UML Class Diagram

PopulationManager
- ancestryForest: List<Individual>
- currentPopulation: List<Individual>

+ PopulationManager(List<Individual>) (all of the roots)
+ getCurrentPopulation(): List<Individual>
+ kill(Individual i): void
+ giveBirth(int numKids, List<Individual> parents): List<Individual>
+ getAncestryForest(): List<Individual> *(all of the roots)*
+ setStartPopulation(List<Individual>): void
+ createMemento(): PopulationManagerMemento
+ setMemento(PopulationManagerMemento m): void

Individual
- genotype: Genotype
- phenotype: Phenotype
- parents: List<Individual>
- children: List<Individual>
- lastGeneticOperation: String

+ Individual(Genotype g, Phenotype p)
+ addParent(Individual): void
+ addChild(Individual): void
+ getParents(): List<Individual>
+ getChildren(): List<Individual>
+ transGeno(): void
+ getGenotype(): Genotype
+ getPhenotype(): Phenotype
+ setLastGenOp(GeneticOperation go): void
+ getLastGenOp(): String
+ cloneGeno(): Genotype

Serializeable

`0..*`

{Abstract} IndividualAbstractFactory

---
newPhenotype(): Phenotype
newGenotypeList(File f): List<Genotype>
newRandomGenotypeList(): List<Genotype>

RunManager
popManager: PopulationManager
mutator: Mutator
judge: Judge
loadSave: LoadSave
af: IndividualAbstractFactory
maxGens: int
hooks: List<RunCommand>

+ RunManager(PopulationManager pm,
Mutator m, Judge j, LoadSave ls,
IndividualAbstractFactory af,
List<RunCommand> hooks, int maxGens)
+ earlyTerminatingCondition(): boolean
(hook TM)
+ run(): void
+ save(): void
+ stop(): void

<<Interface>> Genotype

+ clone(): Genotype

<<Interface>> Phenotype

+ genoToPheno(Genotype g): void

Mutator
geneticOperations: List<GeneticOperation>

+ Mutator(List<GeneticOperation> gOps)
+ addGenOp(GeneticOperation go): void
+ removeGenOp(GeneticOperation go): void
+ mutate(List<Individual> old,
PopulationManager pm): void

StringGenotype
+ genotype: String

+ clone(): Genotype

{Abstract} GeneticOperation

+ execute(Individual child, List<Individual>
parents): void
+ getParentsRequired(): int
+ getName(): String
+ getWeight(): double

`0..*`

LoadSave
- archivePath:String

+ LoadSave(String archivePath)
+ load(int gensBack): PopulationManagerMemento m
+ load(File oldPop): PopulationManagerMemento m
+ save(PopulationManagerMemento m) : void

Judge
- killPercent: double
- mutatePercent: double
fa: FitnessAlgorithm

+ Judge(double d1, double d2,
FitnessAlgorithm fa)
+ judge(PopulationManager
pm): List<Individual>

RunFacade
- runManager: RunManager
- popManager: PopulationManager
- mutator: Mutator
- judge: Judge
- loadSave: LoadSave
- af: IndividualAbstractFactory
- hooks: List<RunCommand>

+ RunFacade(IndividualAbstractFactory af)
+ run(): void
+ save(): void
+ stop(): void
+ configure(File config): void
+ configure(int maxGens, double killPercent, double mutatePercent, String archivePath): void
+ load(int gensBack): void
+ loadInitialPop(File popFile): void
+ loadRandomPop(): void
+ addHook(RunCommand): void
+ removeHook(RunCommand): void
+ setFitnessAlgorithm(FitnessAlgorithm): void
+ getCurrentPopulationGenotype(): List<Genotype>
+ getCurrentPopulationPhenotype(): List<Phenotype>
+ addGenOp(GeneticOperation go): void
+ removeGenOp(GeneticOperation go): void

```
┌─────────────────────────────────┐        ┌─────────────────────────────────────────┐
│              Judge              │        │            <<Interface>>                │
├─────────────────────────────────┤        │           FitnessAlgorithm              │
│ - killPercent: double           │        ├─────────────────────────────────────────┤
│ - mutatePercent: double         │◆───────┤                                         │
│ - fitnessAlgorithm: FitnessAlgorithm │     │                                         │
├─────────────────────────────────┤        ├─────────────────────────────────────────┤
│ + judge(PopulationManager pm): void │     │ + getFitness(Phenotype): Comparable<?>  │
└─────────────────────────────────┘        └─────────────────────────────────────────┘
                                                              △
                                                              ┊
                                             ┌─────────────────────────────────────────┐
                                             │        ConcreteFitnessAlgorithm          │
                                             ├─────────────────────────────────────────┤
                                             │                                         │
                                             │                                         │
                                             ├─────────────────────────────────────────┤
                                             │ + getFitness(Phenotype) : Comparable<?>  │
                                             └─────────────────────────────────────────┘
```

Mutator

geneticOperations: List<GeneticOperation>

+ Mutator(List<GeneticOperation> gOps)
+ addGenOp(GeneticOperation go): void
+ removeGenOp(GeneticOperation go): void
+ mutate(List<Individual> old,
PopulationManager pm): void

0..*

<<Interface>>
GeneticOperation

+ execute(Individual child, List<Individual>
parents): void
+ getParentsRequired(): int
+ getName(): String
+ getWeight(): double

Other

+ execute(Individual child, List<Individual>
parents): void
+ getParentsRequired(): int
+ getName(): String
+ getWeight(): double

StringCrossover

+ execute(Individual child, List<Individual> parents):
void
+ getParentsRequired(): int
+ getName(): String
+ getWeight(): double

StringMutation

+ execute(Individual child, List<Individual>
parents): void
+ getParentsRequired(): int
+ getName(): String
+ getWeight(): double

StringReplication

+ execute(Individual child,
List<Individual> parents): void
+ getParentsRequired(): int
+ getName(): String
+ getWeight(): double

RunManager

- loadSave: LoadSave
- popManager: PopulationManager

+ save()
+ load(int gensBack)

PopulationManager

+ createMemento() : PopulationManagerMemento
+ setMemento(PopulationManagerMemento m):void

LoadSave

- archivePath:String

+ LoadSave(String archivePath)
+ load(int gensBack): PopulationManagerMemento m
+ load(File oldPop): PopulationManagerMemento m
+ save(PopulationManagerMemento m) : void

PopulationManagerMemento

- ancestryForest : List<Individuals>
- currentPopulation : List<Individuals>

+ PopulationManagerMemento(List<Individuals>,
List<Individuals>)
+ getAncestryForest() : List<Individuals>
+ getCurrentPopulation : List<Individuals>
- traverse() : List<Individual>

RunManager

hooks: List<RunCommand>

+ earlyTerminatingCondition(): boolean
+ displayCurrentGeneration(): void
+ addHook(RunCommand):void
+ removeHook(RunCommand): void

ConcreteRunManager

// hook methods
+ earlyTerminatingCondition():boolean
+ displayCurrentGeneration(): void

<<Interface>>
RunCommand

+ execute() : void

0..*

Save

+ execute():void

Display

+ execute(): void