

TP4 Simulation

Généré par Doxygen 1.8.13

Table des matières

1	Index hiérarchique	1
1.1	Hiérarchie des classes	1
2	Index des structures de données	3
2.1	Structures de données	3
3	Index des fichiers	5
3.1	Liste des fichiers	5
4	Documentation des structures de données	7
4.1	Référence de la classe FemaleRabbit	7
4.1.1	Description détaillée	8
4.1.2	Documentation des constructeurs et destructeur	8
4.1.2.1	FemaleRabbit()	8
4.1.2.2	~FemaleRabbit()	9
4.1.3	Documentation des fonctions membres	9
4.1.3.1	grow()	9
4.1.3.2	isMale()	10
4.2	Référence de la classe Population	10
4.2.1	Description détaillée	10
4.2.2	Documentation des constructeurs et destructeur	10
4.2.2.1	Population() [1/2]	11
4.2.2.2	Population() [2/2]	11
4.2.2.3	~Population()	11
4.2.3	Documentation des fonctions membres	11

4.2.3.1	<code>exportToCSV()</code>	11
4.2.3.2	<code>getBirth()</code>	12
4.2.3.3	<code>getDeath()</code>	13
4.2.3.4	<code>getMeanDeathAge()</code>	13
4.2.3.5	<code>getMonth()</code>	14
4.2.3.6	<code>getNumberOfFemaleRabbit()</code>	14
4.2.3.7	<code>getNumberOfMaleRabbit()</code>	15
4.2.3.8	<code>getNumberOfRabbit()</code>	15
4.2.3.9	<code>passTime()</code>	16
4.3	Référence de la classe <code>Rabbit</code>	16
4.3.1	Description détaillée	17
4.3.2	Documentation des constructeurs et destructeur	17
4.3.2.1	<code>Rabbit()</code>	18
4.3.2.2	<code>~Rabbit()</code>	18
4.3.3	Documentation des fonctions membres	18
4.3.3.1	<code>getAge()</code>	18
4.3.3.2	<code>grow()</code>	19
4.3.3.3	<code>hasMajority()</code>	19
4.3.3.4	<code>hasToDie()</code>	20
4.3.3.5	<code>isMale()</code>	20
4.3.4	Documentation des champs	20
4.3.4.1	<code>_age</code>	20
4.3.4.2	<code>_majority</code>	21
4.3.4.3	<code>_proba_to_die_adult</code>	21
4.3.4.4	<code>_proba_to_die_young</code>	21
4.3.4.5	<code>survival_proba_adult</code>	21
4.3.4.6	<code>survival_proba_young</code>	21

5	Documentation des fichiers	23
5.1	Référence du fichier FemaleRabbit.cpp	23
5.1.1	Description détaillée	23
5.2	Référence du fichier FemaleRabbit.hpp	24
5.2.1	Description détaillée	25
5.3	Référence du fichier global.hpp	25
5.3.1	Description détaillée	26
5.3.2	Documentation des variables	27
5.3.2.1	dis_int_0_11	27
5.3.2.2	dis_int_3_6	27
5.3.2.3	dis_int_5_8	27
5.3.2.4	dis_normal_6_1	27
5.3.2.5	dis_real_0_1	27
5.3.2.6	generator	27
5.4	Référence du fichier main.cpp	28
5.4.1	Description détaillée	28
5.4.2	Documentation des fonctions	29
5.4.2.1	dis_int_0_11()	29
5.4.2.2	dis_int_3_6()	29
5.4.2.3	dis_int_5_8()	29
5.4.2.4	dis_normal_6_1()	30
5.4.2.5	dis_real_0_1()	30
5.4.2.6	generator()	30
5.4.2.7	main()	30
5.5	Référence du fichier Population.cpp	31
5.5.1	Description détaillée	32
5.5.2	Documentation des fonctions	32
5.5.2.1	operator<<()	33
5.6	Référence du fichier Population.hpp	33
5.6.1	Description détaillée	35
5.6.2	Documentation des fonctions	35
5.6.2.1	operator<<()	35
5.7	Référence du fichier Rabbit.cpp	36
5.7.1	Description détaillée	37
5.8	Référence du fichier Rabbit.hpp	37
5.8.1	Description détaillée	38
	Index	39

Chapitre 1

Index hiérarchique

1.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

Population	10
Rabbit	16
FemaleRabbit	7

Chapitre 2

Index des structures de données

2.1 Structures de données

Liste des structures de données avec une brève description :

FemaleRabbit	Classe représentant un lapin femelle	7
Population	Représente une population de lapins	10
Rabbit	Classe représentant un lapin (mâle ou femelle)	16

Chapitre 3

Index des fichiers

3.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

FemaleRabbit.cpp	Fichier d'implémentation de la classe FemaleRabbit	23
FemaleRabbit.hpp	Fichier de déclaration de la classe FemaleRabbit	24
global.hpp	Fichier d'include global pour que tous les fichiers ait accès aux includes et au distributions . .	25
main.cpp	Fichier de tests pour les populations de lapins	28
Population.cpp	Fichier d'implémentation de la classe Population	31
Population.hpp	Fichier de déclaration de la classe Population	33
Rabbit.cpp	Fichier d'implémentation de la classe Rabbit	36
Rabbit.hpp	Fichier de déclaration de la classe Rabbit	37

Chapitre 4

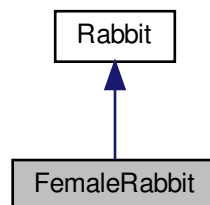
Documentation des structures de données

4.1 Référence de la classe FemaleRabbit

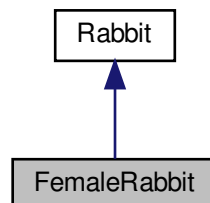
Classe représentant un lapin femelle.

```
#include <FemaleRabbit.hpp>
```

Graphe d'héritage de FemaleRabbit :



Graphe de collaboration de FemaleRabbit :



Fonctions membres publiques

- `FemaleRabbit` (std : :list< `Rabbit` *> &pop)
Construit un nouvel objet "Female Rabbit".
- `~FemaleRabbit` ()
Détruit l'objet "Female Rabbit".
- virtual void `grow` ()
Méthode faisant vieillir le lapin de 1 mois.
- virtual bool `isMale` () const
Méthode permettant de savoir si l'instance est celle d'un lapin mâle ou femelle.

Membres hérités additionnels

4.1.1 Description détaillée

Classe représentant un lapin femelle.

Définition à la ligne 22 du fichier FemaleRabbit.hpp.

4.1.2 Documentation des constructeurs et destructeur

4.1.2.1 FemaleRabbit()

```
FemaleRabbit::FemaleRabbit (
    std::list< Rabbit *> & pop )
```

Construit un nouvel objet "Female Rabbit".

Le constructeur de `FemaleRabbit` appelle le constructeur de `Rabbit` afin d'initialiser les variables (telles que l'âge et les probabilités de mourir)

Paramètres

in	<code>pop</code>	liste représentant la population de lapins où la femelle créée devra enfanter
----	------------------	---

Définition à la ligne 20 du fichier FemaleRabbit.cpp.

Voici le graphe des appelants de cette fonction :



4.1.2.2 ~FemaleRabbit()

```
FemaleRabbit::~~FemaleRabbit ( )
```

Détruit l'objet "Female Rabbit".

Définition à la ligne 31 du fichier FemaleRabbit.cpp.

4.1.3 Documentation des fonctions membres

4.1.3.1 grow()

```
void FemaleRabbit::grow ( ) [virtual]
```

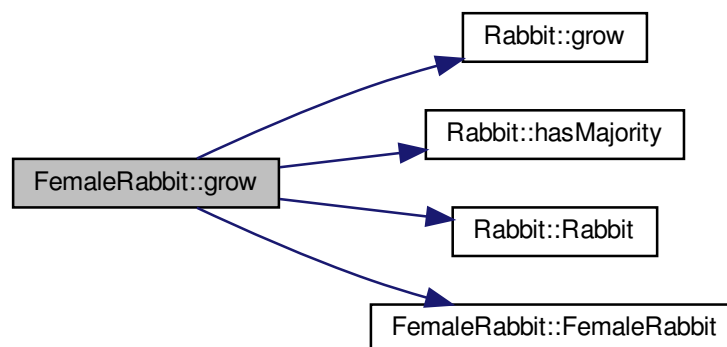
Méthode faisant vieillir le lapin de 1 mois.

Appelle la méthode grow de [Rabbit](#) afin de grandir en tant que lapin Si c'est l'anniversaire de la lapine, les mois auxquels elle enfante dans l'année sont choisis Si c'est un mois où elle doit enfanter, le nombre de bébés lapins dans la portée est décidé et elle rajoute ce nombre de nouveaux lapins à la population

Réimplémentée à partir de [Rabbit](#).

Définition à la ligne 42 du fichier FemaleRabbit.cpp.

Voici le graphe d'appel pour cette fonction :



4.1.3.2 isMale()

```
bool FemaleRabbit::isMale ( ) const [virtual]
```

Méthode permettant de savoir si l'instance est celle d'un lapin mâle ou femelle.

Réimplémentée à partir de [Rabbit](#).

Définition à la ligne 84 du fichier FemaleRabbit.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :

- [FemaleRabbit.hpp](#)
- [FemaleRabbit.cpp](#)

4.2 Référence de la classe Population

Représente une population de lapins.

```
#include <Population.hpp>
```

Fonctions membres publiques

- [Population](#) ()
Construit un nouvel objet "Population".
- [Population](#) (unsigned int numberOfMale, unsigned int numberOfFemale)
Construit un nouvel objet "Population".
- [~Population](#) ()
Détruit l'objet "Population".
- void [passTime](#) (unsigned int nbOfMonths)
passse le temps pour la population de lapins
- unsigned int [getMonth](#) () const
getter pour le nombre de mois passés par la population
- size_t [getNumberOfRabbit](#) () const
getter pour le nombre de lapins de la population à l'instant t
- size_t [getNumberOfFemaleRabbit](#) () const
getter pour le nombre de femelles dans la poplation à l'instant t
- size_t [getNumberOfMaleRabbit](#) () const
getter pour le nombre de mâles dans la poplation à l'instant t
- size_t [getDeath](#) () const
getter pour le nombre de morts depuis le début de la population
- size_t [getBirth](#) () const
getter pour le nombre de naissances depuis le début de la population
- double [getMeanDeathAge](#) () const
getter pour la moyenne d'age auquel les lapins meurent
- void [exportToCSV](#) (std::string const &filename)
ecrit dans un fichier les statistiques de la population à l'instant t

4.2.1 Description détaillée

Représente une population de lapins.

Définition à la ligne 24 du fichier Population.hpp.

4.2.2 Documentation des constructeurs et destructeur

4.2.2.1 Population() [1/2]

```
Population::Population ( )
```

Construit un nouvel objet "Population".

Initialise la population avec 1 lapin mâle et 1 lapin femelle

Définition à la ligne 18 du fichier Population.cpp.

4.2.2.2 Population() [2/2]

```
Population::Population (
    unsigned int numberOfMale,
    unsigned int numberOfFemale )
```

Construit un nouvel objet "Population".

Initialise la population avec un nombre spécifié de mâles et de femelles

Paramètres

in	<i>numberOfMale</i>	nombre de mâles à la date 0 dans la population
in	<i>numberOfFemale</i>	nombre de femelles à la date 0 dans la population

Définition à la ligne 36 du fichier Population.cpp.

4.2.2.3 ~Population()

```
Population::~~Population ( )
```

Détruit l'objet "Population".

Désalloue la mémoire allouée par les lapins de la population

Définition à la ligne 54 du fichier Population.cpp.

4.2.3 Documentation des fonctions membres

4.2.3.1 exportToCSV()

```
void Population::exportToCSV (
    std::string const & filename )
```

écrit dans un fichier les statistiques de la population à l'instant t

Paramètres

in	<i>filename</i>	nom du fichier dans lequel on ajoute une ligne de statistiques
----	-----------------	--

Définition à la ligne 187 du fichier Population.cpp.

Voici le graphe d'appel pour cette fonction :

**4.2.3.2 getBirth()**

```
size_t Population::getBirth ( ) const
```

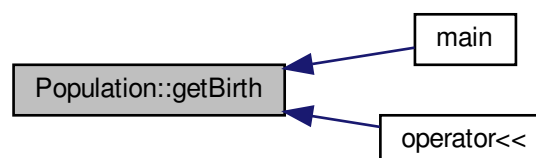
getter pour le nombre de naissances depuis le début de la population

Renvoie

size_t nombre de naissances

Définition à la ligne 167 du fichier Population.cpp.

Voici le graphe des appelants de cette fonction :



4.2.3.3 getDeath()

```
size_t Population::getDeath ( ) const
```

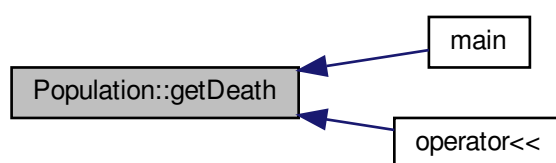
getter pour le nombre de morts depuis le début de la population

Renvoie

size_t nombre de morts

Définition à la ligne 157 du fichier Population.cpp.

Voici le graphe des appelants de cette fonction :



4.2.3.4 getMeanDeathAge()

```
double Population::getMeanDeathAge ( ) const
```

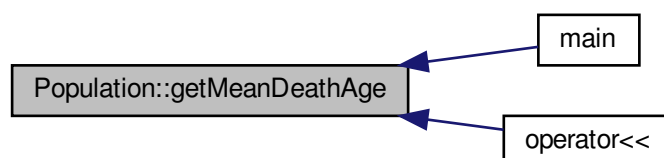
getter pour la moyenne d'age auquel les lapins meurent

Renvoie

double moyenne d'age de mort

Définition à la ligne 177 du fichier Population.cpp.

Voici le graphe des appelants de cette fonction :



4.2.3.5 getMonth()

```
unsigned int Population::getMonth ( ) const
```

getter pour le nombre de mois passés par la population

Renvoie

unsigned int nombre de mois passés par la population

Définition à la ligne 105 du fichier Population.cpp.

4.2.3.6 getNumberOfFemaleRabbit()

```
size_t Population::getNumberOfFemaleRabbit ( ) const
```

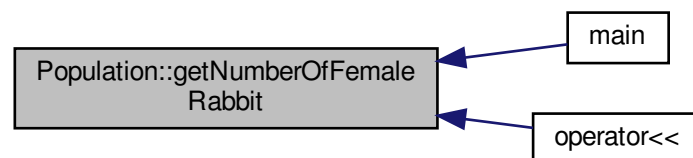
getter pour le nombre de femelles dans la poplation à l'instant t

Renvoie

size_t nombre de lapins femelles dans la population

Définition à la ligne 125 du fichier Population.cpp.

Voici le graphe des appelants de cette fonction :



4.2.3.7 getNumberOfMaleRabbit()

```
size_t Population::getNumberOfMaleRabbit ( ) const
```

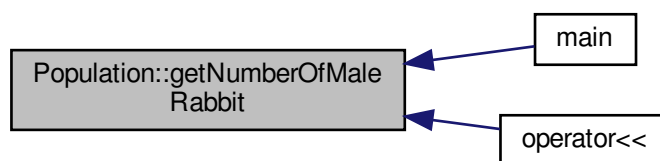
getter pour le nombre de mâles dans la poplation à l'instant t

Renvoie

size_t nombre de lapins mâles dans la population

Définition à la ligne 141 du fichier Population.cpp.

Voici le graphe des appelants de cette fonction :



4.2.3.8 getNumberOfRabbit()

```
size_t Population::getNumberOfRabbit ( ) const
```

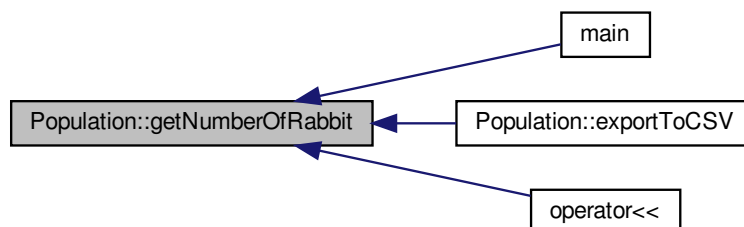
getter pour le nombre de lapins de la population à l'instant t

Renvoie

size_t nombre de lapins (vivants) dans la population

Définition à la ligne 115 du fichier Population.cpp.

Voici le graphe des appelants de cette fonction :



4.2.3.9 passTime()

```
void Population::passTime (
    unsigned int nbOfMonths )
```

passe le temps pour la population de lapins

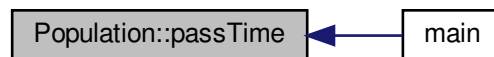
Passe un nombre de mois spécifié pour tous les lapins de la population La méthode les fait grandir avec la fonction grow, et si des lapins meurent, les supprime de la liste

Paramètres

in	<i>nbOfMonths</i>	nombre de mois que l'on veut passer
----	-------------------	-------------------------------------

Définition à la ligne 67 du fichier Population.cpp.

Voici le graphe des appelants de cette fonction :



La documentation de cette classe a été générée à partir des fichiers suivants :

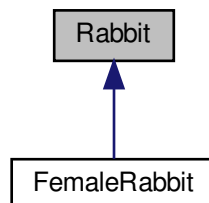
- [Population.hpp](#)
- [Population.cpp](#)

4.3 Référence de la classe Rabbit

Classe représentant un lapin (mâle ou femelle)

```
#include <Rabbit.hpp>
```

Graphe d'héritage de Rabbit :



Fonctions membres publiques

- `Rabbit ()`
Construit un nouvel objet "Rabbit".
- `virtual ~Rabbit ()`
Détruit l'objet "Rabbit".
- `virtual void grow ()`
Méthode qui fait vieillir l'instance d'un lapin de 1 mois.
- `bool hasMajority () const`
Méthode pour savoir si le lapin est jeune ou adulte.
- `bool hasToDie () const`
Méthode pour savoir si le lapin doit mourir ce mois.
- `unsigned int getAge () const`
Getter pour l'age.
- `virtual bool isMale () const`
Méthode permettant de savoir si l'instance est celle d'un lapin mâle ou femelle.

Attributs publics statiques

- `static constexpr double survival_proba_young = 0.2`
probabilité de survivre à la jeunesse (les 5 à 8 premiers mois)
- `static constexpr double survival_proba_adult = 0.5`
probabilité de survivre pour un lapin adulte (11 ans moins les mois de jeunesse)

Attributs protégés

- `unsigned int _age`
L'age du lapin en mois.
- `unsigned int _majority`
Nombre de mois avant que le lapin n'atteigne la majorité et passe à l'age adulte et puisse enfanter (si c'est une femelle)
- `double _proba_to_die_young`
Probabilité de mourir (par mois) avant de devenir adulte, calculée avec la probabilité de survivre et le nombre de mois de jeunesse.
- `double _proba_to_die_adult`
Probabilité de mourir (par mois) en étant adulte, calculée avec la probabilité de survivre et le nombre de mois de jeunesse.

4.3.1 Description détaillée

Classe représentant un lapin (mâle ou femelle)

Si l'objet est créé en tant que `Rabbit`, c'est qu'il représente un lapin mâle, sinon la classe utilisée sera `Female` ← `Rabbit`. Cependant la classe `Rabbit` synthétise les deux genres. Il n'y a juste pas de classe `MaleRabbit` puisque celle-ci n'implémenterait rien de plus que la classe `Rabbit`.

Définition à la ligne 23 du fichier `Rabbit.hpp`.

4.3.2 Documentation des constructeurs et destructeur

4.3.2.1 Rabbit()

```
Rabbit::Rabbit ( )
```

Construit un nouvel objet "Rabbit".

Définit l'age de la majorité, la probabilité de mourir jeune et la probabilité de mourir en étant adulte

Définition à la ligne 18 du fichier Rabbit.cpp.

Voici le graphe des appelants de cette fonction :



4.3.2.2 ~Rabbit()

```
Rabbit::~~Rabbit ( ) [virtual]
```

Détruit l'objet "Rabbit".

Définition à la ligne 29 du fichier Rabbit.cpp.

4.3.3 Documentation des fonctions membres

4.3.3.1 getAge()

```
unsigned int Rabbit::getAge ( ) const
```

Getter pour l'age.

Renvoie

unsigned int l'age du lapin

Définition à la ligne 68 du fichier Rabbit.cpp.

4.3.3.2 grow()

```
void Rabbit::grow ( ) [virtual]
```

Méthode qui fait vieillir l'instance d'un lapin de 1 mois.

Réimplémentée dans [FemaleRabbit](#).

Définition à la ligne 36 du fichier Rabbit.cpp.

Voici le graphe des appelants de cette fonction :



4.3.3.3 hasMajority()

```
bool Rabbit::hasMajority ( ) const
```

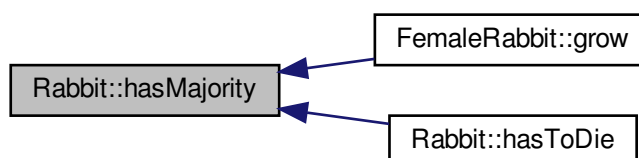
Méthode pour savoir si le lapin est jeune ou adulte.

Renvoie

true si le lapin est adulte
false si le lapin est jeune

Définition à la ligne 47 du fichier Rabbit.cpp.

Voici le graphe des appelants de cette fonction :



4.3.3.4 hasToDie()

```
bool Rabbit::hasToDie ( ) const
```

Méthode pour savoir si le lapin doit mourir ce mois.

Renvoie

true si il doit mourir
false si il survit

Définition à la ligne 58 du fichier Rabbit.cpp.

Voici le graphe d'appel pour cette fonction :



4.3.3.5 isMale()

```
bool Rabbit::isMale ( ) const [virtual]
```

Méthode permettant de savoir si l'instance est celle d'un lapin mâle ou femelle.

Réimplémentée dans [FemaleRabbit](#).

Définition à la ligne 76 du fichier Rabbit.cpp.

4.3.4 Documentation des champs

4.3.4.1 _age

```
unsigned int Rabbit::_age [protected]
```

L'age du lapin en mois.

Définition à la ligne 39 du fichier Rabbit.hpp.

4.3.4.2 `_majority`

```
unsigned int Rabbit::_majority [protected]
```

Nombre de mois avant que le lapin n'atteigne la majorité et passe à l'âge adulte et puisse enfanter (si c'est une femelle)

Définition à la ligne 40 du fichier Rabbit.hpp.

4.3.4.3 `_proba_to_die_adult`

```
double Rabbit::_proba_to_die_adult [protected]
```

Probabilité de mourir (par mois) en étant adulte, calculée avec la probabilité de survivre et le nombre de mois de jeunesse.

Définition à la ligne 42 du fichier Rabbit.hpp.

4.3.4.4 `_proba_to_die_young`

```
double Rabbit::_proba_to_die_young [protected]
```

Probabilité de mourir (par mois) avant de devenir adulte, calculée avec la probabilité de survivre et le nombre de mois de jeunesse.

Définition à la ligne 41 du fichier Rabbit.hpp.

4.3.4.5 `survival_proba_adult`

```
constexpr double Rabbit::survival_proba_adult = 0.5 [static]
```

probabilité de survivre pour un lapin adulte (11 ans moins les mois de jeunesse)

Définition à la ligne 36 du fichier Rabbit.hpp.

4.3.4.6 `survival_proba_young`

```
constexpr double Rabbit::survival_proba_young = 0.2 [static]
```

probabilité de survivre à la jeunesse (les 5 à 8 premiers mois)

Définition à la ligne 35 du fichier Rabbit.hpp.

La documentation de cette classe a été générée à partir des fichiers suivants :

- [Rabbit.hpp](#)
- [Rabbit.cpp](#)

Chapitre 5

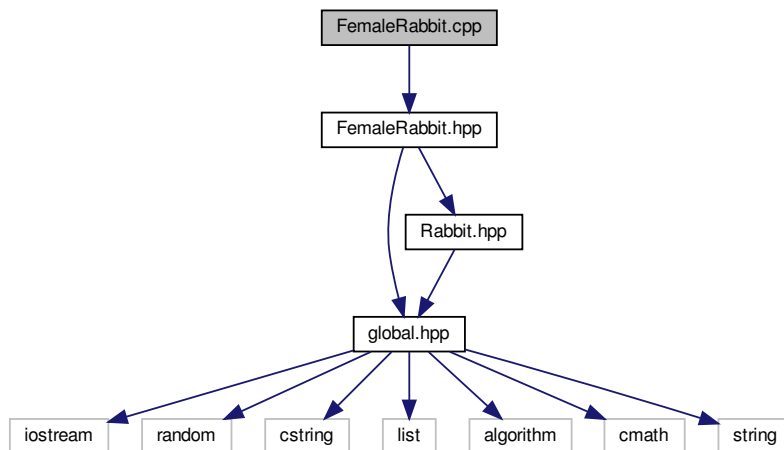
Documentation des fichiers

5.1 Référence du fichier FemaleRabbit.cpp

Fichier d'implémentation de la classe [FemaleRabbit](#).

```
#include "FemaleRabbit.hpp"
```

Graphe des dépendances par inclusion de FemaleRabbit.cpp :



5.1.1 Description détaillée

Fichier d'implémentation de la classe [FemaleRabbit](#).

Auteur

Mathieu Arquilliere (mathieu.arquilliere@etu.uca.fr)

Version

0.1

Date

2019-11-17

Copyright

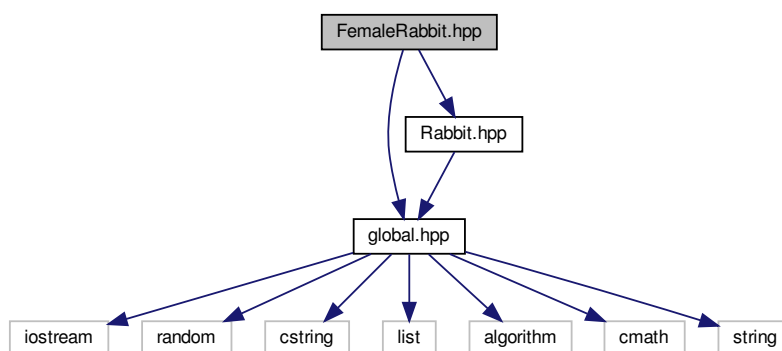
Copyright (c) 2019

5.2 Référence du fichier FemaleRabbit.hpp

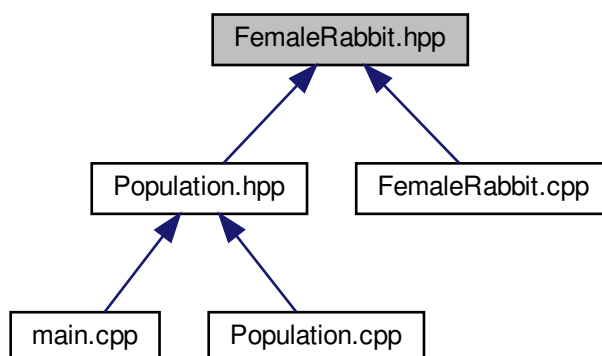
Fichier de déclaration de la classe [FemaleRabbit](#).

```
#include "global.hpp"  
#include "Rabbit.hpp"
```

Graphe des dépendances par inclusion de FemaleRabbit.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Structures de données

- class [FemaleRabbit](#)
Classe représentant un lapin femelle.

5.2.1 Description détaillée

Fichier de déclaration de la classe [FemaleRabbit](#).

Auteur

Mathieu Arquilliere (mathieu.arquilliere@etu.uca.fr)

Version

0.1

Date

2019-11-17

Copyright

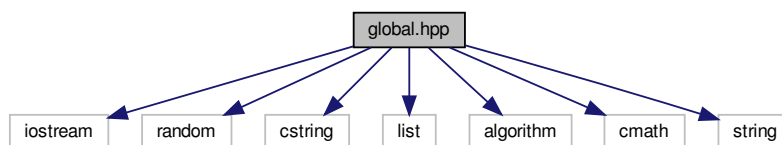
Copyright (c) 2019

5.3 Référence du fichier global.hpp

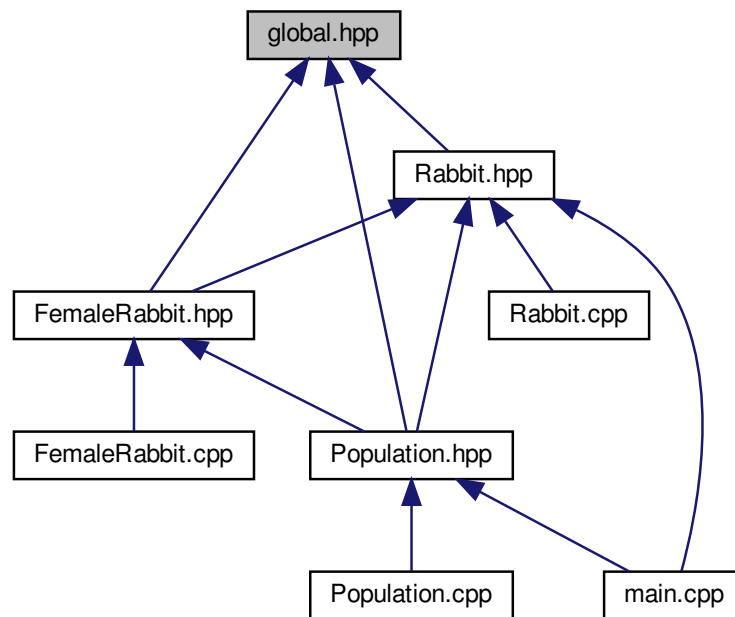
Fichier d'include global pour que tous les fichiers ait accès aux includes et au distributions.

```
#include <iostream>
#include <random>
#include <cstring>
#include <list>
#include <algorithm>
#include <cmath>
#include <string>
```

Grappe des dépendances par inclusion de global.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Variables

- std : :mt19937 [generator](#)
- std : :uniform_real_distribution [dis_real_0_1](#)
- std : :uniform_int_distribution [dis_int_3_6](#)
- std : :uniform_int_distribution [dis_int_0_11](#)
- std : :uniform_int_distribution [dis_int_5_8](#)
- std : :normal_distribution [dis_normal_6_1](#)

5.3.1 Description détaillée

Fichier d'inclure global pour que tous les fichiers ait accès aux includes et au distributions.

Auteur

Mathieu Arquilliere (mathieu.arquilliere@etu.uca.fr)

Version

0.1

Date

2019-11-18

Copyright

Copyright (c) 2019

5.3.2 Documentation des variables

5.3.2.1 dis_int_0_11

```
std::uniform_int_distribution dis_int_0_11
```

5.3.2.2 dis_int_3_6

```
std::uniform_int_distribution dis_int_3_6
```

5.3.2.3 dis_int_5_8

```
std::uniform_int_distribution dis_int_5_8
```

5.3.2.4 dis_normal_6_1

```
std::normal_distribution dis_normal_6_1
```

5.3.2.5 dis_real_0_1

```
std::uniform_real_distribution dis_real_0_1
```

5.3.2.6 generator

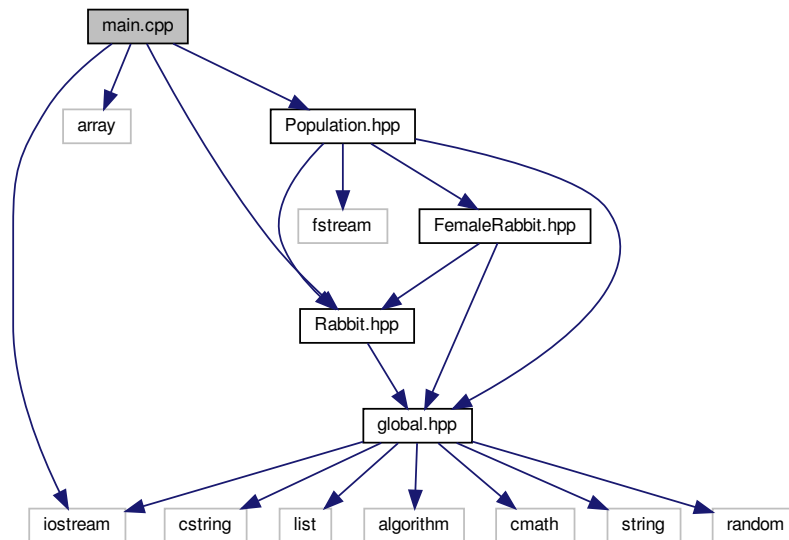
```
std::mt19937 generator
```

5.4 Référence du fichier main.cpp

Fichier de tests pour les populations de lapins.

```
#include <iostream>
#include <array>
#include "Rabbit.hpp"
#include "Population.hpp"
```

Graphe des dépendances par inclusion de main.cpp :



Fonctions

- std : :mt19937 [generator](#) (456)
générateur de nombre aléatoire utilisant Mersenne Twister
- std : :uniform_real_distribution [dis_real_0_1](#) (0.0, 1.0)
distribution réelle uniforme entre 0 et 1
- std : :uniform_int_distribution [dis_int_3_6](#) (3, 6)
distribution entière uniforme entre 3 et 6 (nombre de bébés dans chaque portée)
- std : :uniform_int_distribution [dis_int_0_11](#) (0, 11)
distribution entière uniforme entre 0 et 11 (mois auquel une femelle enfante)
- std : :uniform_int_distribution [dis_int_5_8](#) (5, 8)
distribution entière uniforme entre 5 et 8 (nombre de mois avant la majorité d'un lapin)
- std : :normal_distribution [dis_normal_6_1](#) (6, 1)
distribution normale (nombre de portées d'une femelle dans l'année)
- int [main](#) (int argc, char *argv[])

5.4.1 Description détaillée

Fichier de tests pour les populations de lapins.

Auteur

Mathieu Arquilliere (mathieu.arquilliere@etu.uca.fr)

Version

0.1

Date

2019-11-17

Copyright

Copyright (c) 2019

5.4.2 Documentation des fonctions**5.4.2.1 dis_int_0_11()**

```
std::uniform_int_distribution dis_int_0_11 (  
    0 ,  
    11 )
```

distribution entière uniforme entre 0 et 11 (mois auquel une femelle enfante)

5.4.2.2 dis_int_3_6()

```
std::uniform_int_distribution dis_int_3_6 (  
    3 ,  
    6 )
```

distribution entière uniforme entre 3 et 6 (nombre de bébés dans chaque portée)

5.4.2.3 dis_int_5_8()

```
std::uniform_int_distribution dis_int_5_8 (  
    5 ,  
    8 )
```

distribution entière uniforme entre 5 et 8 (nombre de mois avant la majorité d'un lapin)

5.4.2.4 `dis_normal_6_1()`

```
std::normal_distribution dis_normal_6_1 (
    6 ,
    1 )
```

distribution normale (nombre de portées d'une femelle dans l'année)

5.4.2.5 `dis_real_0_1()`

```
std::uniform_real_distribution dis_real_0_1 (
    0. 0,
    1. 0 )
```

distribution réelle uniforme entre 0 et 1

5.4.2.6 `generator()`

```
std::mt19937 generator (
    456 )
```

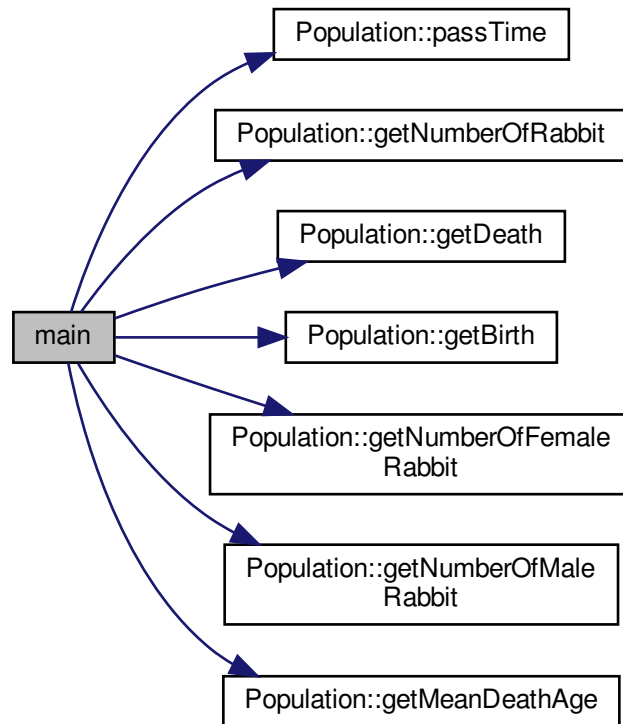
générateur de nombre aléatoire utilisant Mersenne Twister

5.4.2.7 `main()`

```
int main (
    int argc,
    char * argv[] )
```

Définition à la ligne 46 du fichier main.cpp.

Voici le graphe d'appel pour cette fonction :

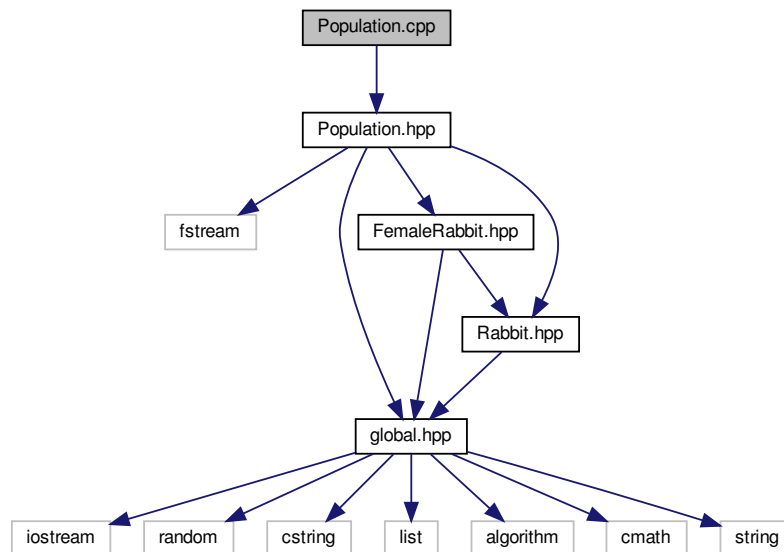


5.5 Référence du fichier Population.cpp

Fichier d'implémentation de la classe [Population](#).

```
#include "Population.hpp"
```

Graphe des dépendances par inclusion de Population.cpp :



Fonctions

— `std::ostream & operator<< (std::ostream &out, Population const &pop)`
surcharge de l'opérateur <<

5.5.1 Description détaillée

Fichier d'implémentation de la classe `Population`.

Auteur

Mathieu Arquilliere (mathieu.arquilliere@etu.uca.fr)

Version

0.1

Date

2019-11-17

Copyright

Copyright (c) 2019

5.5.2 Documentation des fonctions

5.5.2.1 operator<<()

```
std::ostream& operator<< (
    std::ostream & out,
    Population const & pop )
```

surcharge de l'opérateur <<

Écrit dans un flux donné l'état actuel de la population

Paramètres

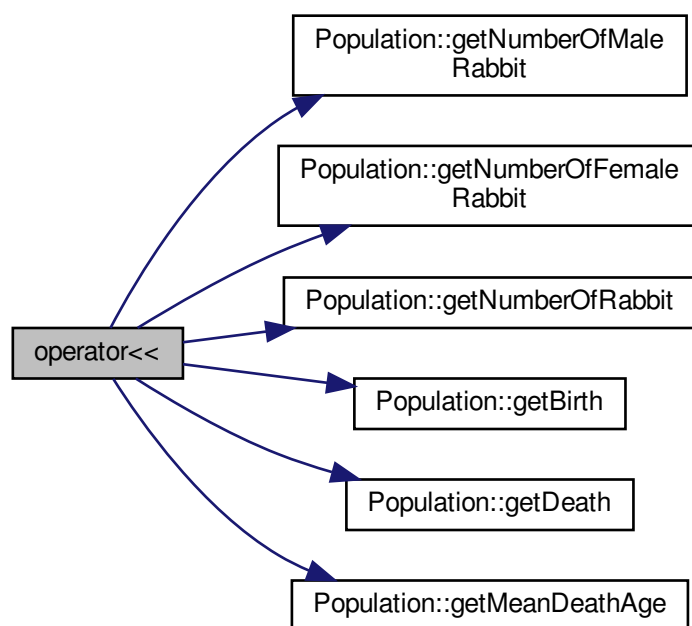
in	<i>out</i>	flux de sortie dans lequel on écrit
in	<i>pop</i>	population dont on veut écrire l'état

Renvoie

std : ostream& flux de sortie

Définition à la ligne 206 du fichier Population.cpp.

Voici le graphe d'appel pour cette fonction :

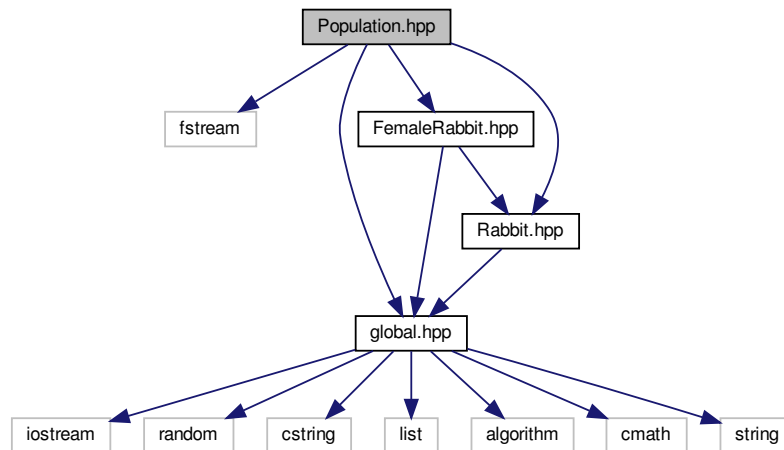


5.6 Référence du fichier Population.hpp

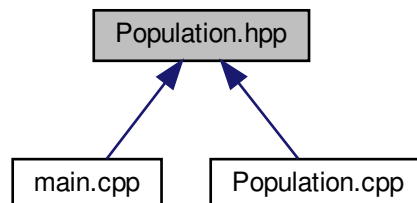
Fichier de déclaration de la classe [Population](#).

```
#include <fstream>
#include "global.hpp"
#include "FemaleRabbit.hpp"
#include "Rabbit.hpp"
```

Graphe des dépendances par inclusion de Population.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Structures de données

- class `Population`
Représente une population de lapins.

Fonctions

- `std::ostream & operator<< (std::ostream &out, Population const &pop)`
surcharge de l'opérateur <<

5.6.1 Description détaillée

Fichier de déclaration de la classe [Population](#).

Auteur

Mathieu Arquilliere (mathieu.arquilliere@etu.uca.fr)

Version

0.1

Date

2019-11-17

Copyright

Copyright (c) 2019

5.6.2 Documentation des fonctions

5.6.2.1 `operator<<()`

```
std::ostream& operator<< (
    std::ostream & out,
    Population const & pop )
```

surcharge de l'opérateur <<

Ecrit dans un flux donné l'état actuel de la population

Paramètres

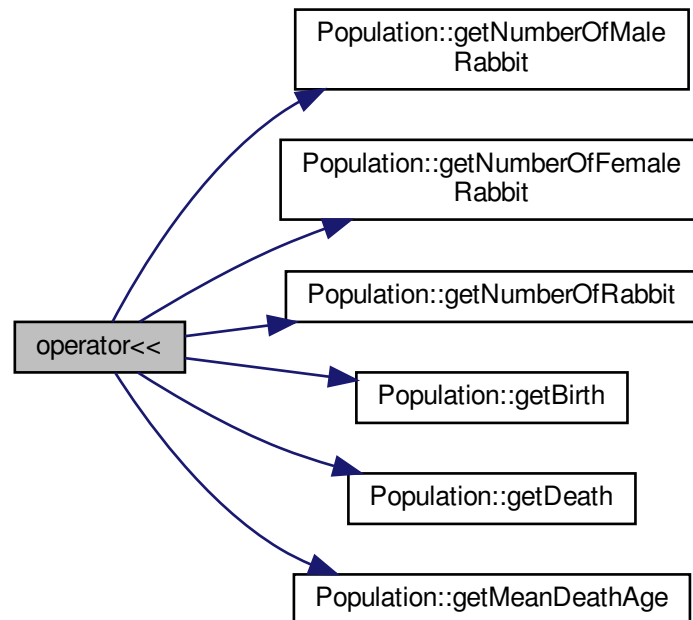
in	<i>out</i>	flux de sortie dans lequel on écrit
in	<i>pop</i>	population dont on veut écrire l'état

Renvoie

std : ostream& flux de sortie

Définition à la ligne 206 du fichier Population.cpp.

Voici le graphe d'appel pour cette fonction :

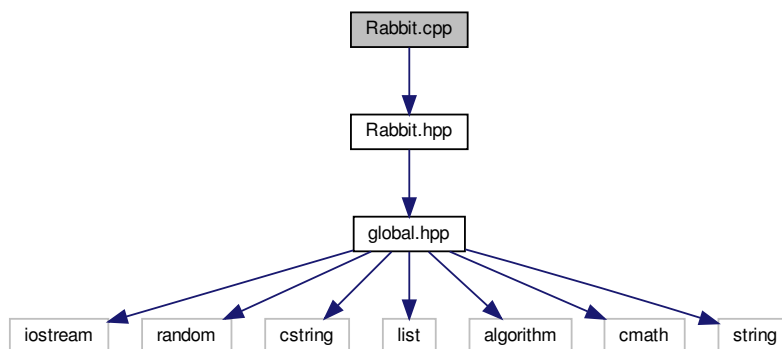


5.7 Référence du fichier Rabbit.cpp

Fichier d'implémentation de la classe [Rabbit](#).

```
#include "Rabbit.hpp"
```

Graphe des dépendances par inclusion de Rabbit.cpp :



5.7.1 Description détaillée

Fichier d'implémentation de la classe [Rabbit](#).

Auteur

Mathieu Arquilliere (mathieu.arquilliere@etu.uca.fr)

Version

0.1

Date

2019-11-17

Copyright

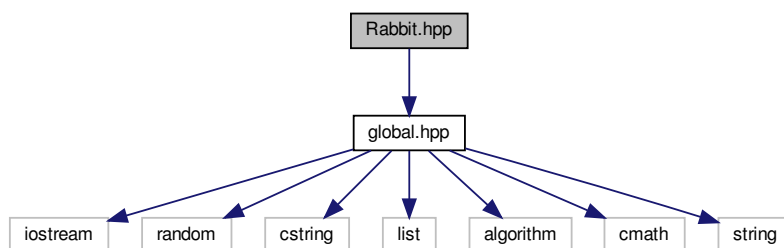
Copyright (c) 2019

5.8 Référence du fichier Rabbit.hpp

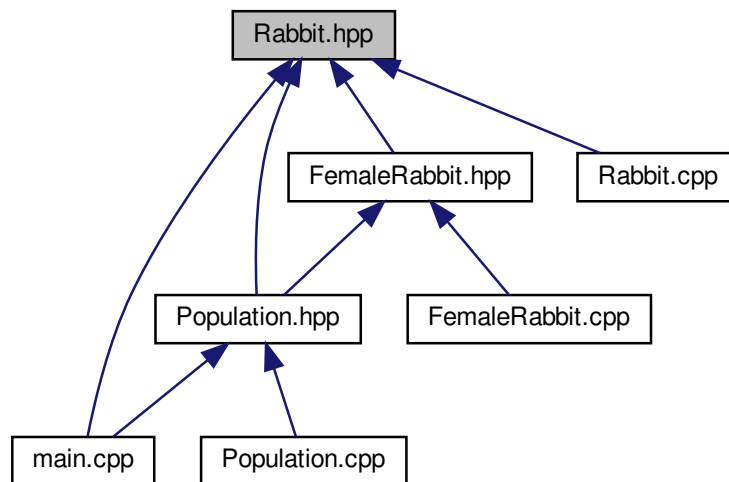
Fichier de déclaration de la classe [Rabbit](#).

```
#include "global.hpp"
```

Graphe des dépendances par inclusion de Rabbit.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Structures de données

— class `Rabbit`
Classe représentant un lapin (mâle ou femelle)

5.8.1 Description détaillée

Fichier de déclaration de la classe `Rabbit`.

Auteur

Mathieu Arquilliere (mathieu.arquilliere@etu.uca.fr)

Version

0.1

Date

2019-11-17

Copyright

Copyright (c) 2019

Index

- [_age](#)
 - [Rabbit, 20](#)
 - [_majority](#)
 - [Rabbit, 20](#)
 - [_proba_to_die_adult](#)
 - [Rabbit, 21](#)
 - [_proba_to_die_young](#)
 - [Rabbit, 21](#)
 - [~FemaleRabbit](#)
 - [FemaleRabbit, 8](#)
 - [~Population](#)
 - [Population, 11](#)
 - [~Rabbit](#)
 - [Rabbit, 18](#)
- [dis_int_0_11](#)
 - [global.hpp, 27](#)
 - [main.cpp, 29](#)
- [dis_int_3_6](#)
 - [global.hpp, 27](#)
 - [main.cpp, 29](#)
- [dis_int_5_8](#)
 - [global.hpp, 27](#)
 - [main.cpp, 29](#)
- [dis_normal_6_1](#)
 - [global.hpp, 27](#)
 - [main.cpp, 29](#)
- [dis_real_0_1](#)
 - [global.hpp, 27](#)
 - [main.cpp, 30](#)
- [exportToCSV](#)
 - [Population, 11](#)
- [FemaleRabbit, 7](#)
 - [~FemaleRabbit, 8](#)
 - [FemaleRabbit, 8](#)
 - [grow, 9](#)
 - [isMale, 9](#)
- [FemaleRabbit.cpp, 23](#)
- [FemaleRabbit.hpp, 24](#)
- [generator](#)
 - [global.hpp, 27](#)
 - [main.cpp, 30](#)
- [getAge](#)
 - [Rabbit, 18](#)
- [getBirth](#)
 - [Population, 12](#)
- [getDeath](#)
 - [Population, 12](#)
- [getMeanDeathAge](#)
 - [Population, 13](#)
- [getMonth](#)
 - [Population, 13](#)
- [getNumberOfFemaleRabbit](#)
 - [Population, 14](#)
- [getNumberOfMaleRabbit](#)
 - [Population, 14](#)
- [getNumberOfRabbit](#)
 - [Population, 15](#)
- [global.hpp, 25](#)
 - [dis_int_0_11, 27](#)
 - [dis_int_3_6, 27](#)
 - [dis_int_5_8, 27](#)
 - [dis_normal_6_1, 27](#)
 - [dis_real_0_1, 27](#)
 - [generator, 27](#)
- [grow](#)
 - [FemaleRabbit, 9](#)
 - [Rabbit, 18](#)
- [hasMajority](#)
 - [Rabbit, 19](#)
- [hasToDie](#)
 - [Rabbit, 19](#)
- [isMale](#)
 - [FemaleRabbit, 9](#)
 - [Rabbit, 20](#)
- [main](#)
 - [main.cpp, 30](#)
- [main.cpp, 28](#)
 - [dis_int_0_11, 29](#)
 - [dis_int_3_6, 29](#)
 - [dis_int_5_8, 29](#)
 - [dis_normal_6_1, 29](#)
 - [dis_real_0_1, 30](#)
 - [generator, 30](#)
 - [main, 30](#)
- [operator<<](#)
 - [Population.cpp, 32](#)
 - [Population.hpp, 35](#)
- [passTime](#)
 - [Population, 15](#)
- [Population, 10](#)
 - [~Population, 11](#)
 - [exportToCSV, 11](#)

- getBirth, [12](#)
- getDeath, [12](#)
- getMeanDeathAge, [13](#)
- getMonth, [13](#)
- getNumberOfFemaleRabbit, [14](#)
- getNumberOfMaleRabbit, [14](#)
- getNumberOfRabbit, [15](#)
- passTime, [15](#)
- Population, [10](#), [11](#)
- Population.cpp, [31](#)
 - operator<<, [32](#)
- Population.hpp, [33](#)
 - operator<<, [35](#)
- Rabbit, [16](#)
 - _age, [20](#)
 - _majority, [20](#)
 - _proba_to_die_adult, [21](#)
 - _proba_to_die_young, [21](#)
 - ~Rabbit, [18](#)
 - getAge, [18](#)
 - grow, [18](#)
 - hasMajority, [19](#)
 - hasToDie, [19](#)
 - isMale, [20](#)
 - Rabbit, [17](#)
 - survival_proba_adult, [21](#)
 - survival_proba_young, [21](#)
- Rabbit.cpp, [36](#)
- Rabbit.hpp, [37](#)
- survival_proba_adult
 - Rabbit, [21](#)
- survival_proba_young
 - Rabbit, [21](#)