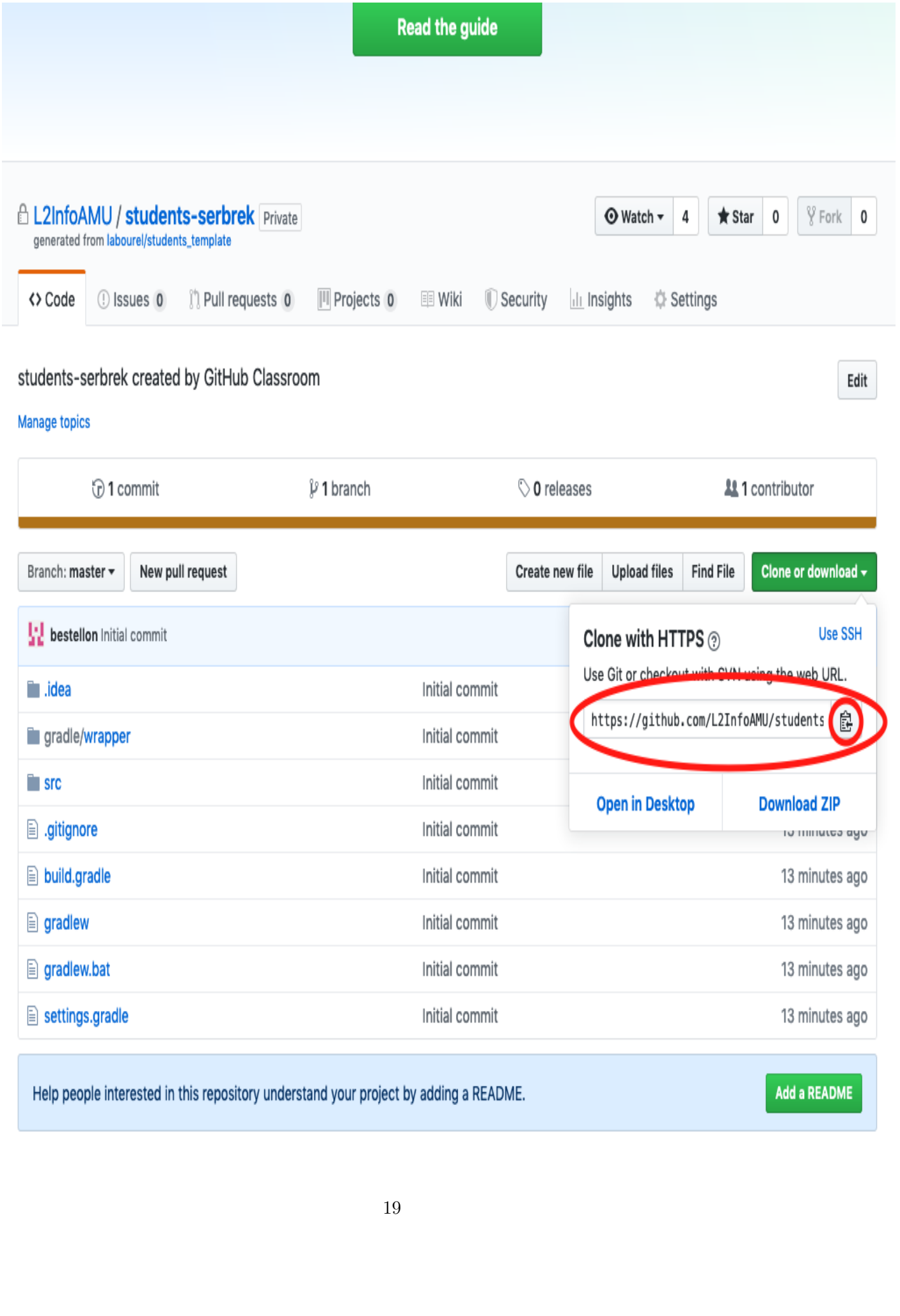
Après cette étape, github va créer un dépôt à partir des sources. Vous devriez avoir accès à l'écran suivant et il vous suffit de cliquer sur le lien.


GitHub Classroom
GitHub Education
Programmation 2 (2019-2020)
L2InfoAMU
Accepted the students assignment
You are ready to go!
You may receive an invitation to join @L2InfoAMU via email invitation on your behalf. No further action is necessary.
Your assignment has been created here: https://github.com/L2InfoAMU/students-

3. Maintenant vous devriez être sur la page de votre dépôt. Cliquez sur le bouton `clone or download`.


L2InfoAMU / students- Private
generated from labourel/students_template
Watch 4
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Security
Insights
Settings
students-serbrek created by GitHub Classroom
Edit
Manage topics
1 commit
1 branch
0 releases
1 contributor
Branch: master
New pull request
Create new file
Upload files
Find File
Clone or download
bestellon Initial commit
Latest commit 9e0ef53 12 minutes ago
.idea Initial commit 12 minutes ago
gradle/wrapper Initial commit 12 minutes ago
src Initial commit 12 minutes ago
.gitignore Initial commit 12 minutes ago
build.gradle Initial commit 12 minutes ago
gradlew Initial commit 12 minutes ago
gradlew.bat Initial commit 12 minutes ago
settings.gradle Initial commit 12 minutes ago
Help people interested in this repository understand your project by adding a README.
Add a README
© 2019 GitHub, Inc. Terms Privacy Security Status Help
Contact GitHub Pricing API Training Blog About

Cela va vous donner l'accès au lien vous permettant de récupérer votre dépôt. Notez bien ce lien. Vous pouvez le copier (le sauver dans le presse-papier) en cliquant sur le bouton à coté.


Read the guide
L2InfoAMU / students-serbrek Private
generated from labourel/students_template
Watch 4
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Security
Insights
Settings
students-serbrek created by GitHub Classroom
Edit
Manage topics
1 commit
1 branch
0 releases
1 contributor
Branch: master
New pull request
Create new file
Upload files
Find File
Clone or download
bestellon Initial commit
Clone with HTTPS
Use SSH
Use Git or checkout with SVN using the web URL.
https://github.com/L2InfoAMU/students
Open in Desktop
Download ZIP
.idea Initial commit
gradle/wrapper Initial commit
src Initial commit
.gitignore Initial commit
build.gradle Initial commit 13 minutes ago
gradlew Initial commit 13 minutes ago
gradlew.bat Initial commit 13 minutes ago
settings.gradle Initial commit 13 minutes ago
Help people interested in this repository understand your project by adding a README.
Add a README

1.3 IDE IntelliJ

1.3.1 Environnement de développement (IDE)

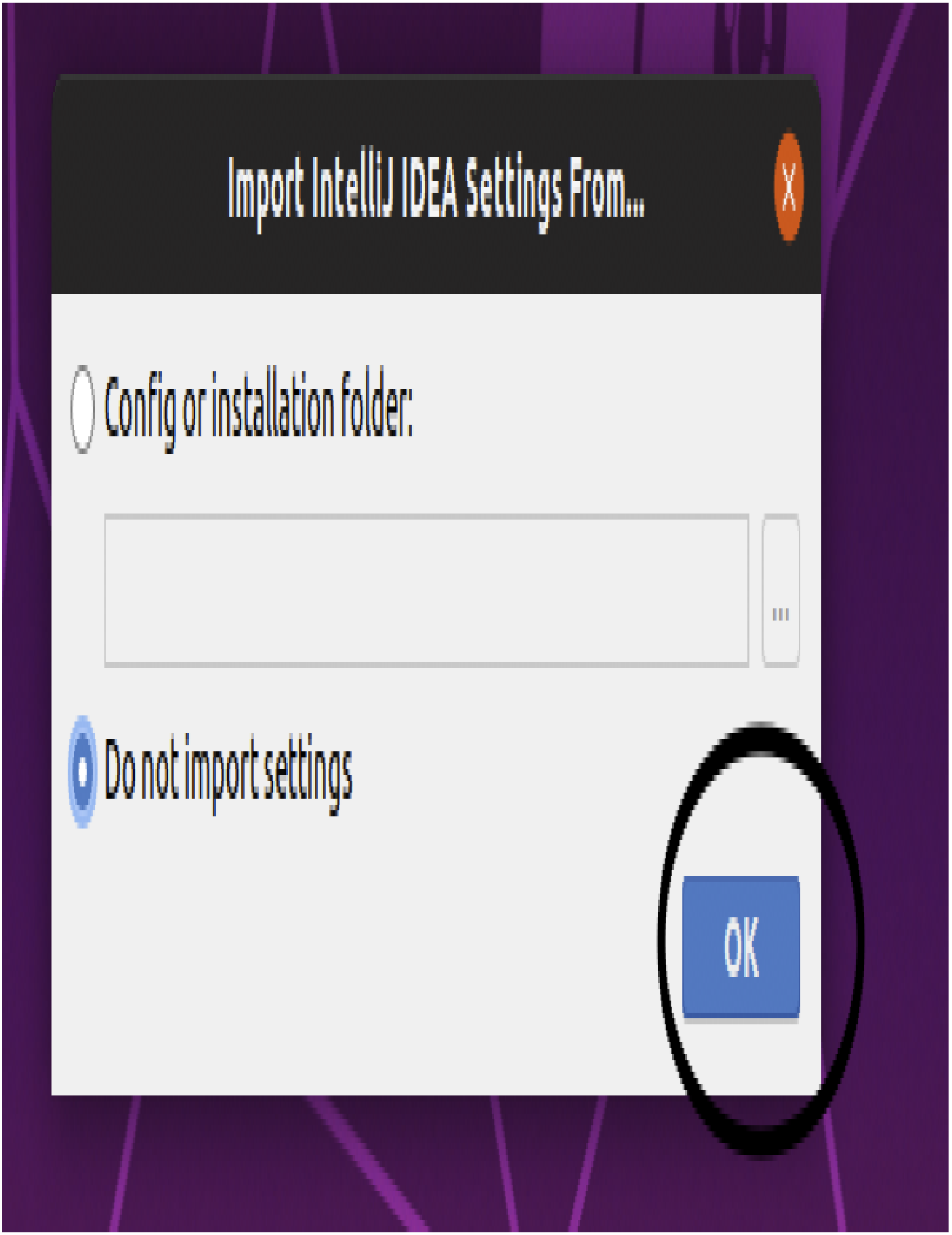
Afin de programmer dans ce cours, nous allons utiliser un environnement de développement (Integrated Development Environment : IDE). Il existe de nombreux IDE pour Java. Dans ce cours, nous vous conseillons d'utiliser IntelliJ IDEA de **JetBrains** mais vous pouvez aussi utiliser Eclipse netbeans ou un autre IDE si vous êtes familier avec ceux-ci.

1.3.2 Lancement de l'IDE

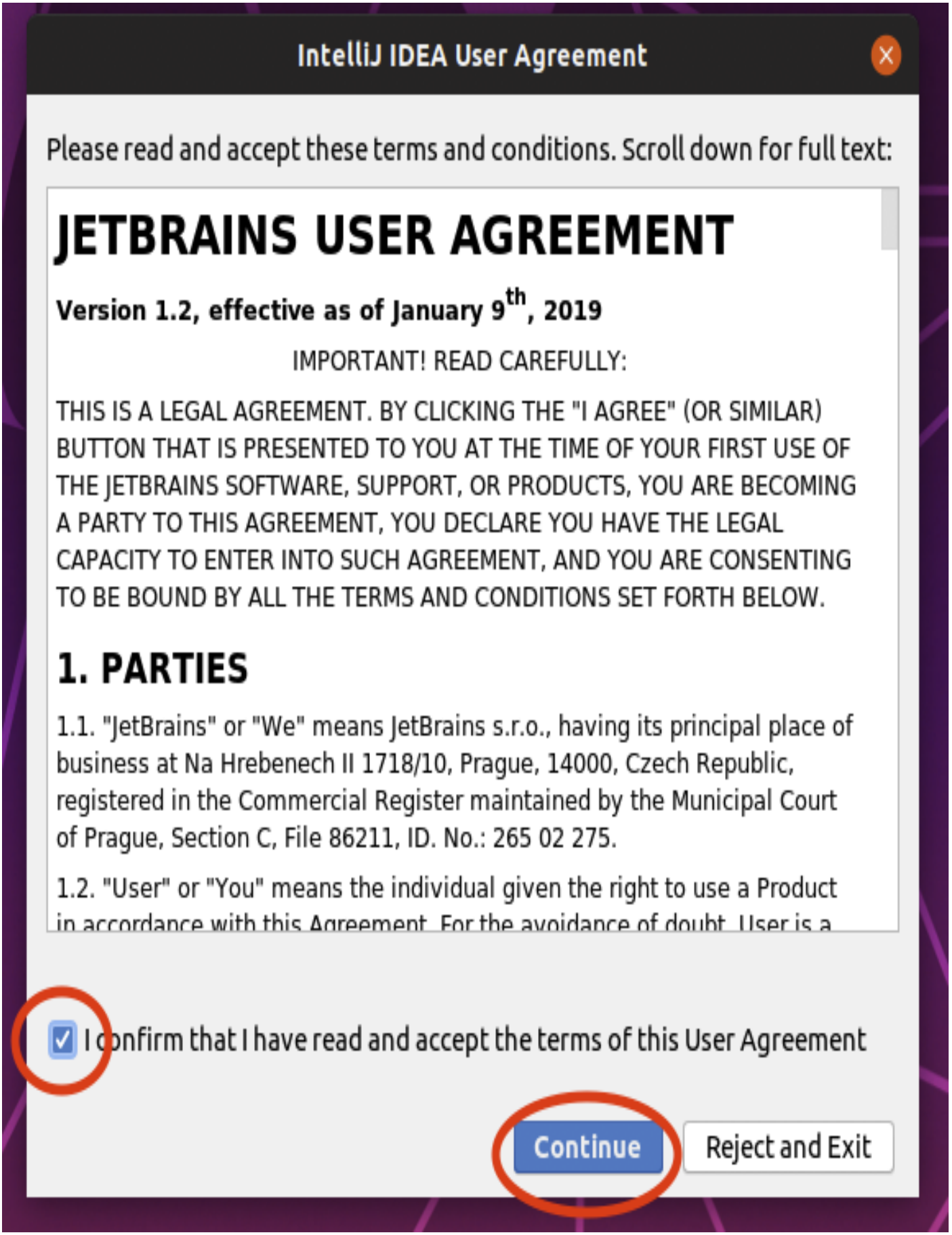
Pour lancer le logiciel sur les machines de l'université, il vous faut aller dans le Menu (en haut à gauche de l'écran) et cliquer sur 'IntelliJ IDEA 2019.2' dans la partie programmation du menu.

Après le chargement, vous pouvez tomber sur une première fenêtre vous proposant d'importer vos paramètres. Il faudra dans l'ordre :

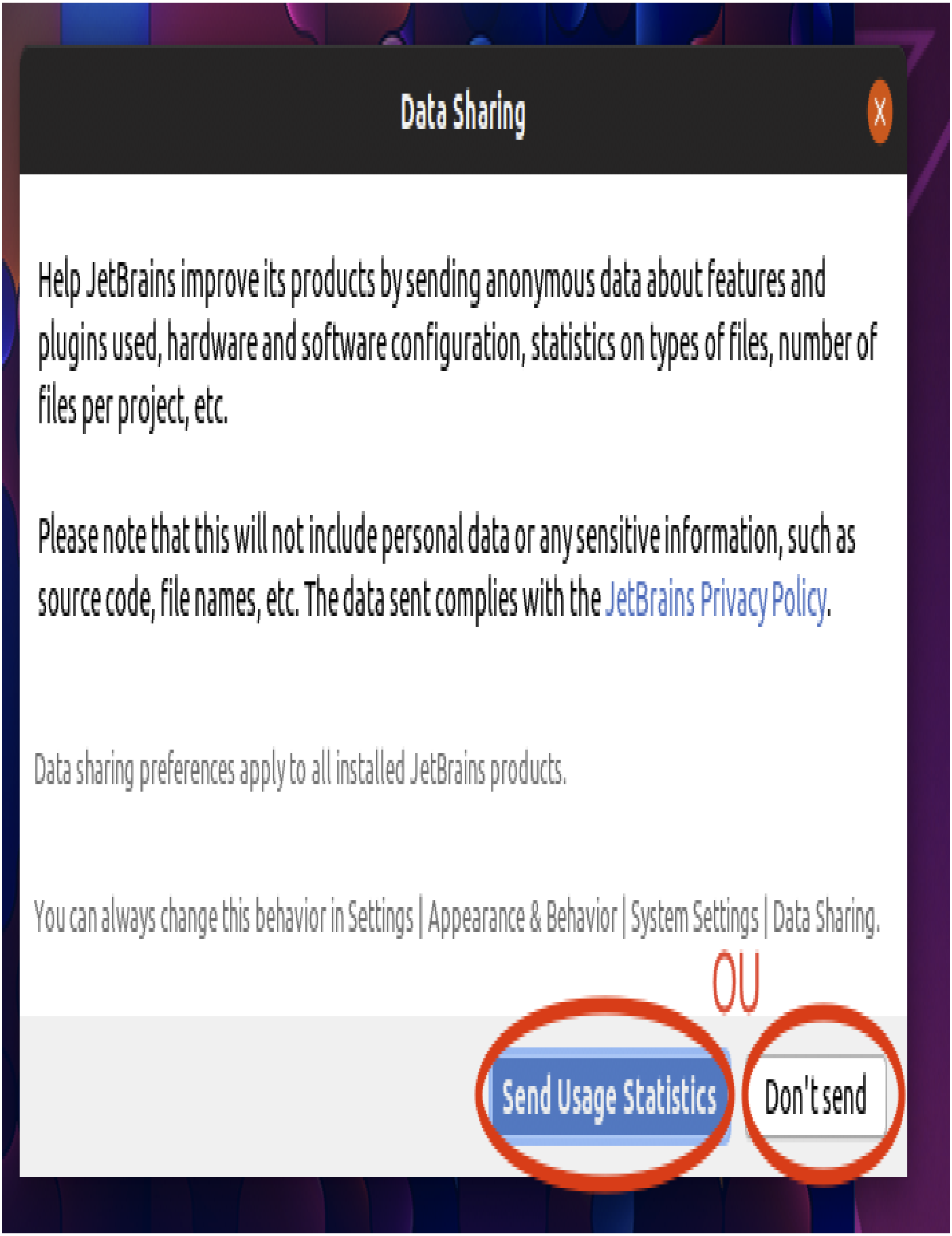
1. Laissez sur 'Do not import settings' et cliquez sur 'OK'.


Import IntelliJ IDEA Settings From...
Config or installation folder:
...
Do not import settings
OK

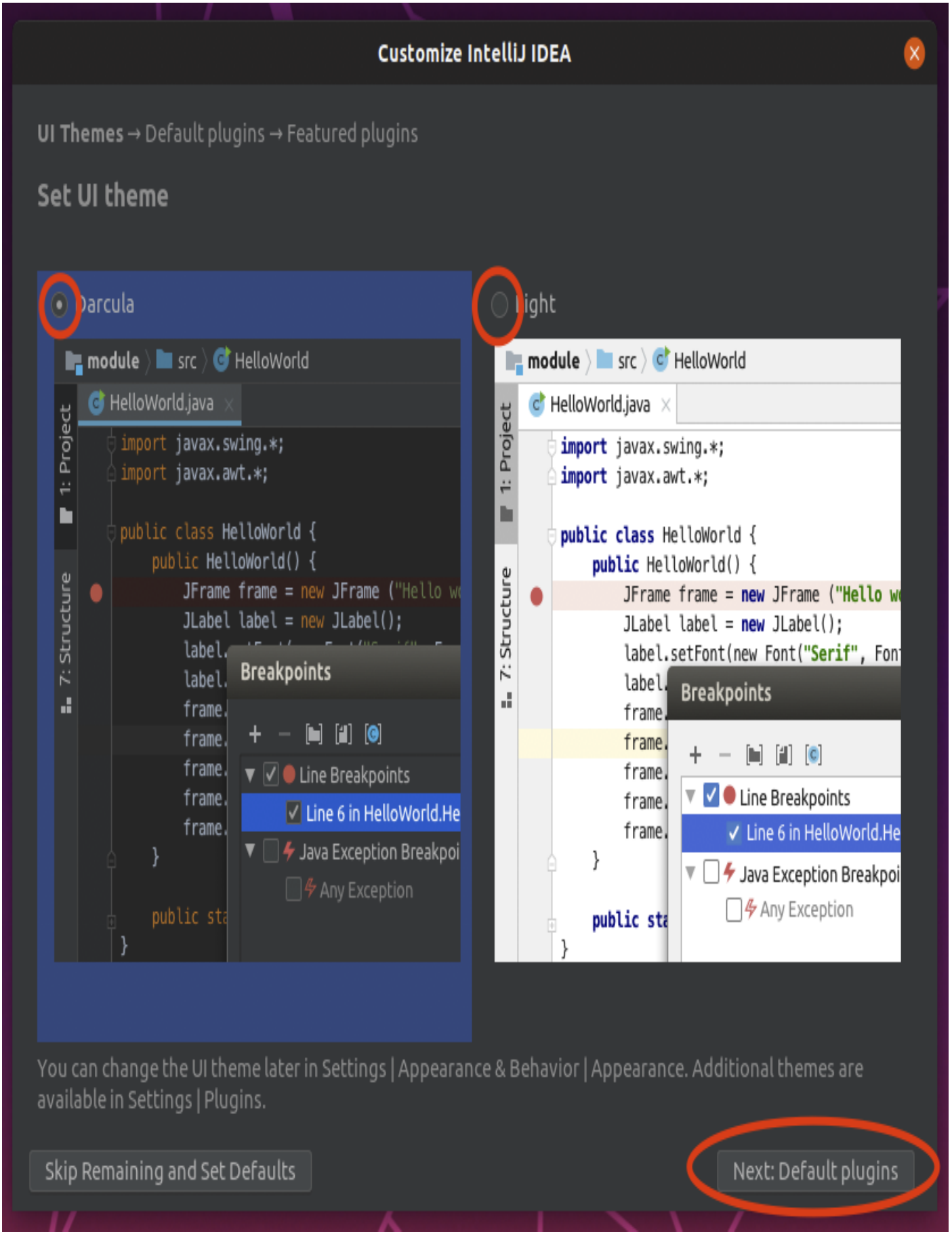
2. Cochez la case comme quoi vous acceptez les termes d'utilisation d'IntelliJ IDEA puis sur continue.


IntelliJ IDEA User Agreement
Please read and accept these terms and conditions. Scroll down for full text:
JETBRAINS USER AGREEMENT
Version 1.2, effective as of January 9th, 2019
IMPORTANT! READ CAREFULLY:
THIS IS A LEGAL AGREEMENT. BY CLICKING THE "I AGREE" (OR SIMILAR) BUTTON THAT IS PRESENTED TO YOU AT THE TIME OF YOUR FIRST USE OF THE JETBRAINS SOFTWARE, SUPPORT, OR PRODUCTS, YOU ARE BECOMING A PARTY TO THIS AGREEMENT, YOU DECLARE YOU HAVE THE LEGAL CAPACITY TO ENTER INTO SUCH AGREEMENT, AND YOU ARE CONSENTING TO BE BOUND BY ALL THE TERMS AND CONDITIONS SET FORTH BELOW.
1. PARTIES
1.1. "JetBrains" or "We" means JetBrains s.r.o., having its principal place of business at Na Hrebenech II 1718/10, Prague, 14000, Czech Republic, registered in the Commercial Register maintained by the Municipal Court of Prague, Section C, File 86211, ID. No.: 265 02 275.
1.2. "User" or "You" means the individual given the right to use a Product
I confirm that I have read and accept the terms of this User Agreement
Continue
Reject and Exit

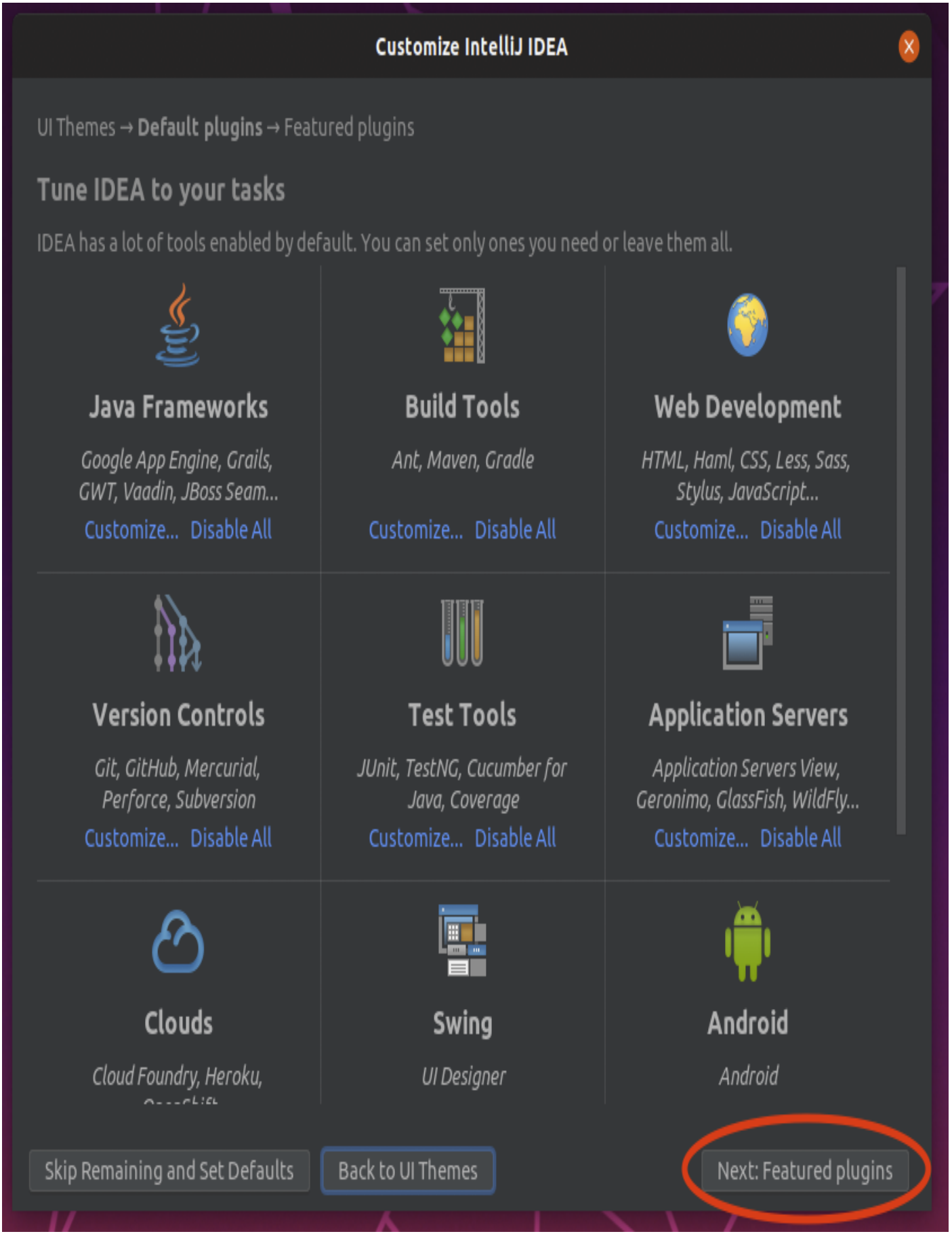
3. Choisissez si vous acceptez ou non d'envoyer les données d'utilisation du logiciel à Jetbrains. À partir de là vous allez personnaliser votre installation pour qu'elle corresponde à vos besoins.

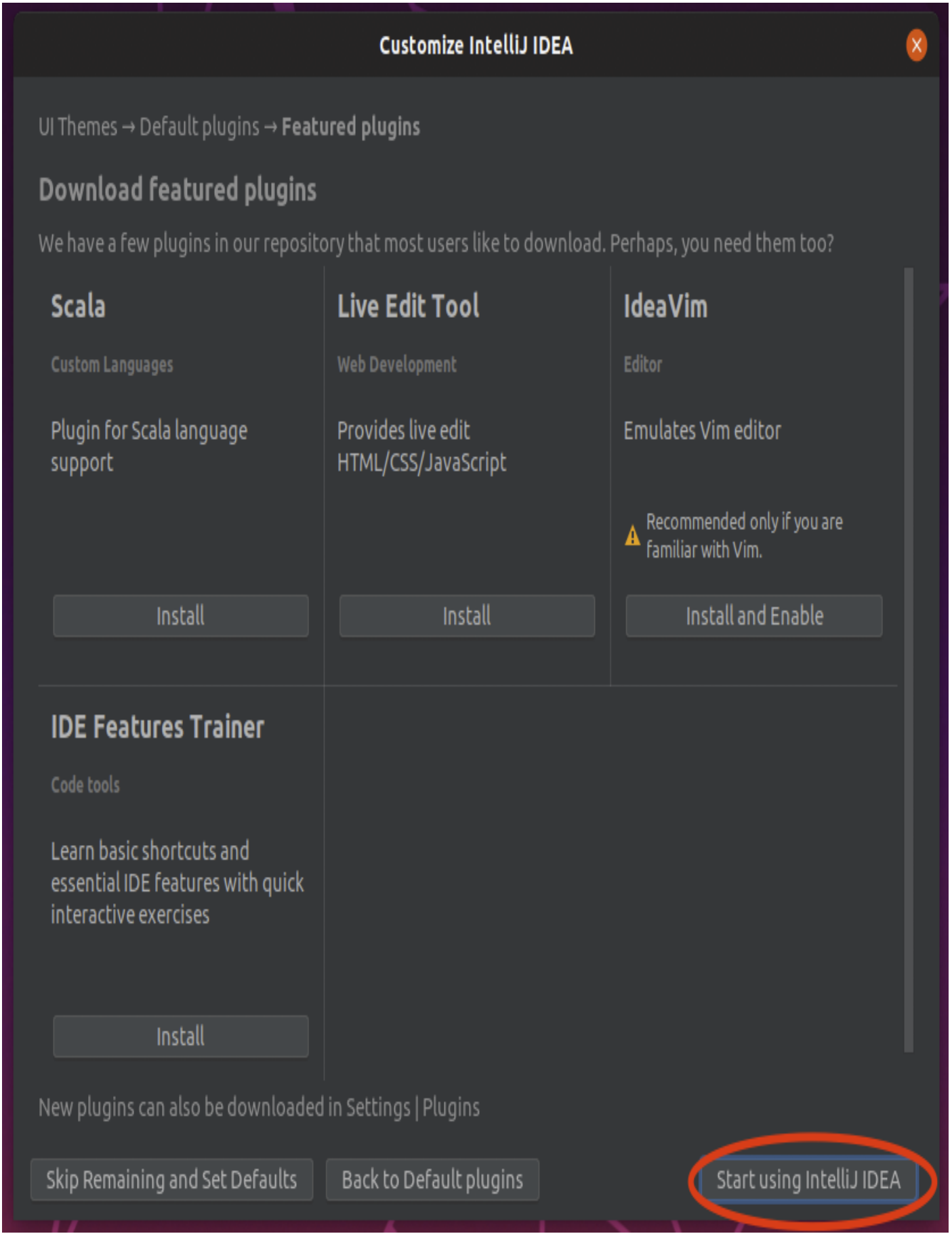

Data Sharing
Help JetBrains improve its products by sending anonymous data about features and plugins used, hardware and software configuration, statistics on types of files, number of files per project, etc.
Please note that this will not include personal data or any sensitive information, such as source code, file names, etc. The data sent complies with the JetBrains Privacy Policy.
Data sharing preferences apply to all installed JetBrains products.
You can always change this behavior in Settings | Appearance & Behavior | System Settings | Data Sharing.
OU
Send Usage Statistics
Don't send

4. Sur la première fenêtre de personnalisation, vous devez choisir l'apparence de votre IDE. Ici rien de fondamental, vous pouvez choisir l'option que vous voulez entre `darcula` (texte blanc sur fond noir) ou `light` (texte noir sur fond clair). Cliquez sur next quand vous avez choisi.

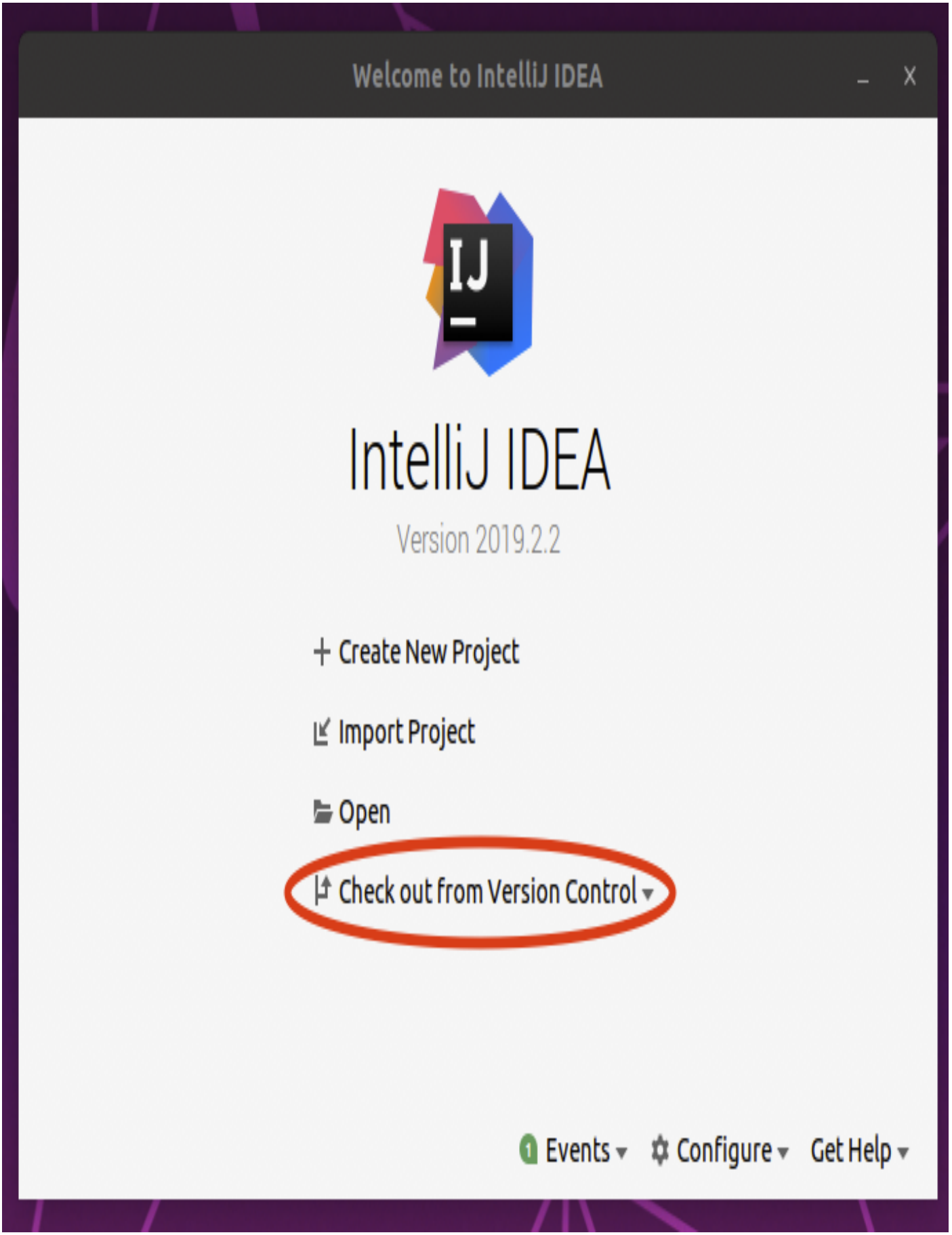

Customize IntelliJ IDEA
UI Themes → Default plugins → Featured plugins
Set UI theme
Darcula
Light
module
src
HelloWorld
HelloWorld.java
1: Project
7: Structure
import javax.swing.*;
import javax.awt.*;
public class HelloWorld {
public HelloWorld() {
JFrame frame = new JFrame ("Hello w
JLabel label = new JLabel();
label.setFont(new Font("Serif", Font
}
public sta
Breakpoints
Line Breakpoints
Line 6 in HelloWorld.He
Java Exception Breakpoi
Any Exception
You can change the UI theme later in Settings | Appearance & Behavior | Appearance. Additional themes are available in Settings | Plugins.
Skip Remaining and Set Defaults
Next: Default plugins

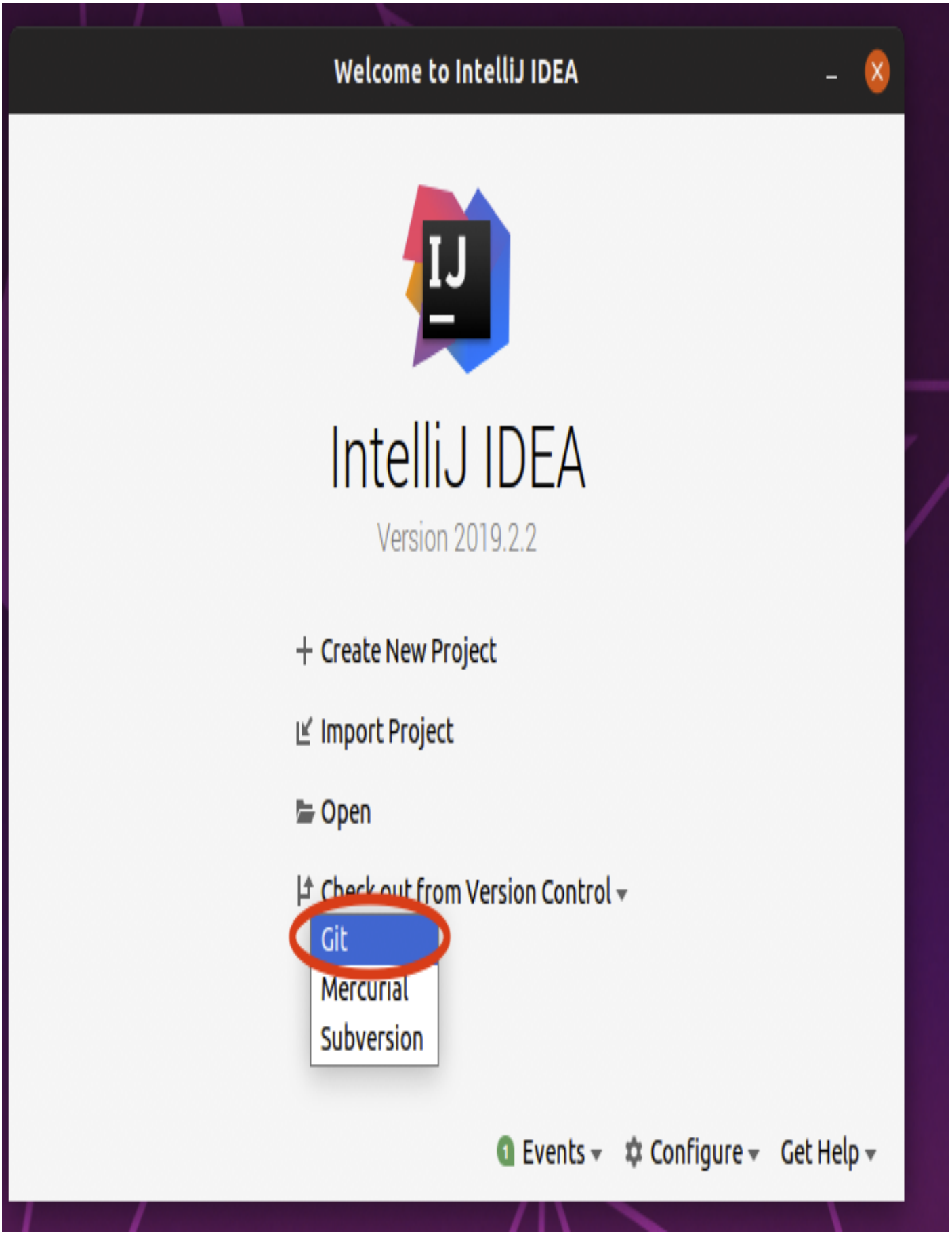
5. Les deux écrans suivants vous permettent de choisir les plugins que vous allez activer. De manière générale, il vaut mieux en activer le moins possible de plugins pour éviter les mauvaises surprises. Vous pouvez donc cliquez sur `next` puis `start`


Customize IntelliJ IDEA
UI Themes → Default plugins → Featured plugins
Tune IDEA to your tasks
IDEA has a lot of tools enabled by default. You can set only ones you need or leave them all.
Java Frameworks
Google App Engine, Grails, GWT, Vaadin, JBoss Seam...
Customize... Disable All
Build Tools
Ant, Maven, Gradle
Customize... Disable All
Web Development
HTML, Haml, CSS, Less, Sass, Stylus, JavaScript...
Customize... Disable All
Version Controls
Git, GitHub, Mercurial, Perforce, Subversion
Customize... Disable All
Test Tools
JUnit, TestNG, Cucumber for Java, Coverage
Customize... Disable All
Application Servers
Application Servers View, Geronimo, GlassFish, WildFly...
Customize... Disable All
Clouds
Cloud Foundry, Heroku,
Swing
UI Designer
Android
Android
Skip Remaining and Set Defaults
Back to UI Themes
Next: Featured plugins


Customize IntelliJ IDEA
UI Themes → Default plugins → Featured plugins
Download featured plugins
We have a few plugins in our repository that most users like to download. Perhaps, you need them too?
Scala
Custom Languages
Plugin for Scala language support
Install
Live Edit Tool
Web Development
Provides live edit HTML/CSS/JavaScript
Install
IdeaVim
Editor
Emulates Vim editor
Recommended only if you are familiar with Vim.
Install and Enable
IDE Features Trainer
Code tools
Learn basic shortcuts and essential IDE features with quick interactive exercises
Install
New plugins can also be downloaded in Settings | Plugins
Skip Remaining and Set Defaults
Back to Default plugins
Start using IntelliJ IDEA

6. Vous devriez accéder à l'écran d'accueil d'IntelliJ IDEA. Vous allez récupérer le contenu de votre dépôt. Pour cela, cliquez sur `Check out from version control` puis sur `git`dans le menu qui apparaît.

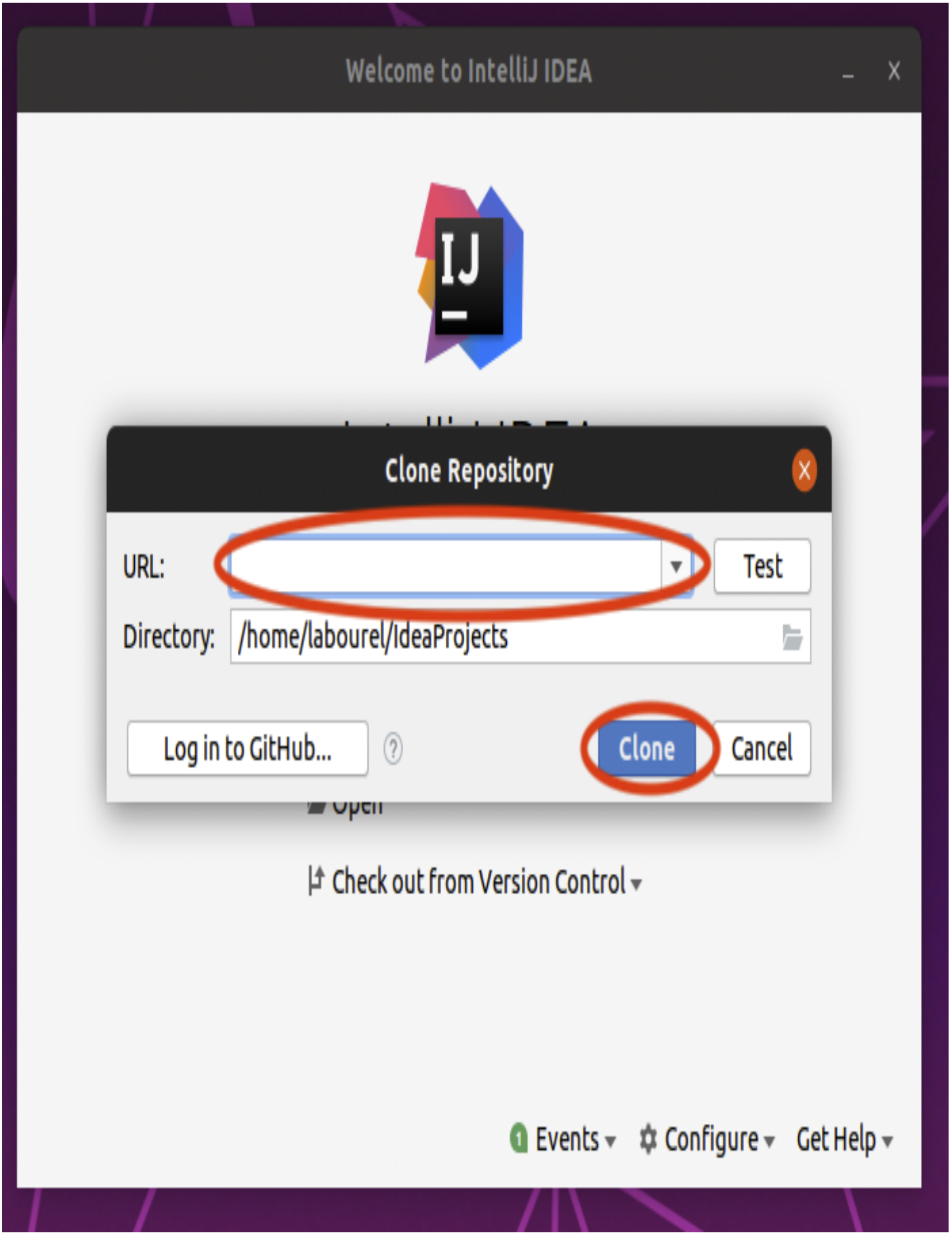

Welcome to IntelliJ IDEA
IntelliJ IDEA
Version 2019.2.2
Create New Project
Import Project
Open
Check out from Version Control
Events
Configure
Get Help


Welcome to IntelliJ IDEA
IntelliJ IDEA
Version 2019.2.2
Create New Project
Import Project
Open
Check out from Version Control
Git
Mercurial
Subversion
Events
Configure
Get Help

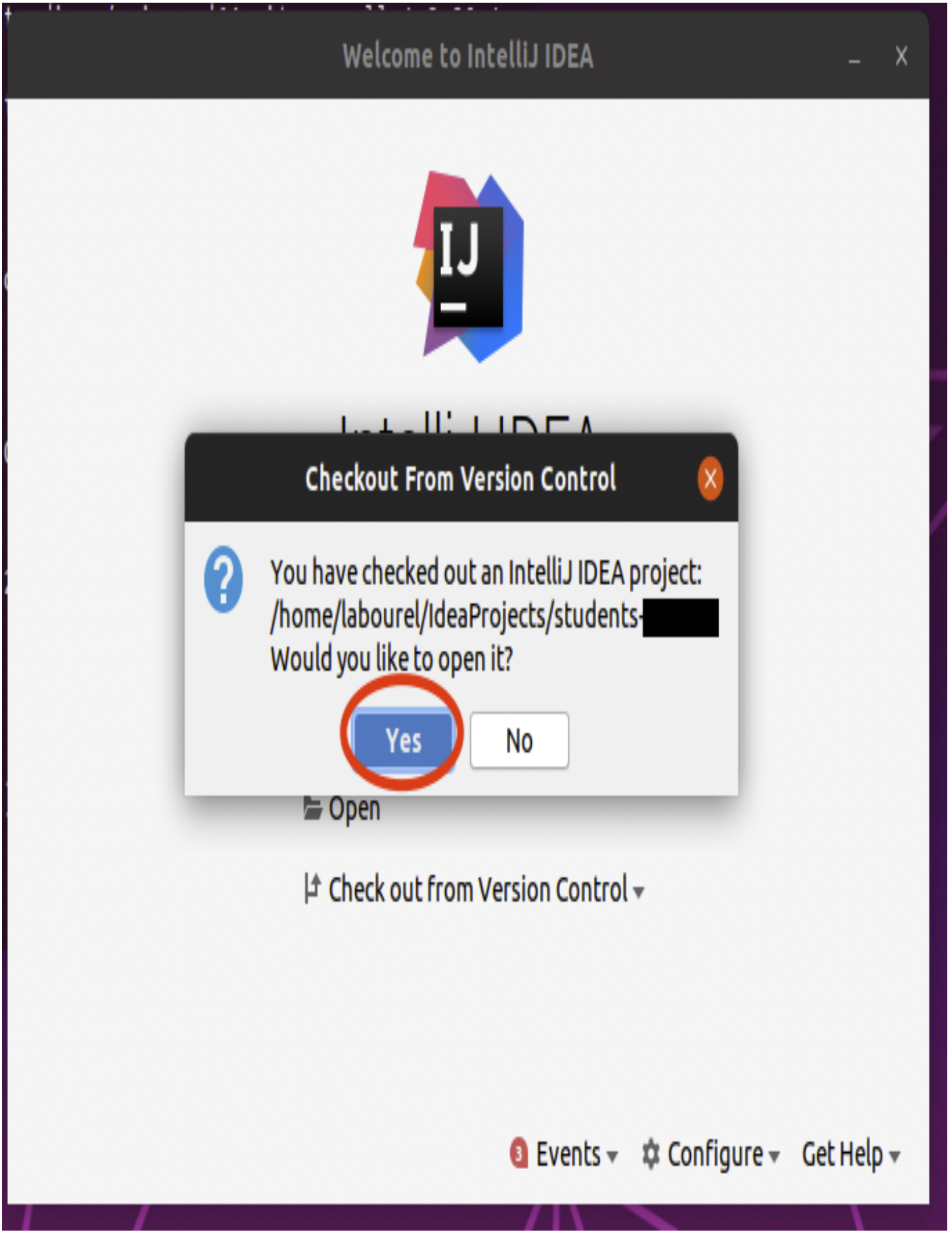
7. Vous allez maintenant récupérer votre dépôt (en fait le cloner). Pour cela, connectez-vous à votre compte github en cliquant sur le bouton `log in to github`. Vous devriez obtenir la fenêtre suivante dans laquelle vous devez remplir votre identifiant et mot de passe de votre compte github puis cliquer sur `Log in`.


Welcome to IntelliJ IDEA
Log In to GitHub
Server:
github.com
Login:
Password:
Password is not saved and used only to acquire GitHub token. Enter token
Sign up for GitHub
Log In
Cancel
URL:
Test
Cancel
Events
Configure
Get Help

Entrez dans le champs URL, l'adresse de votre dépôt (celle que vous récupérer sur la page web de votre dépôt en cliquant sur le bouton `clone or download`) puis cliquez sur le bouton `clone`.


Welcome to IntelliJ IDEA
Clone Repository
URL:
Test
Directory:
/home/labourel/IdeaProjects
Log in to GitHub...
Clone
Cancel
Check out from Version Control
Events
Configure
Get Help

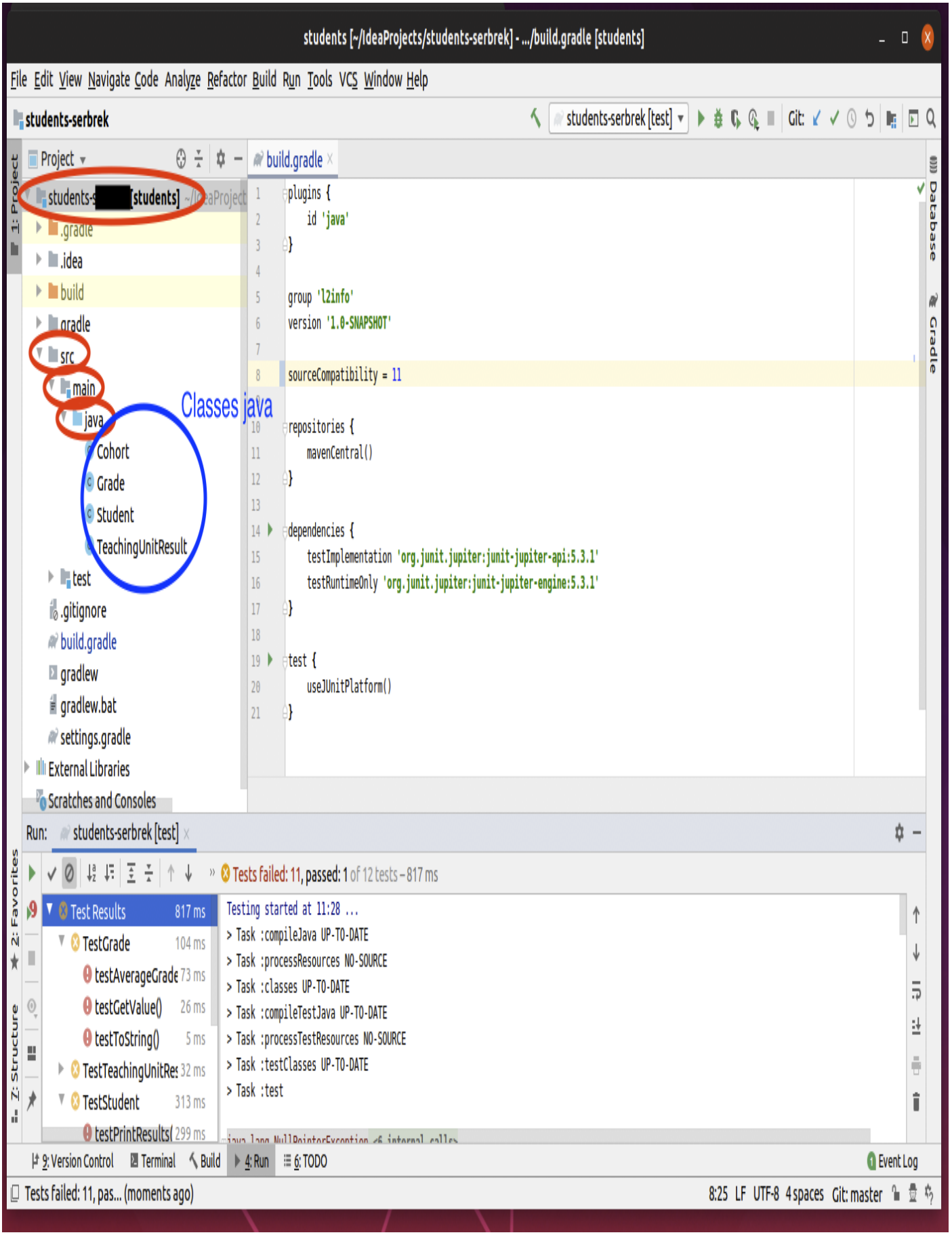
8. Après quelques instants de chargement, on vous demandera si vous souhaitez ouvrir votre projet. Il faut bien évidemment cliquer sur le bouton `yes`.


Welcome to IntelliJ IDEA
Checkout From Version Control
You have checked out an IntelliJ IDEA project:
/home/labourel/IdeaProjects/students-[redacted]
Would you like to open it?
Yes
No
Open
Check out from Version Control
Events
Configure
Get Help

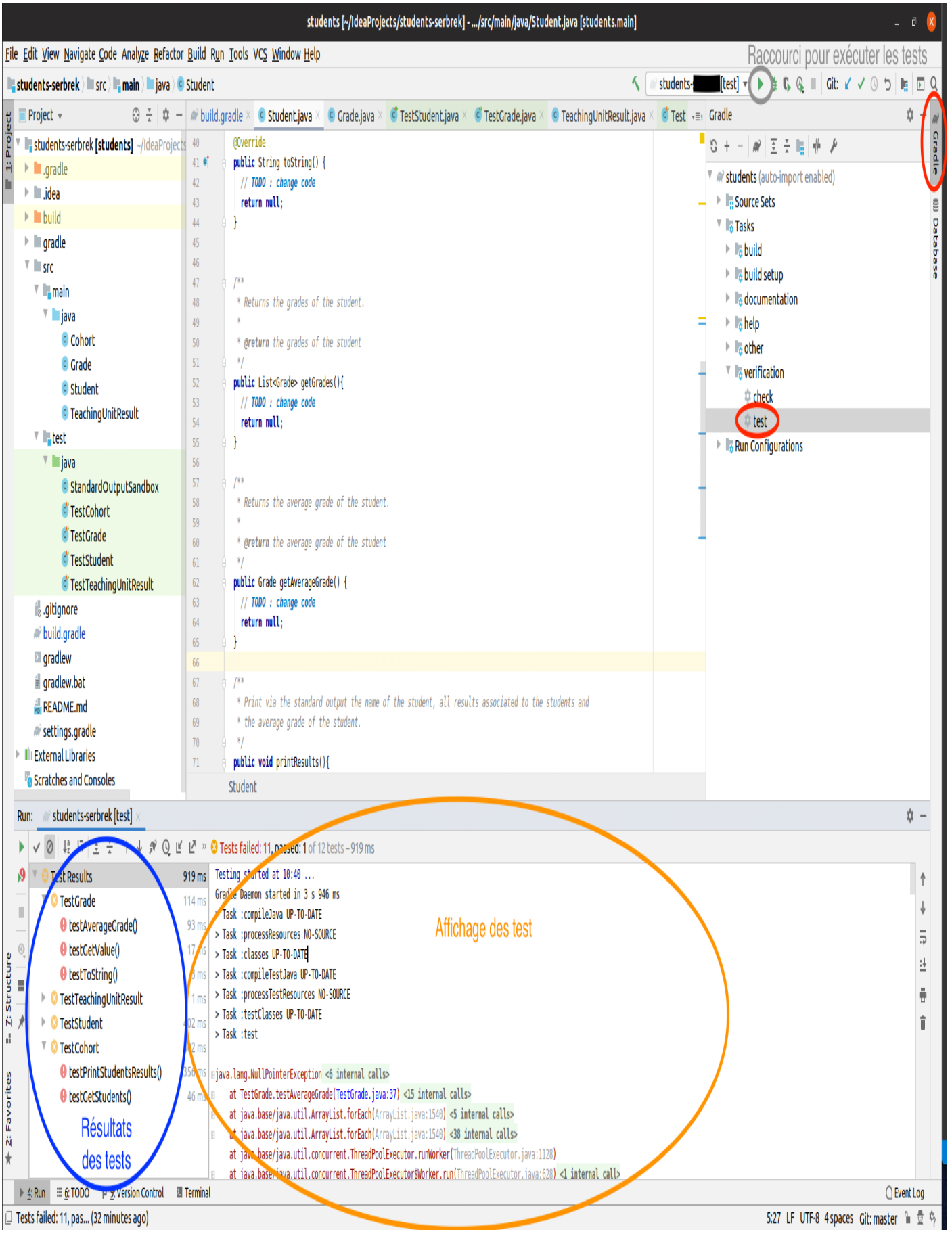
9. Une fois le projet ouvert, vous pouvez fermer la fenêtre `tip of the day`. Il vous faut autoriser l'import gradle en cliquant sur le pop up en bas à droite de votre fenêtre.

1.3.3 Prise en main d'IntelliJ IDEA

1. Pour accéder aux fichiers `java` de votre dépôt il vous faut naviguer dans l'arborescence du projet. Les fichiers que vous devez modifier sont dans `students-***` -> `src` -> `main` -> `java` (il faut cliquer sur les triangles pour déplier l'arborescence).


students [~/IdeaProjects/students-serbrek] - .../build.gradle [students]
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
students-serbrek
students-serbrek [test]
Git:
Project
build.gradle
students-[redacted] [students] ~/IdeaProject
.gradle
.idea
build
gradle
src
main
java
Classes java
Cohort
Grade
Student
TeachingUnitResult
test
.gitignore
build.gradle
gradlew
gradlew.bat
settings.gradle
External Libraries
Scratches and Consoles
plugins {
id 'java'
}
group 'l2info'
version '1.0-SNAPSHOT'
sourceCompatibility = 11
repositories {
mavenCentral()
}
dependencies {
testImplementation 'org.junit.jupiter:junit-jupiter-api:5.3.1'
testRuntimeOnly 'org.junit.jupiter:junit-jupiter-engine:5.3.1'
}
test {
useJUnitPlatform()
}
Run: students-serbrek [test]
Tests failed: 11, passed: 1 of 12 tests – 817 ms
Test Results 817 ms
TestGrade 104 ms
testAverageGrade 73 ms
testGetValue() 26 ms
testToString() 5 ms
TestTeachingUnitRes 32 ms
TestStudent 313 ms
testPrintResults(299 ms
Testing started at 11:28 ...
> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE
> Task :compileTestJava UP-TO-DATE
> Task :processTestResources NO-SOURCE
> Task :testClasses UP-TO-DATE
> Task :test
9: Version Control
Terminal
Build
4: Run
6: TODO
Event Log
Tests failed: 11, pas... (moments ago)
8:25 LF UTF-8 4 spaces Git: master
1: Project
2: Favorites
Z: Structure
Database
Gradle

2. Pour compiler et exécuter le programme, il faut passer par l'onglet gradle à droite et cliquer deux fois sur `students` -> `Tasks` -> `verification` -> `test`. Cela va compiler et exécuter les tests. Pour le moment, les tests ne passeront pas car certaines classes sont incomplètes.


Raccourci pour exécuter les tests
Affichage des test
Résultats
des tests

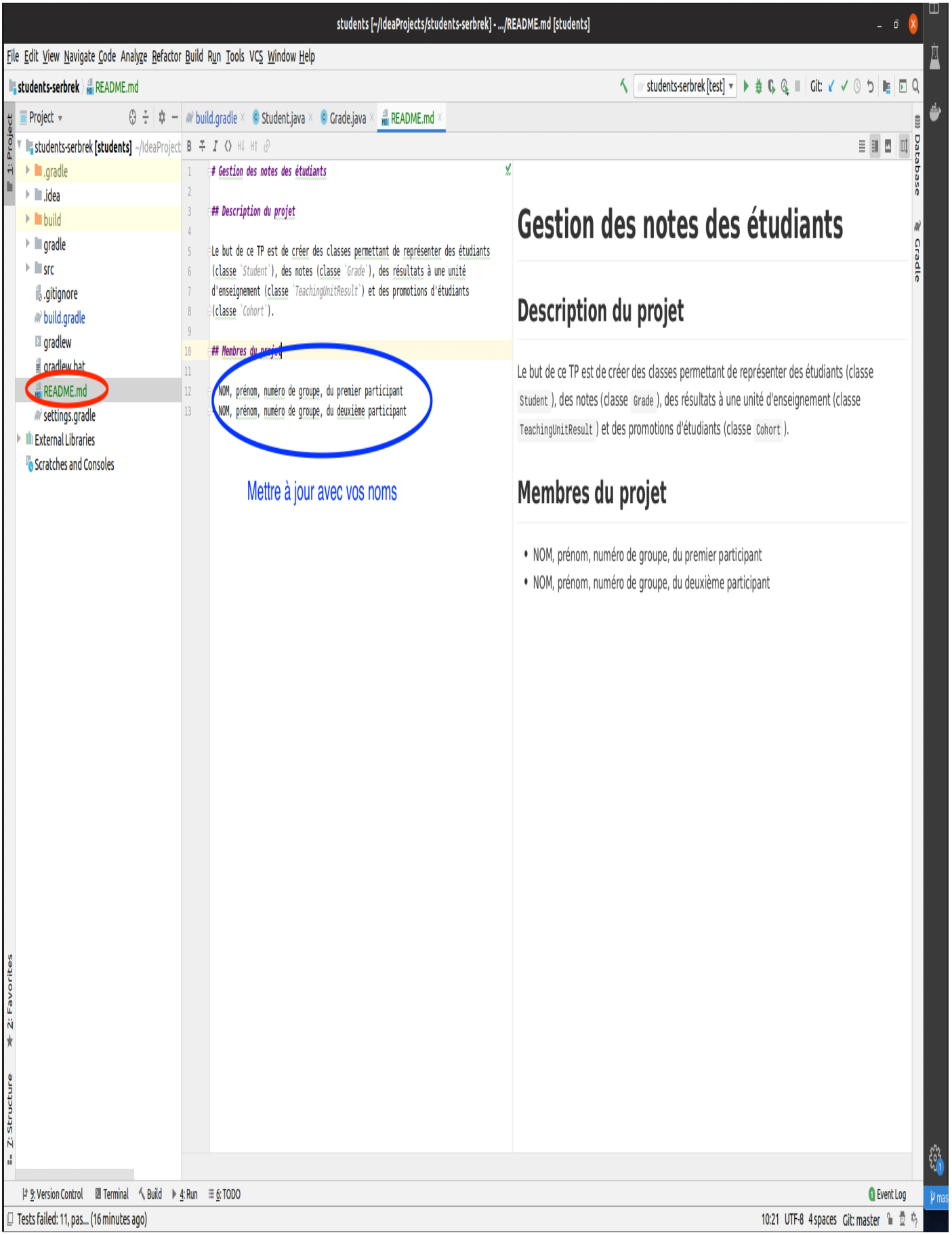
1.4 Git

1.4.1 Principe de git

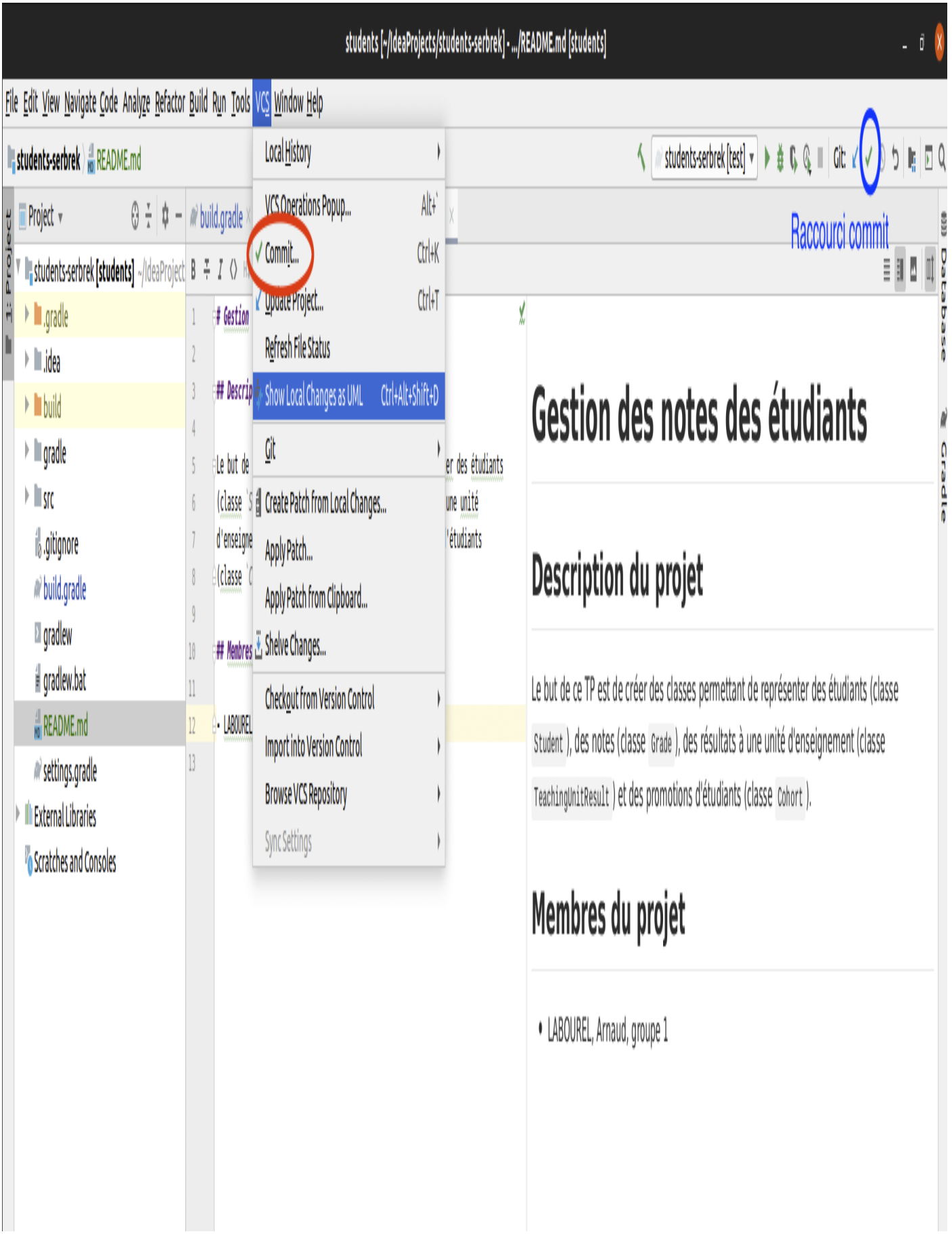
Le principe de `git` est d'avoir un *dépot* distant : une version de votre projet stockée sur un serveur accessible par Internet (en l'occurrence hébergé par `github`). Vous disposez en plus de dépots locaux sur les ordinateurs sur lesquels vous travaillez. Vous faites vos modifications sur votre ordinateur, et lorsque vous avez accompli une amélioration qui fonctionne bien, vous pouvez la faire enregistrer (`commit`) par `git`. Ces enregistrements sont locaux à votre ordinateur, et vous pouvez en faire autant que vous le souhaitez. Si vous voulez partager votre travail avec votre équipe, il vous faut l'envoyer vers le dépot distant (`push`). À l'inverse, si vous souhaitez récupérer le travail fait par vos co-équipiers, il faut ramener ces modifications depuis le dépot distant (`pull`). IntelliJ est capable de gérer `git` ; vous trouverez dans le menu `VCS` l'option `Commit`, et l'option `Git` qui contient `push` et `pull`.

1.4.2 Première modification de votre dépôt

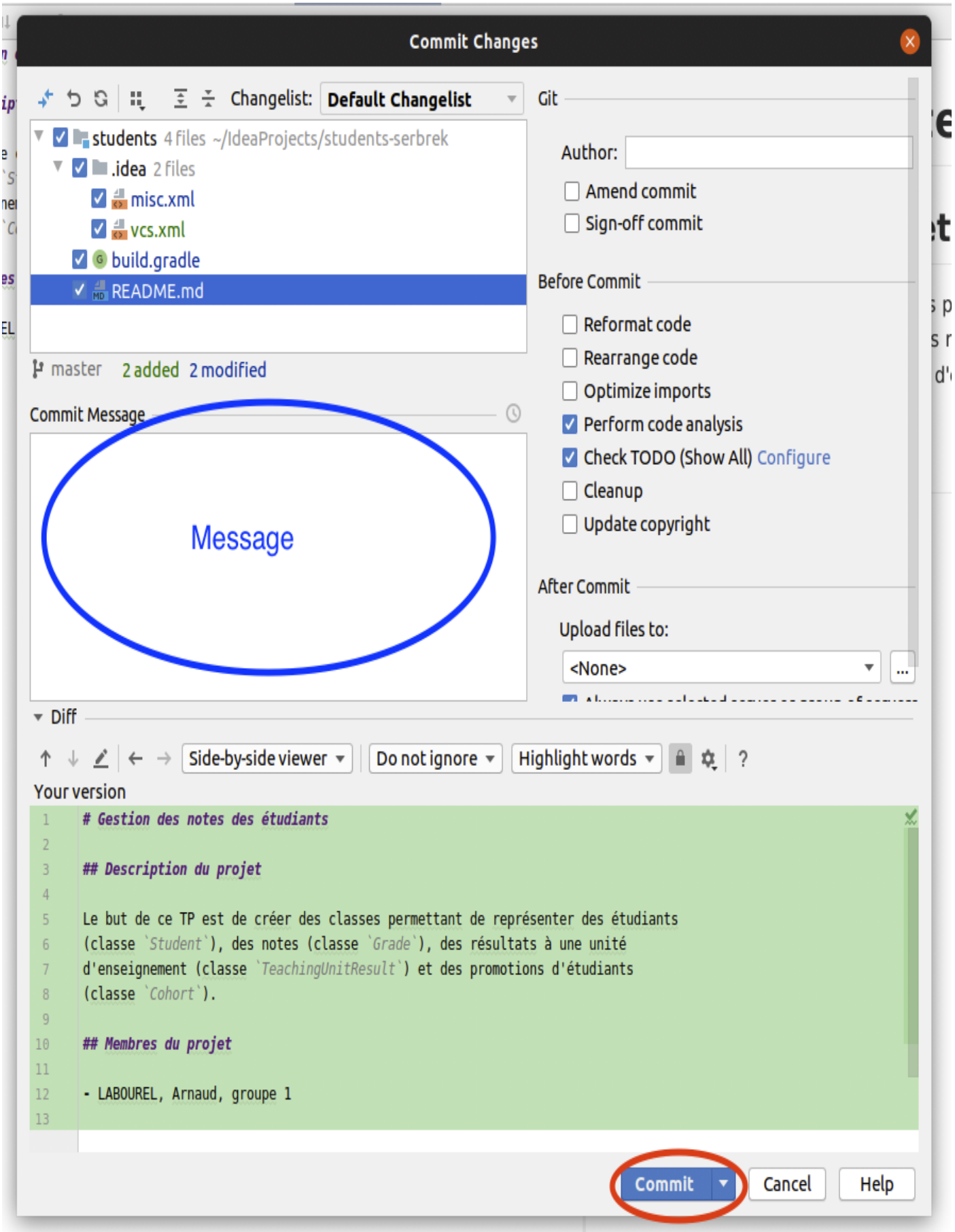
1. Modifiez le fichier `README.md`. Mettez votre nom ainsi que le nom de votre éventuel co-équipier (si vous êtes seul enlever la deuxième ligne de participants).


students [~/IdeaProjects/students-serbrek] - .../README.md [students]
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
students-serbrek README.md
students-serbrek [test]
Git:
Project
build.gradle
Student.java
Grade.java
README.md
students-serbrek [students] ~/IdeaProject
.gradle
.idea
build
gradle
src
.gitignore
build.gradle
gradlew
gradlew.bat
README.md
settings.gradle
External Libraries
Scratches and Consoles
Gestion des notes des étudiants
Description du projet
Le but de ce TP est de créer des classes permettant de représenter des étudiants (classe `Student`), des notes (classe `Grade`), des résultats à une unité d'enseignement (classe `TeachingUnitResult`) et des promotions d'étudiants (classe `Cohort`).
Membres du projet
NOM, prénom, numéro de groupe, du premier participant
NOM, prénom, numéro de groupe, du deuxième participant
Mettre à jour avec vos noms
Gestion des notes des étudiants
Description du projet
Le but de ce TP est de créer des classes permettant de représenter des étudiants (classe Student), des notes (classe Grade), des résultats à une unité d'enseignement (classe TeachingUnitResult) et des promotions d'étudiants (classe Cohort).
Membres du projet
NOM, prénom, numéro de groupe, du premier participant
NOM, prénom, numéro de groupe, du deuxième participant
9: Version Control
Terminal
Build
4: Run
6: TODO
Event Log
Tests failed: 11, pas... (16 minutes ago)
10:21 UTF-8 4 spaces Git: master

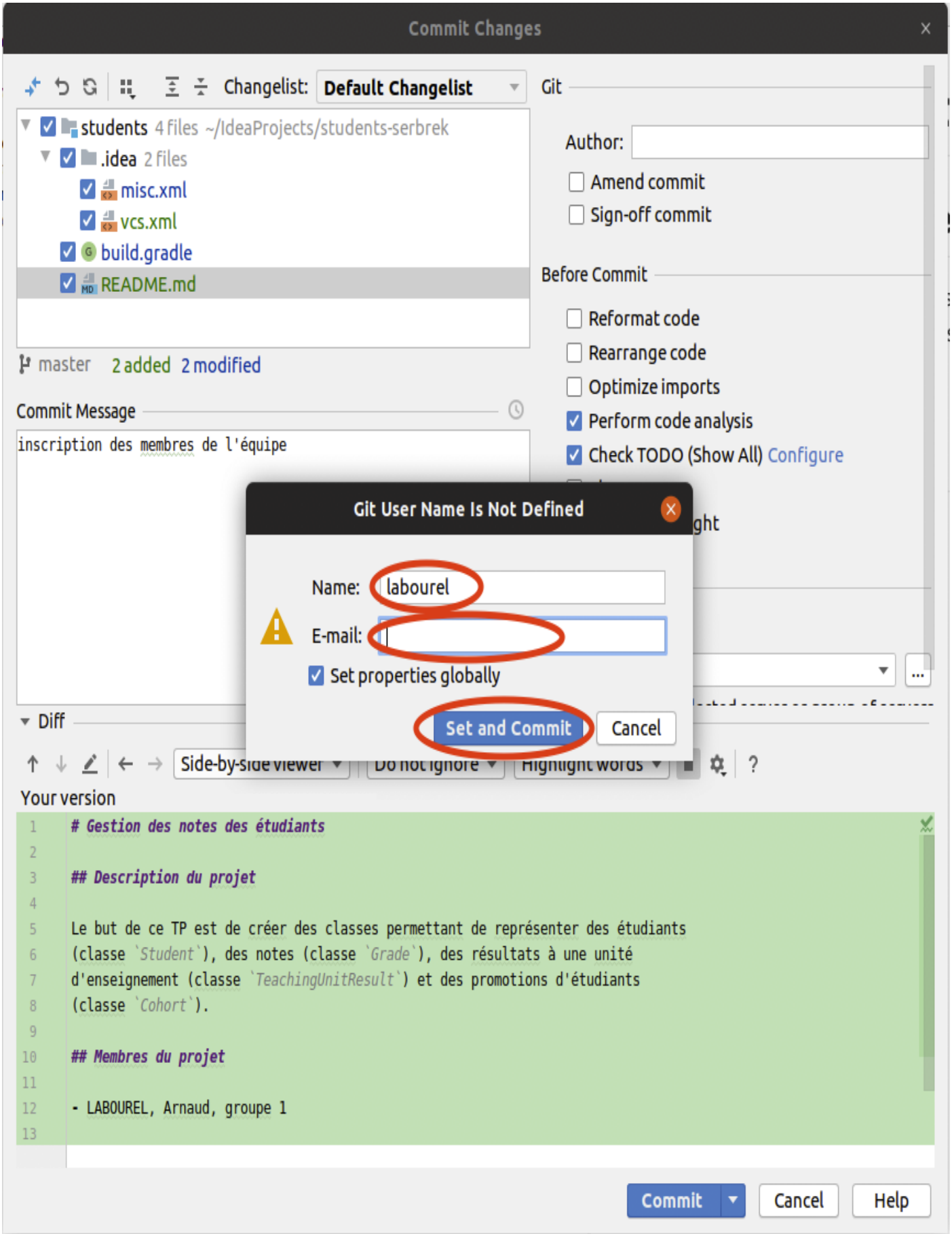
2. Vous allez maintenant mettre à jour votre dépôt local (celui sur votre machine) en effectuant un `commit`. Pour cela vous pouvez aller dans le menu `VCS` -> `commit` ou bien cliquer sur le raccourci en haut à droite de votre fenêtre.


students [~/IdeaProjects/students-serbrek] - .../README.md [students]
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
students-serbrek README.md
students-serbrek [test]
Git:
Raccourci commit
Local History
VCS Operations Popup... Alt+`
Commit... Ctrl+K
Update Project... Ctrl+T
Refresh File Status
Show Local Changes as UML Ctrl+Alt+Shift+D
Git
Create Patch from Local Changes...
Apply Patch...
Apply Patch from Clipboard...
Shelve Changes...
Checkout from Version Control
Import into Version Control
Browse VCS Repository
Sync Settings
Project
build.gradle
students-serbrek [students]
.gradle
.idea
build
gradle
src
.gitignore
build.gradle
gradlew
gradlew.bat
README.md
settings.gradle
External Libraries
Scratches and Consoles
Gestion des notes des étudiants
Description du projet
Le but de ce TP est de créer des classes permettant de représenter des étudiants (classe Student), des notes (classe Grade), des résultats à une unité d'enseignement (classe TeachingUnitResult) et des promotions d'étudiants (classe Cohort).
Membres du projet
LABOUREL, Arnaud, groupe 1

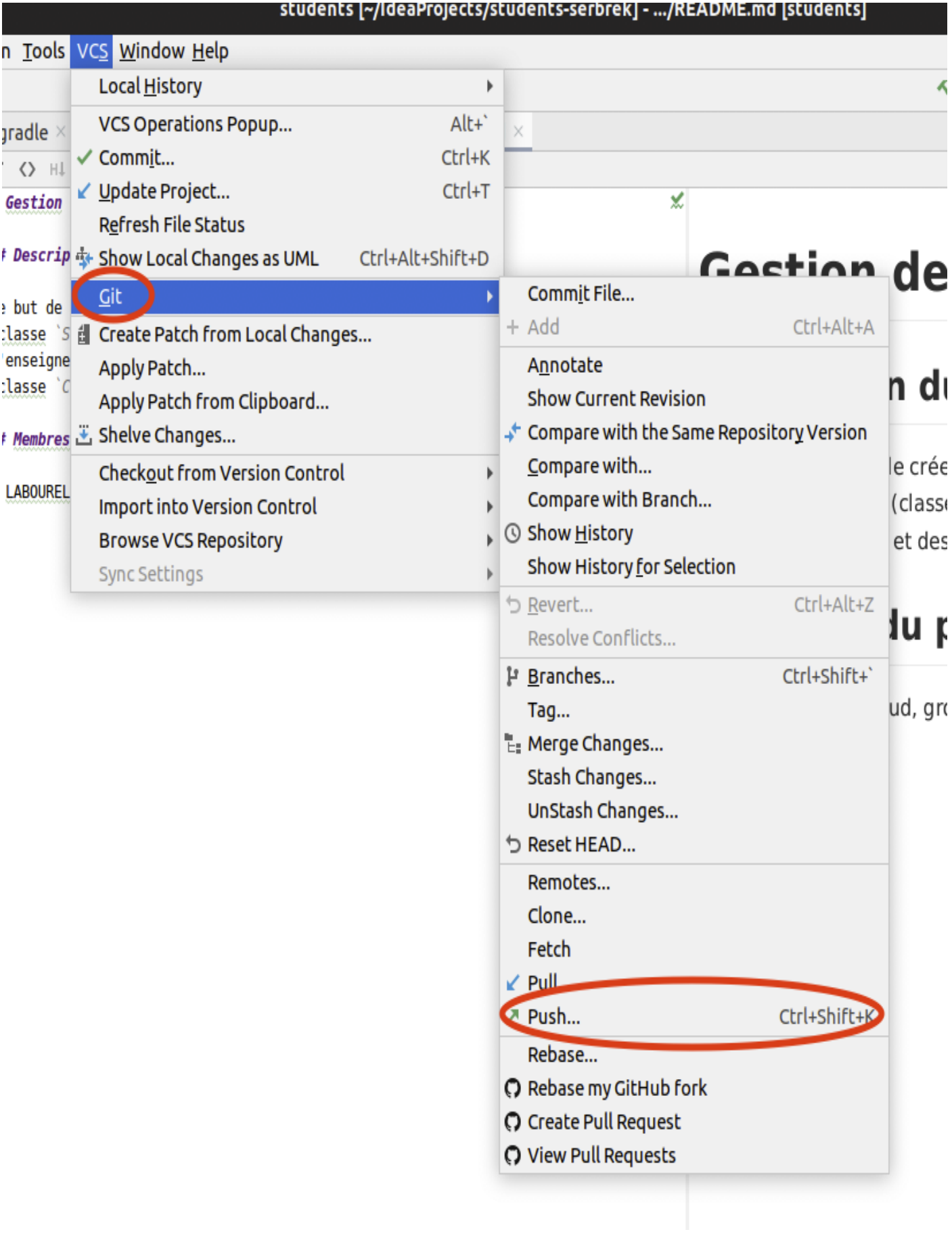
Faites un `commit` avec pour message “inscription des membres de l’équipe” en cliquant sur commit après avoir rempli le champs message.


Commit Changes
Changelist: Default Changelist
Git
students 4 files ~/IdeaProjects/students-serbrek
.idea 2 files
misc.xml
vcs.xml
build.gradle
README.md
Author:
Amend commit
Sign-off commit
Before Commit
Reformat code
Rearrange code
Optimize imports
Perform code analysis
Check TODO (Show All) Configure
Cleanup
Update copyright
master 2 added 2 modified
Commit Message
Message
After Commit
Upload files to:
<None>
Diff
Side-by-side viewer
Do not ignore
Highlight words
Your version
Gestion des notes des étudiants
Description du projet
Le but de ce TP est de créer des classes permettant de représenter des étudiants
(classe `Student`), des notes (classe `Grade`), des résultats à une unité
d'enseignement (classe `TeachingUnitResult`) et des promotions d'étudiants
(classe `Cohort`).
Membres du projet
- LABOUREL, Arnaud, groupe 1
Commit
Cancel
Help

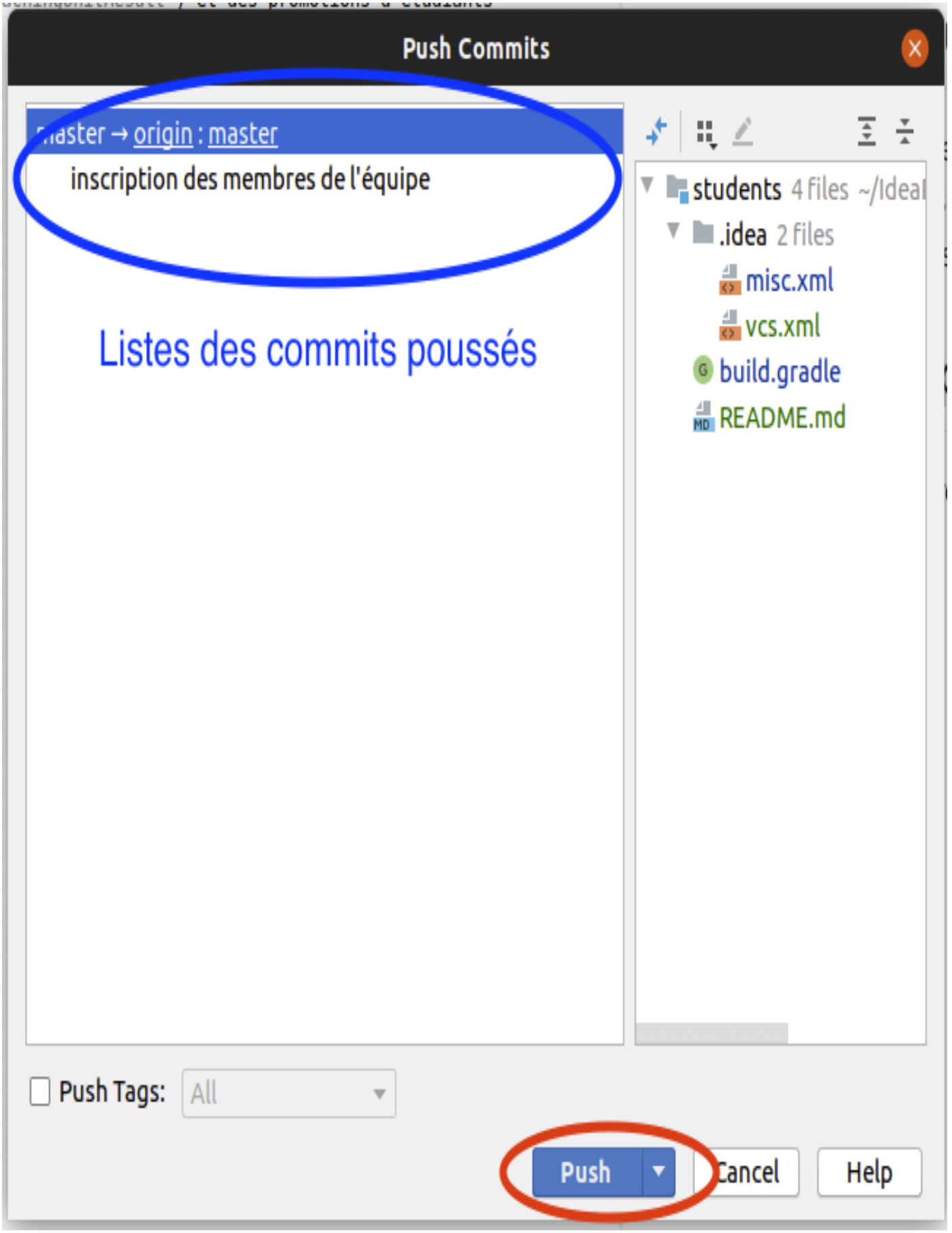
Préciser votre nom et email et cliquez sur le bouton `Set and Commit`.


Commit Changes
Changelist: Default Changelist
Git
students 4 files ~/IdeaProjects/students-serbrek
.idea 2 files
misc.xml
vcs.xml
build.gradle
README.md
Author:
Amend commit
Sign-off commit
Before Commit
Reformat code
Rearrange code
Optimize imports
Perform code analysis
Check TODO (Show All) Configure
master 2 added 2 modified
Commit Message
inscription des membres de l'équipe
Git User Name Is Not Defined
Name: labourel
E-mail:
Set properties globally
Set and Commit
Cancel
Diff
Side-by-side viewer
Do not ignore
Highlight words
Your version
Gestion des notes des étudiants
Description du projet
Le but de ce TP est de créer des classes permettant de représenter des étudiants
(classe `Student`), des notes (classe `Grade`), des résultats à une unité
d'enseignement (classe `TeachingUnitResult`) et des promotions d'étudiants
(classe `Cohort`).
Membres du projet
- LABOUREL, Arnaud, groupe 1
Commit
Cancel
Help

3. Vous avez mis à jour votre dépôt local (sur votre machine) mais pas le dépôt distant (celui sur les serveurs de github). Pour cela, il faut faire un `push`. Allez dans le menu `VCS` -> `git` -> `push`.

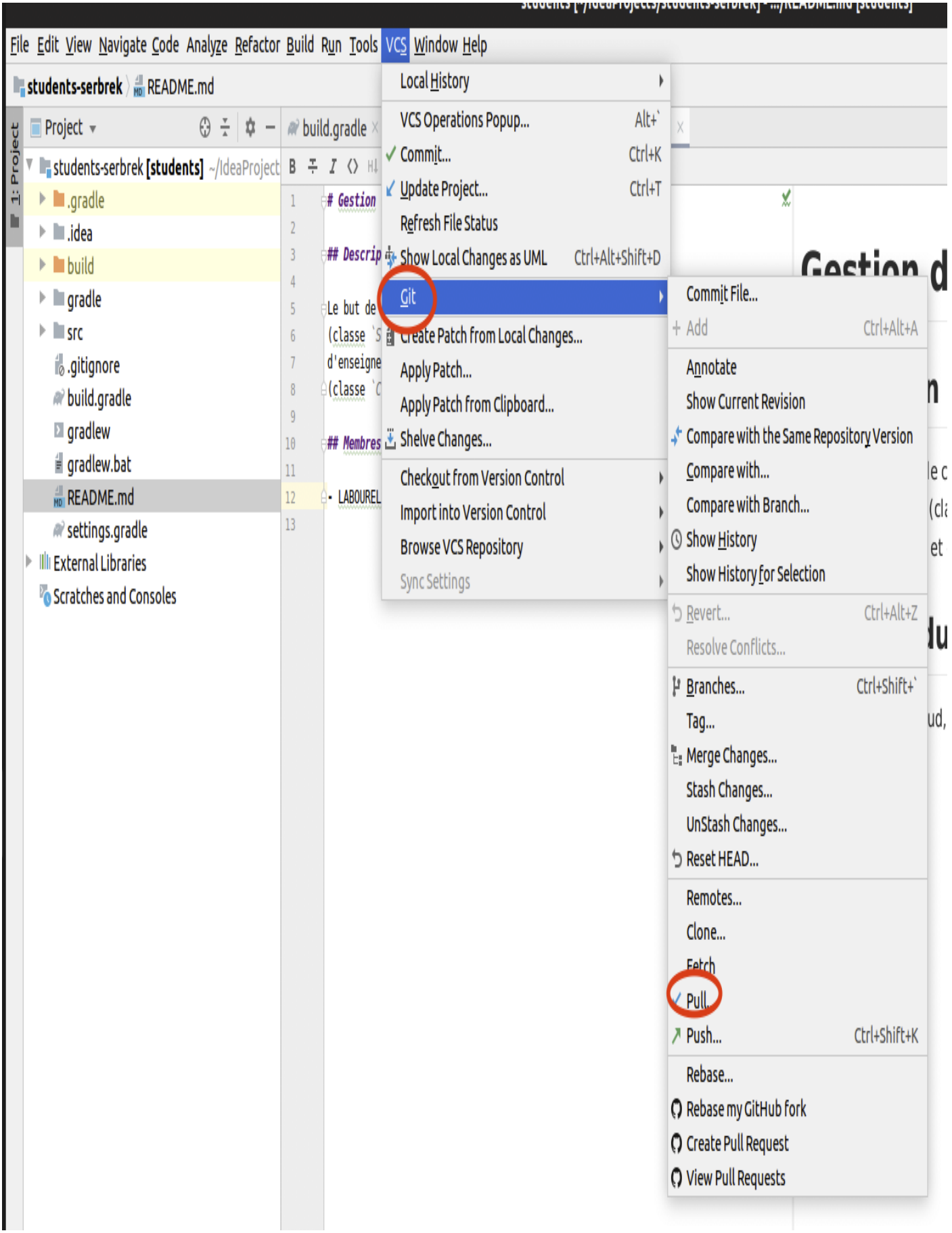

students [~/IdeaProjects/students-serbrek] - .../README.md [students]
Tools VCS Window Help
Local History
VCS Operations Popup... Alt+`
Commit... Ctrl+K
Update Project... Ctrl+T
Refresh File Status
Show Local Changes as UML Ctrl+Alt+Shift+D
Git
Create Patch from Local Changes...
Apply Patch...
Apply Patch from Clipboard...
Shelve Changes...
Checkout from Version Control
Import into Version Control
Browse VCS Repository
Sync Settings
Commit File...
Add Ctrl+Alt+A
Annotate
Show Current Revision
Compare with the Same Repository Version
Compare with...
Compare with Branch...
Show History
Show History for Selection
Revert... Ctrl+Alt+Z
Resolve Conflicts...
Branches... Ctrl+Shift+`
Tag...
Merge Changes...
Stash Changes...
UnStash Changes...
Reset HEAD...
Remotes...
Clone...
Fetch
Pull
Push... Ctrl+Shift+K
Rebase...
Rebase my GitHub fork
Create Pull Request
View Pull Requests

Cliquez sur le bouton push de la fenêtre qui vient d'apparaître. Vous pouvez voir la liste des commits que vous vous apprêtez à pousser.


Push Commits
master → origin : master
inscription des membres de l'équipe
Listes des commits poussés
students 4 files ~/Ideal
.idea 2 files
misc.xml
vcs.xml
build.gradle
README.md
Push Tags:
All
Push
Cancel
Help

Si tout se passe bien un popup `push successful` devrait apparaître en bas à droite de votre fenêtre.

4. Si jamais vous avez besoin de récupérer le projet sur le serveur (par exemple après un push de votre camarade), il vous suffit de faire un pull. Allez dans le menu `VCS` -> `git` -> `pull`.


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
students-serbrek README.md
Project
build.gradle
students-serbrek [students] ~/IdeaProject
.gradle
.idea
build
gradle
src
.gitignore
build.gradle
gradlew
gradlew.bat
README.md
settings.gradle
External Libraries
Scratches and Consoles
1: Project
Gestion
Descrip
Le but de
(classe `S
d'enseigne
(classe `C
Membres
- LABOUREL
Local History
VCS Operations Popup... Alt+`
Commit... Ctrl+K
Update Project... Ctrl+T
Refresh File Status
Show Local Changes as UML Ctrl+Alt+Shift+D
Git
Create Patch from Local Changes...
Apply Patch...
Apply Patch from Clipboard...
Shelve Changes...
Checkout from Version Control
Import into Version Control
Browse VCS Repository
Sync Settings
Commit File...
Add Ctrl+Alt+A
Annotate
Show Current Revision
Compare with the Same Repository Version
Compare with...
Compare with Branch...
Show History
Show History for Selection
Revert... Ctrl+Alt+Z
Resolve Conflicts...
Branches... Ctrl+Shift+`
Tag...
Merge Changes...
Stash Changes...
UnStash Changes...
Reset HEAD...
Remotes...
Clone...
Fetch
Pull...
Push... Ctrl+Shift+K
Rebase...
Rebase my GitHub fork
Create Pull Request
View Pull Requests

1.5 Correction du programme

1.5.1 Méthodologie pour le TP

Vous allez maintenant pouvoir attaquer la correction du programme java du dépôt. Pour celà vous allez devoir respecter les consignes suivantes :

— À chaque modification de programme, faites un `commit` en sélectionnant les fichiers modifiés. Chaque `commit` doit contenir un message précisant la nature des modifications effectuées.
— À chaque tâche terminée, faites un `push` de votre travail.
— Ceci est le minimum. Vous pouvez faire plus de `commit` et plus de `push`, ainsi que des `pull` pour récupérer le travail de votre co-équipier.
— Si vous avez un problème et souhaitez l'aide de votre instructeur en dehors des séances, un `push` lui permet de voir votre programme.

1.5.2 Tâche 1 : classe `Grade`

documentation de la classe

Cette classe va permettre de représenter une note obtenue par un étudiant. Une note est une valeur flottante comprise entre 0 et 20.

Cette classe contient les éléments suivants qui sont corrects :

— `private static final int MAXIMUM_GRADE` : un attribut statique représentant la valeur de la note maximale qui est égal à 20.
— `private final double value` : la valeur de la note comprise entre 0 et `MAXIMUM_GRADE`.
— `public Grade(double value)` : constructeur évident.
— `public boolean equals(Object o)` : méthode permettant de tester l'égalité de deux notes.

Votre but est de compléter les instructions des méthodes suivantes :

— `public double getValue()` : retourne la valeur (`value`) de la note.
— `String toString()` : retourne une représentation de la note sous forme de chaîne de caractères. Pour une note ayant une valeur 12, cette méthode devra retourner la chaîne de caractères : `"12.0/20"`.
— `public static Grade averageGrade(List<Grade> grades)` : calcule et renvoie la moyenne d'une liste de notes.

Assurez-vous que votre classe est correcte en exécutant à nouveau les tests et en vérifiant que votre classe passe les tests `testToString`, `testGetValue` et `testAverageGrade` de `TestGrade` avec succès.

1.5.3 Tâche 2 : classe `TeachingUnitResult`

documentation de la classe

Cette classe va nous permettre de représenter un résultat obtenu par un étudiant, c'est-à-dire une note associée à une Unité d'Enseignement (UE).

Cette classe contient les éléments suivants qui sont corrects :

— `private final String teachingUnitName` : le nom de l'unité d'enseignement du résultat.
— `private final Grade grade` : la note du résultat.
— `public TeachingUnitResult(String teachingUnitName, Grade grade)` : constructeur évident.
— `public boolean equals(Object o)` : méthode permettant de tester l'égalité de deux résultats.

Votre but est de compléter les instructions des méthodes suivantes :

— `public Grade getGrade()` : retourne la note associée au résultat.
— `public String toString()` : renvoie le nom de l'unité d'enseignement suivi de `" : "` suivi de la représentation en chaîne de caractère de la note. Par exemple, un résultat d'une UE de Programmation 2 avec une note de 20 devra renvoyer la chaîne de caractères suivante : `"Programmation 2 : 20.0/20"`.

Assurez-vous que votre classe est correct en exécutant à nouveau les tests et en vérifiant que votre classe passe les tests `testToString` et `testGetGrade` de `TestTeachingUnitResult` avec succès.

1.5.4 Tâche 3 : classe `Student`

documentation de la classe

Cette classe va nous permettre de représenter un étudiant.

Cette classe contient les éléments suivants qui sont corrects :

— `private final String firstName` : le prénom de l'étudiant.
— `private final String lastName` : le nom de famille de l'étudiant.
— `private final List<TeachingUnitResult> results` : les résultats de l'étudiant.
— `public Student(String firstName, String lastName)` : constructeur initialisant le nom et prénom de l'étudiant avec les valeurs données et créant une liste vide pour les résultats.
— `public boolean equals(Object o)` : méthode permettant de tester l'égalité de deux étudiants.

Votre but est de compléter les instructions des méthodes suivantes :

— `public void addResult(String teachingUnitName, Grade grade)` : ajoute un nouveau résultat à partir du nom de l'UE et d'une note.
— `public List<Grade> getGrades()` : renvoie la liste des notes associées aux résultats de l'étudiant.
— `public String toString()` : renvoie le nom de l'étudiant, c'est-à-dire son prénom, suivi d'un espace, suivi de son nom.
— `public Grade averageGrade()` : renvoie la moyenne des notes associés aux résultats de l'étudiant.
— `public void printResults()` : affiche les résultats de l'étudiant en sortie standard. Un étudiant nommé Arnaud Labourel et ayant 20 en Programmation 2 et en structures discrètes devra produire l'affichage suivant (avec un saut de ligne à la fin) :

```
Arnaud Labourel
Programmation 2 : 20.0/20
Structures discrètes : 20.0/20
Note moyenne : 20.0/20
```

Assurez-vous que votre classe est correcte en exécutant à nouveau les tests et en vérifiant que votre classe passe les tests `testToString`, `testGetGrades`, `testGetAverageGrade` et `testPrintResults` de `TestTeachingUnitResult` avec succès.

1.5.5 Tâche 4 : classe `Cohort`

documentation de la classe

Cette classe va nous permettre de représenter une promotion d'étudiants. La classe `Cohort` contiendra les attributs, méthodes et constructeurs suivants :

Cette classe contient les éléments suivants qui sont corrects :

— `private final String name` : le nom de la promotion
— `private final List<Student> students` : les étudiants de la promotion
— `public Cohort(String name)` : constructeur à partir du nom de la promotion et initialisant à vide la liste des étudiants

Votre but est de compléter les instructions des méthodes suivantes :

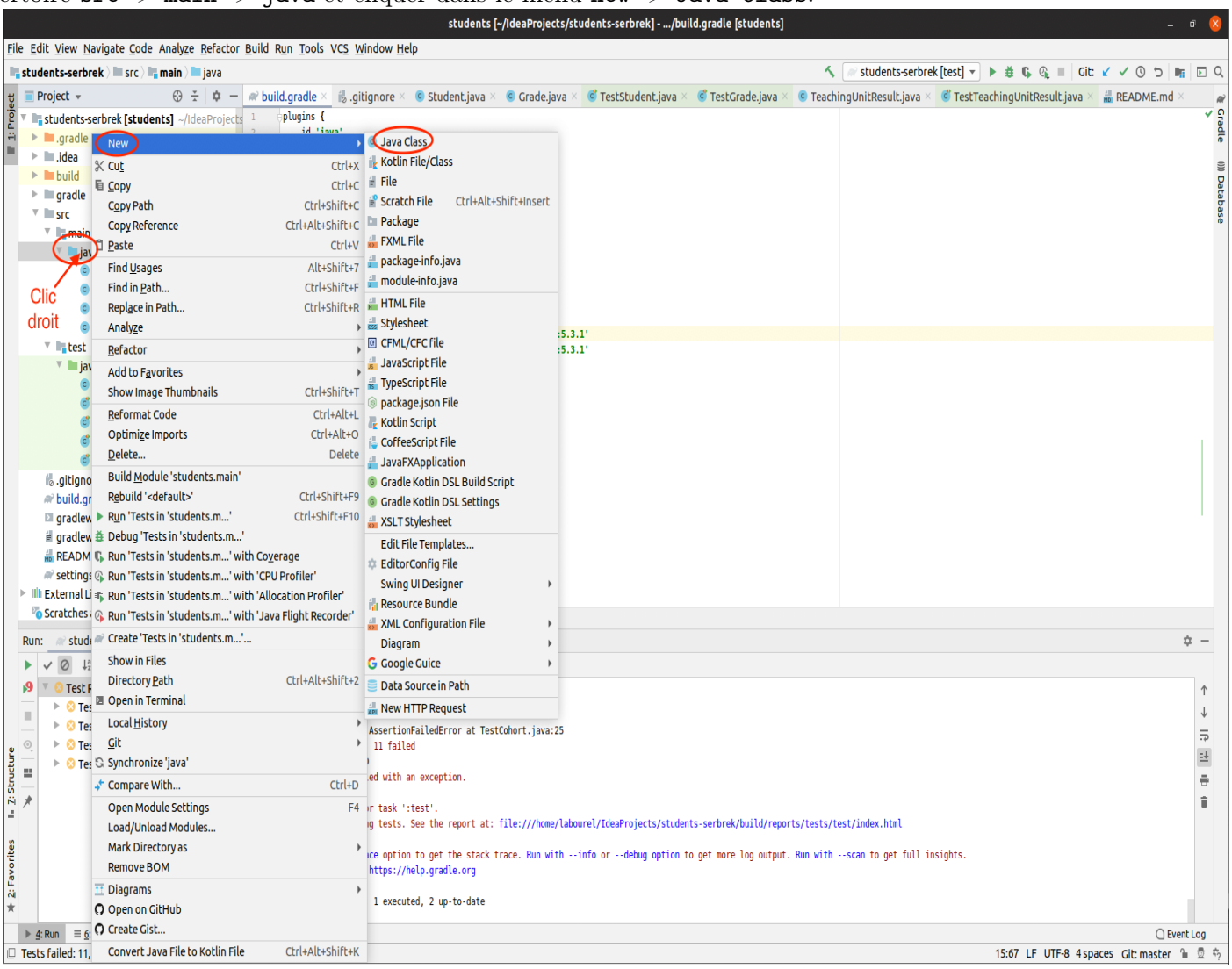
— `public void addStudent(Student student)` : ajoute un étudiant à la promotion.
— `public List<Student> getStudents()` : renvoie la liste des étudiants de la promotion.
— `String toString()` : retourne une représentation de la promotion correspondant à son nom.
— `public void printStudentsResults()` : affiche les résultats de l'étudiant en sortie standard. Une promotion ayant pour nom L2 informatique et deux étudiants devra produire l'affichage suivant (avec un saut de ligne à la fin)

```
L2 informatique

Paul Calcul
Programmation 2 : 10.0/20
Structures discrètes : 20.0/20
Note moyenne : 15.0/20

Pierre Kiroul
Programmation 2 : 10.0/20
Structures discrètes : 0.0/20
Note moyenne : 5.0/20
```

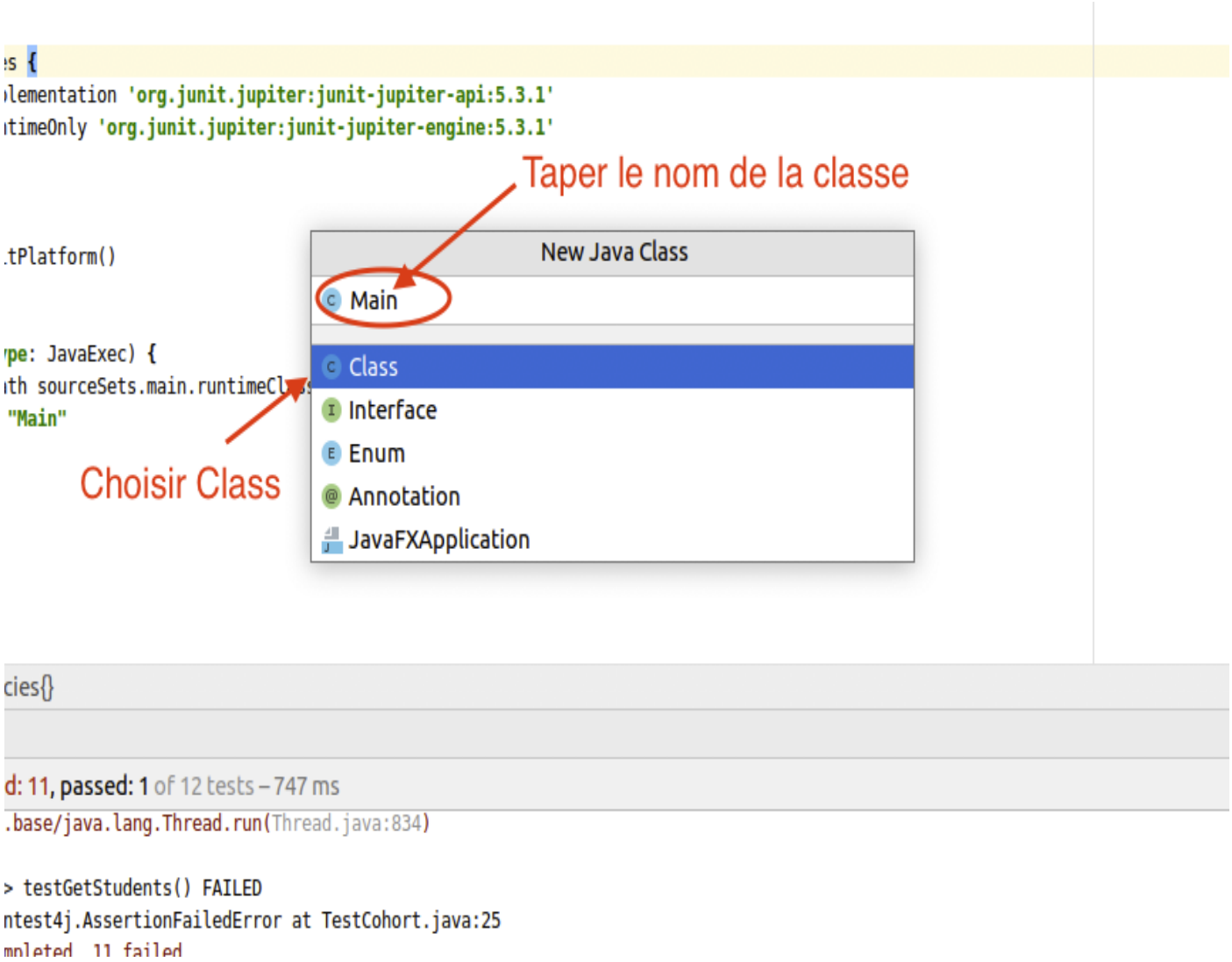
Créer la classe **`Cohort`** *avec les méthodes demandées.*

1.5.6 Tâche 5 : classe `Main`

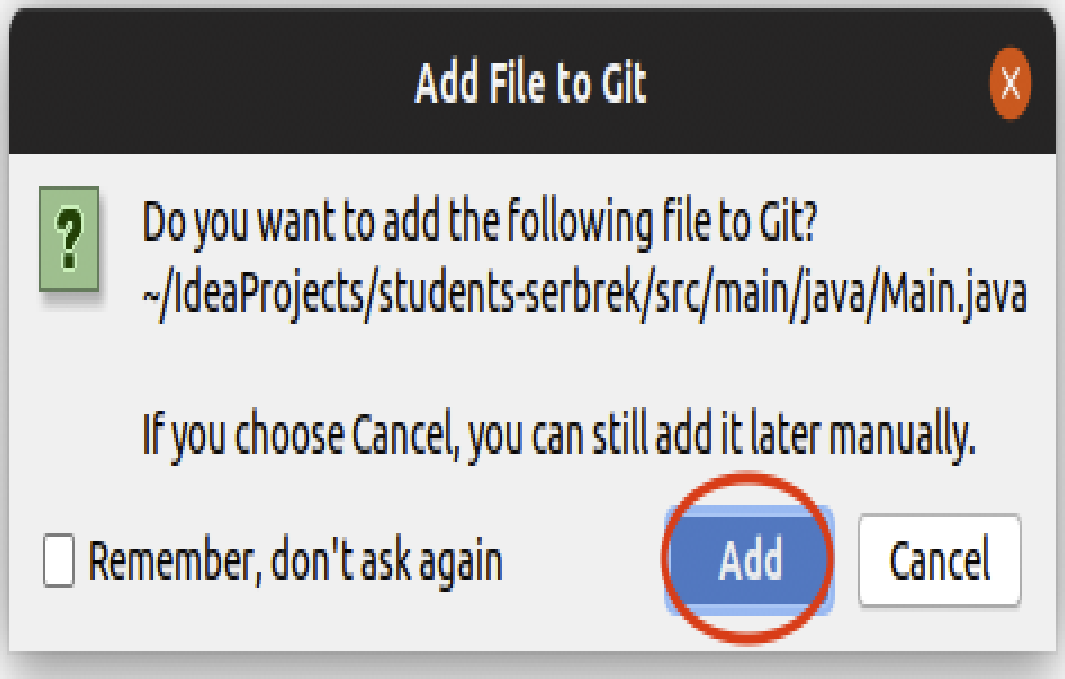
Vous allez maintenant ajouter une classe `Main` au projet. Pour cela, il vous faut faire un clic droit sur le répertoire `src` -> `main` -> `java` et cliquer dans le menu `new` -> `Java Class`.



Tapez le nom de la classe, choisissez bien que vous souhaitez créer un classe et validez en appuyant sur entrée.



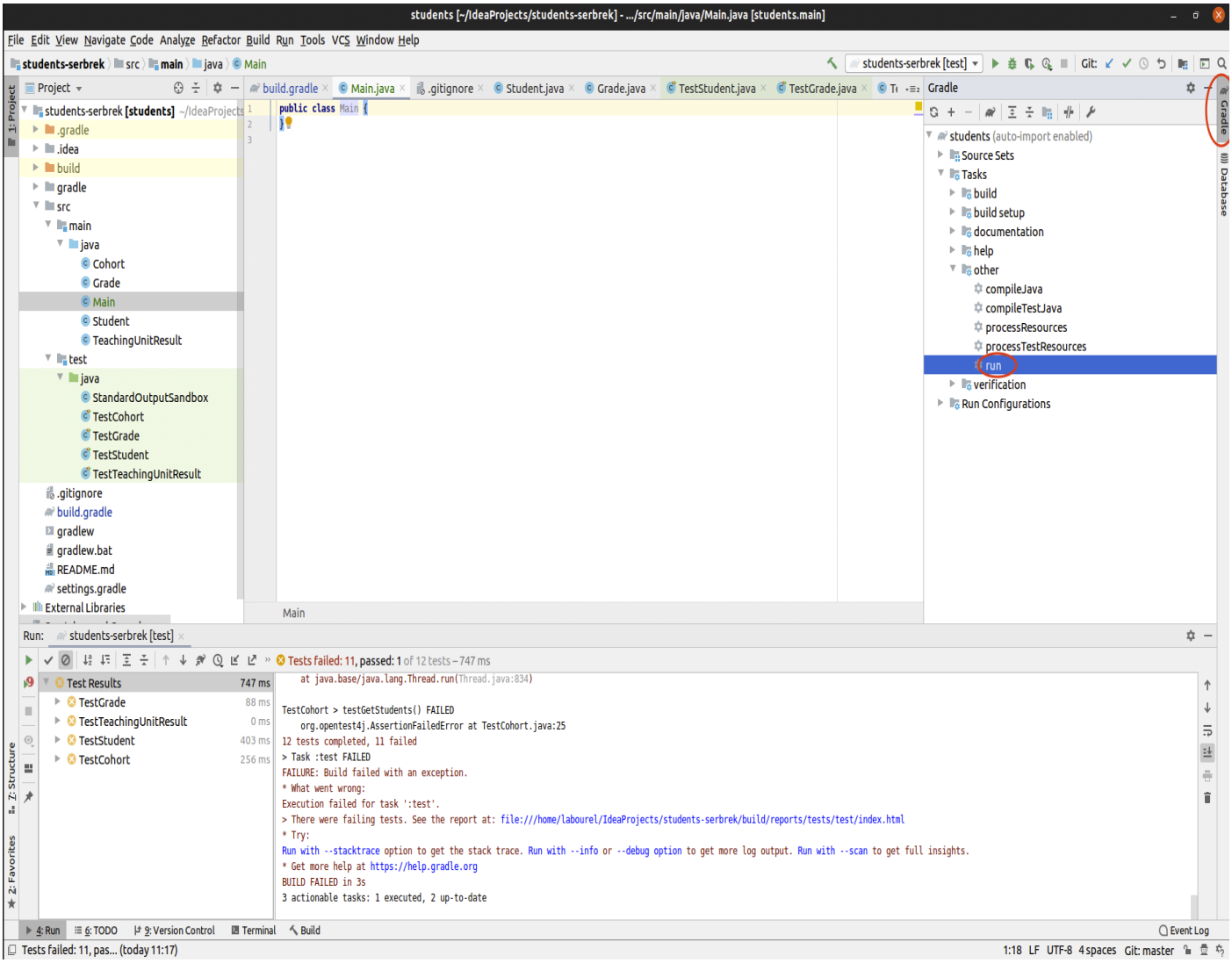
Normalement, on va vous demandez si vous souhaitez ajouter ce nouveau fichier au dépôt git. Cliquez sur `add` pour l'ajouter afin qu'il soit mis à jour lors de votre prochain *commit*.



Ajoutez dans votre classe `Main` le code d'une méthode `public static void main(String[] args)` qui :

1. crée des instances de `Student` ayant les noms et prénoms des membres du projets,
2. ajoute à ces étudiants les notes en "Programmation 2" et "Structures discrètes" que vous aimeriez avoir,
3. crée une promotion (instance de `Cohort`) ayant nommée "L2 informatique",
4. ajoute les étudiants créés à la promotion et
5. affiche les résultats de la promotion.

Pour compiler et exécuter le `main`, il faut passer par l'onglet gradle à droite et cliquer deux fois sur `students` -> `Tasks` -> `other` -> `run`. Cela va compiler et exécuter votre méthode `main`.



1.6 Tâches optionnelles

Si vous avez fini les tâches précédentes, vous pouvez améliorer votre projet en rajoutant les fonctionnalités suivantes :

- Ajout d'une méthode comptant le nombre d'étudiants ayant validé leur année (moyenne supérieure ou égale à 10) dans une promotion.
- Changement de la classe `Grade` pour qu'elle permette de stocker des notes correspondant à une absence du résultat (affiché `ABS`).
- Calcul du nombre d'absents d'une promotion.
- Calcul de la note maximum et minimum (hors absence) d'une promotion.
- Création d'une classe `TeachingUnit` qui permet d'associer des crédits à une UE.
- Calcul pondéré de la moyenne de résultats en fonction du nombre de crédits des UE.
- Calcul de la moyenne (hors absence) d'une promotion.