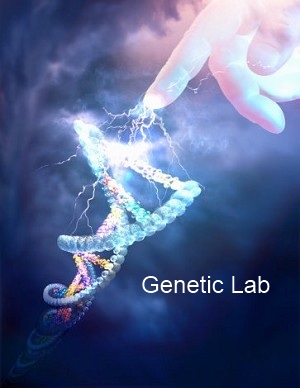
**Classes v0.2**



|  |  |
| --- | --- |
| Projecto: | **Life Inspiration** |
| Autor: | Genetic Lab |
| Data de preparação: | 2012-04-17 |
| Circulação: | Genetic Lab, Power Computing |

# Descrição sucinta das classes

**Classe PopulationUtils**

|  |
| --- |
| **PopulationUtils**  *implements Comparator<Individual>* |
| - final Population \_\_newPopulation;  - int \_\_bestFitness;  - int \_\_numberOfIndividualsWithBestFitness;  - final ArrayList<Individual> \_\_newArrayIndividual  - int \_\_countIndividual;  - int \_\_indexIndividual;  - int \_\_numberIndividualToReturn;  - final int NUMBER\_INDIVIDUALS\_TO\_GET\_FROM\_HALL\_OF\_FAME;  - final int FIRST\_INDIVIDUAL;  - int \_\_totalFitness;  - final Population \_\_newPopulation; |
| + public static Population getHallOfFame(Population population, int sizeHallOfFame);  + public static void orderPopulation(Population population);  + public static int getNumberIndividualsWithBestFitness(Population population);  + public static ArrayList<Individual> getArrayIndividualsRandom(Population population, int numberIndividual, boolean removeIndividualFromPopulation);  + public static int getBestFitness(Population population);  + public static int getFitnessTotal(Population population);  + public static Population getHallOfFameWithoutDuplicateIndividuals(Population population, int sizeHallOfFame); |

Descrição: Classe que servirá de apoio à classe Population para que da mesma possam ser retiradas informações acerca dos seus indivíduos, fitness's, hall of fame's e afins.

O método public static Population getHallOfFame(Population population, int sizeHallOfFame)devolve uma população que só irá conter os melhores indivíduos da população recebida por parâmetro.

O método public static void orderPopulation(Population population) ordena a população, ou seja, os seus individuos pelo seu fitness, do melhor para o pior.

O método public static int getNumberIndividualsWithBestFitness(Population population) devolve o número de individúos que tem o fitness máximo na população.

O método public static ArrayList<Individual> getArrayIndividualsRandom(Population population, int numberIndividual, boolean removeIndividualFromPopulation) devolve um ArrayList de individuos da população, sendo que esse individuos são uma amosta ao acaso da população em causa.

O método public static int getBestFitness(Population population) devolve o valor de fitness mais alto da população.

O método public static int getFitnessTotal(Population population) devolve o fitness total da população (soma do fitness de todos os individuos da população).

O método public static Population getHallOfFameWithoutDuplicateIndividuals(Population population, int sizeHallOfFame) devolve uma população que não irá conter individuos duplicados.

**Classe Ponteiro**

|  |
| --- |
| **Ponteiro** |
| - int totalFitness;  - double ponteiro; |
| + public static double pontoAleatorio(int totalFitnessPopulacao);  + public static Individual devolveIndividuoParaOndeOPonteiroAponta(double ponteiro, Population population) throws PonteiroForaDoLimiteException; |

Descrição: Classe Ponteiro que será um ponteiro para a população. Este ponteiro será calculado aleatoriamente e apontará para um individuo da população passada por parâmetro.

O método public static double pontoAleatorio(int totalFitnessPopulacao)devolve um ponto aleatório entre 0 e o total de fitness da população.

O método public static Individual devolveIndividuoParaOndeOPonteiroAponta(double ponteiro, Population population) throws PonteiroForaDoLimiteException devolve o individuo para onde o ponteiro aponta.