

---

# Java Avancé

## Cours 1 : Outillage

---

Arsène Lapostolet

10 Février 2025

---

# Les outils du module

- Plateforme de développement : Machine Virtuelle Java (JVM)
- Langage de programmation : Java 21
- Environnement de développement intégré : IntelliJ IDEA Community
- Système de Gestion de Version : Git et GitLab
- Système de build : Gradle

# Jetbrains IntelliJ IDEA

---

# Qu'est ce que IntelliJ ?

- Environnement de développement intégré pour Java et langages JVM
- Standard depuis 15 ans
- Moderne
- Version community gratuite et open source
- Version ultimate payante mais gratuite pour étudiants



La version community suffit parfaitement pour le module

# GitLab



# Qu'est ce que GitLab ?

- Serveur git
- Open source
- Intègre un module d'intégration continue (Gitlab CI)
- Revue de code
- Très utilisé dans l'industrie
- Projets (et forks) privés gratuits sur [gitlab.com](https://gitlab.com)

# Gradle

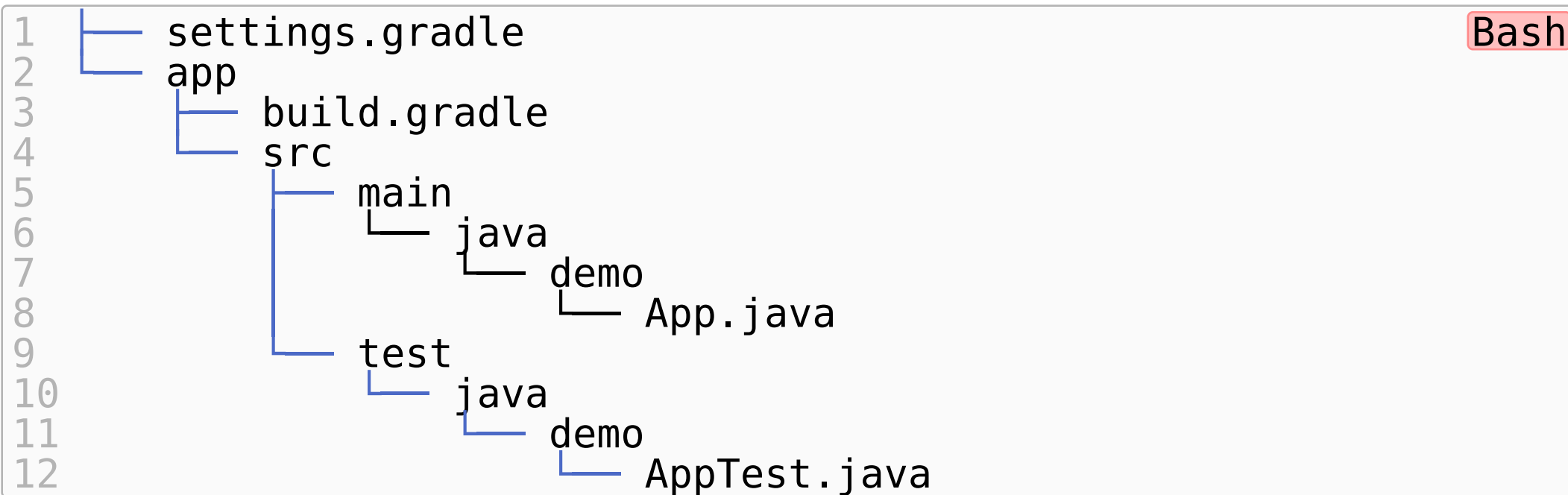
---

# Qu'est ce que Gradle ?

- Outil de build
  - Structurer le projet
  - Décrire comment le compiler, packager, exécuter, tester



# Structure d'un projet Gradle



## settings.gradle

```
1 rootProject.name = 'mon-projet'  
2 include('app')
```

Groovy

## app/build.gradle

```
1  plugins {  
2      id 'application'  
3  }  
4  
5  repositories {  
6      mavenCentral()  
7  }  
8  
9  application {  
10     mainClass = 'fr.arsenelapostolet.App'  
11 }  
12
```

Groovy

```
13
14 dependencies {
15     testImplementation 'org.junit.jupiter:junit-jupiter:5.9.2'
16     testRuntimeOnly 'org.junit.platform:junit-platform-
    launcher'
17     implementation 'com.google.guava:guava:31.1-jre'
18 }
19
20 java {
21     toolchain {
22         languageVersion = JavaLanguageVersion.of(21)
23     }
24 }
25
```

```
26
27 tasks.named('test') {
28     useJUnitPlatform()
29 }
```

# Commandes de base

- Pour exécuter l'application : `./gradlew run`
- Pour exécuter les tests : `./gradlew test`

Sur windows : `./gradlew` est remplacé par `./gradlew.bat`



Des questions ?

**PMD**

---



# Qu'est ce que PMD ?

- Analyseur statique de code
- Automatise une partie de la revue
- Avoir un code consistant

# Comment ça marche ?

→ Un fichier ruleset.xml

```

1 <?xml version="1.0"
2   xmlns="http://pmd.sourceforge.net/ruleset/2.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://pmd.sourceforge.net/ruleset/2.0.0
   https://pmd.sourceforge.io/ruleset_2_0_0.xsd">
5
6   <ruleset name="Custom Rules">
7     <description>My custom PMD rules </description>
8     <rule ref="category/java/errorprone.xml/EmptyCatchBlock" />
9 </ruleset>

```

xml

# Intégration Gradle

➔ Un fichier `ruleset.xml`

```

1 plugins {
2     ..
3     id("pmd")
4 }
5
6 pmd {
7     ruleSetFiles = files("ruleset.xml")
8     ruleSets = listOf()
9     toolVersion = "7.5.0"
10 }
```

Groovy

# Tâche Gradle

→ Un fichier `ruleset.xml`

```
1 ./gradlew check
```

bash



Des questions ?