

# Jeu de la Vie - Java : Projet de Simulation de Population

## Description du Projet

Ce projet consiste à simuler l'évolution d'une population d'humains au fil des générations. L'objectif est de créer une simulation qui modélise l'interaction, la reproduction, et la mortalité au sein d'une population. La simulation permet de tester différentes dynamiques de croissance démographique en fonction de plusieurs paramètres d'entrée.

README-pdf

## Fonctionnalités

- **Création de Population Initiale** : Génère une population initiale composée d'hommes et de femmes avec des caractéristiques spécifiques (âge, fertilité, etc.).
- **Simulation des Interactions** : Les individus peuvent se rencontrer et potentiellement se reproduire pour créer de nouveaux individus (bébés).
- **Croissance de la Population** : La population peut croître ou diminuer en fonction des règles de reproduction et de mortalité.
- **Vieillesse et Mortalité** : Chaque cycle de génération fait vieillir les individus, et certains peuvent mourir de vieillesse ou d'autres causes aléatoires.
- **Modes de Fonctionnement** : Trois modes de simulation sont proposés :
  - **Mode 0 (Normal)** : Croissance démographique basée sur des rencontres aléatoires.
  - **Mode 1 (Croissance Forcée)** : Maximum de reproduction selon le nombre d'hommes et de femmes fertiles.
  - **Mode 2 (Croissance Régulée)** : Croissance contrôlée avec un nombre spécifique de naissances par génération (nbb).

## Prérequis

Avant de commencer, assurez-vous d'avoir : - **Java Development Kit (JDK)** installé sur votre machine. Vous pouvez vérifier cela en exécutant les commandes suivantes : `bash java -version javac -version` - Tous les fichiers source nécessaires : `Main.java`, `Population.java`, `Humain.java`, `Homme.java`, `Femme.java`, `PopulationException.java`, `MeetingException.java`, etc.

## Structure des Fichiers

Votre projet doit être organisé comme suit :

```
Projet/  
  Main.java  
  Population.java  
  Homme.java  
  Femme.java  
  Humain.java  
  PopulationException.java  
  NoBreedingException.java  
  MeetingException.java
```

## Compilation et Exécution

### Étape 1 : Compilation

Pour compiler le projet, naviguez vers le répertoire contenant vos fichiers `.java` et utilisez la commande suivante :

```
javac *.java
```

Cette commande compile tous les fichiers source Java et génère les fichiers `.class` nécessaires.

### Étape 2 : Exécution

Une fois la compilation terminée, exécutez le programme avec la commande suivante :

```
java Main <nbTourDeJeu> <tailleInit> <param> [nbb]
```

## Paramètres

- `<nbTourDeJeu>` : Nombre de cycles de génération à exécuter (exemple : 10).
- `<tailleInit>` : Taille initiale de la population (exemple : 20).
- `<param>` : Mode de fonctionnement :
  - 0 : Mode normal
  - 1 : Croissance forcée
  - 2 : Croissance régulée
- `[nbb]` : Nombre de naissances par génération, requis seulement si `<param>` est 2 (doit être entre 1 et 100).

## Exemples

### 1. Exécution en mode normal (paramètre 0) :

```
java Main 10 20 0
```

Cela exécutera 10 cycles de génération avec une population initiale de 20 individus.

### 2. Exécution en mode croissance forcée (paramètre 1) :

```
java Main 15 50 1
```

Cela exécutera 15 cycles de génération avec une population initiale de 50 individus, en maximisant le nombre de rencontres fertiles.

### 3. Exécution en mode croissance régulée (paramètre 2) :

```
java Main 20 30 2 50
```

Cela exécutera 20 cycles de génération avec une population initiale de 30 individus, en fixant le nombre de naissances à 50 par cycle.

## Gestion des Exceptions

Le programme gère plusieurs types d'erreurs à travers des exceptions personnalisées : - **PopulationException** : Levée si la population devient trop petite pour se reproduire. - **MeetingException** : Levée si une erreur survient lors d'une rencontre entre individus. - **BreedingForbiddenException** et **NoBreedingException** : Levées en cas de restrictions liées à la reproduction.

En cas d'erreur, le programme affichera un message explicatif et arrêtera la simulation.

## Conclusion

Ce projet offre une simulation simple mais flexible de la dynamique d'une population humaine, permettant de tester différents scénarios de croissance et de visualiser les effets des paramètres démographiques sur plusieurs générations. N'hésitez pas à modifier et étendre le code pour explorer davantage les dynamiques de population !