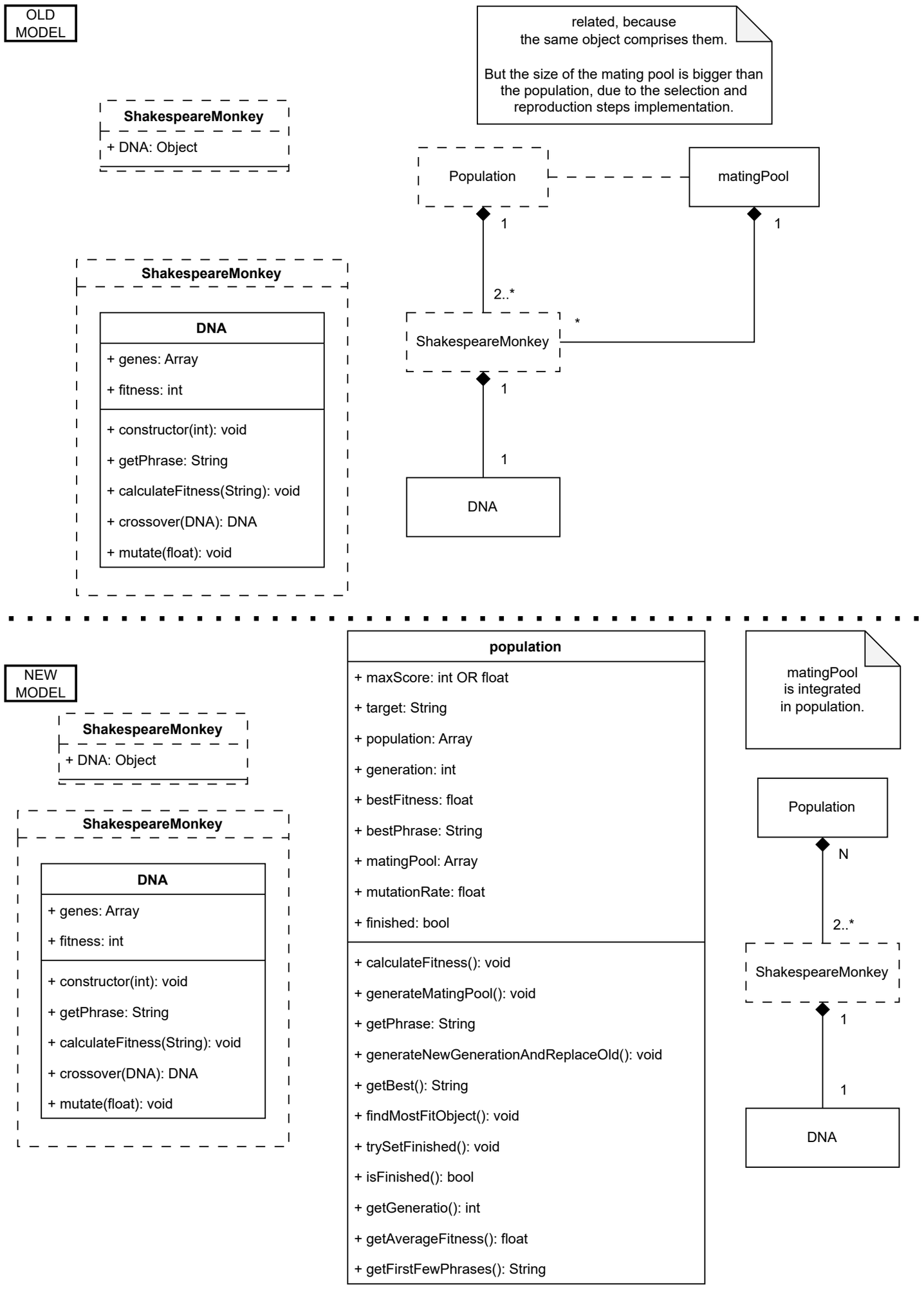
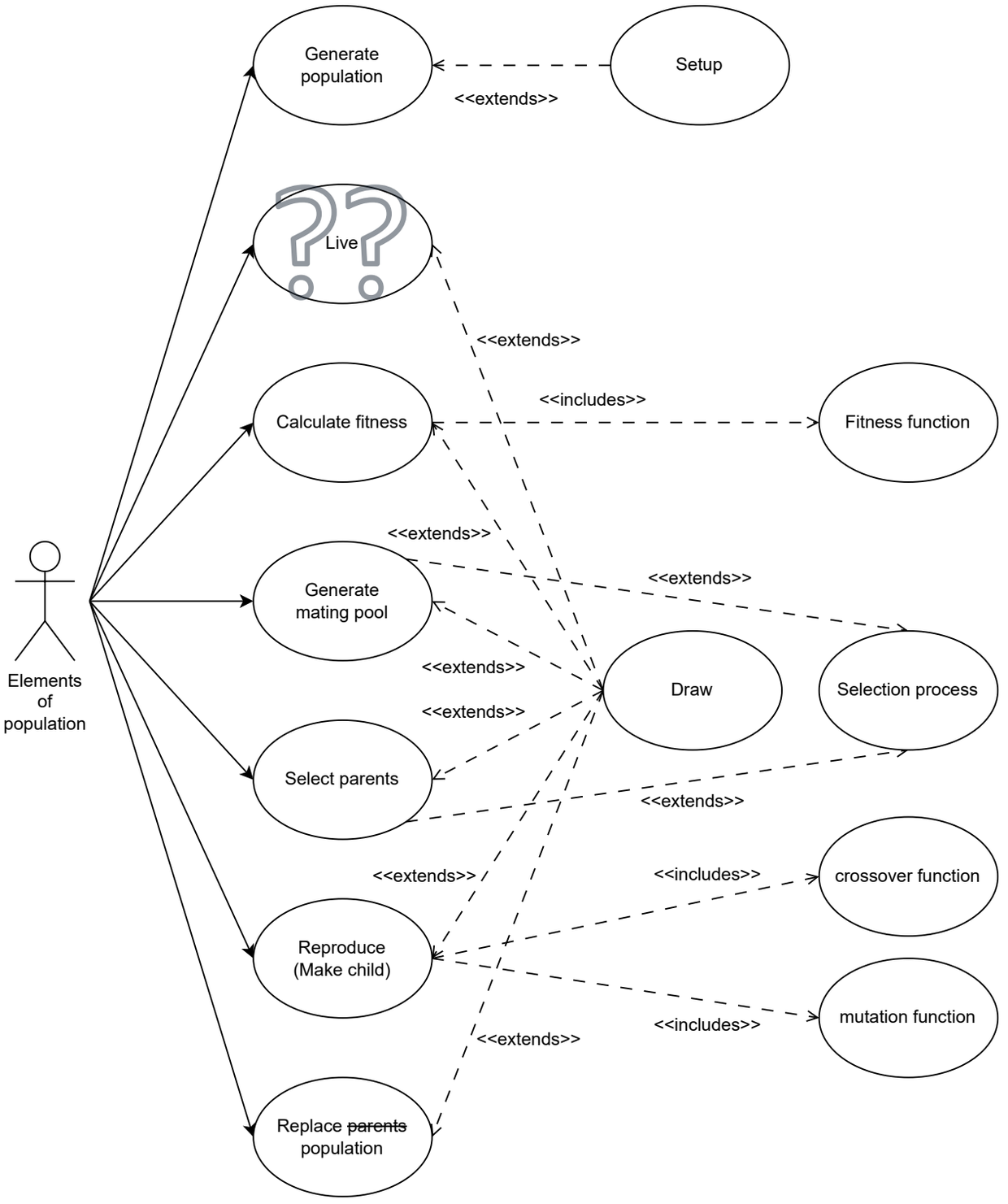
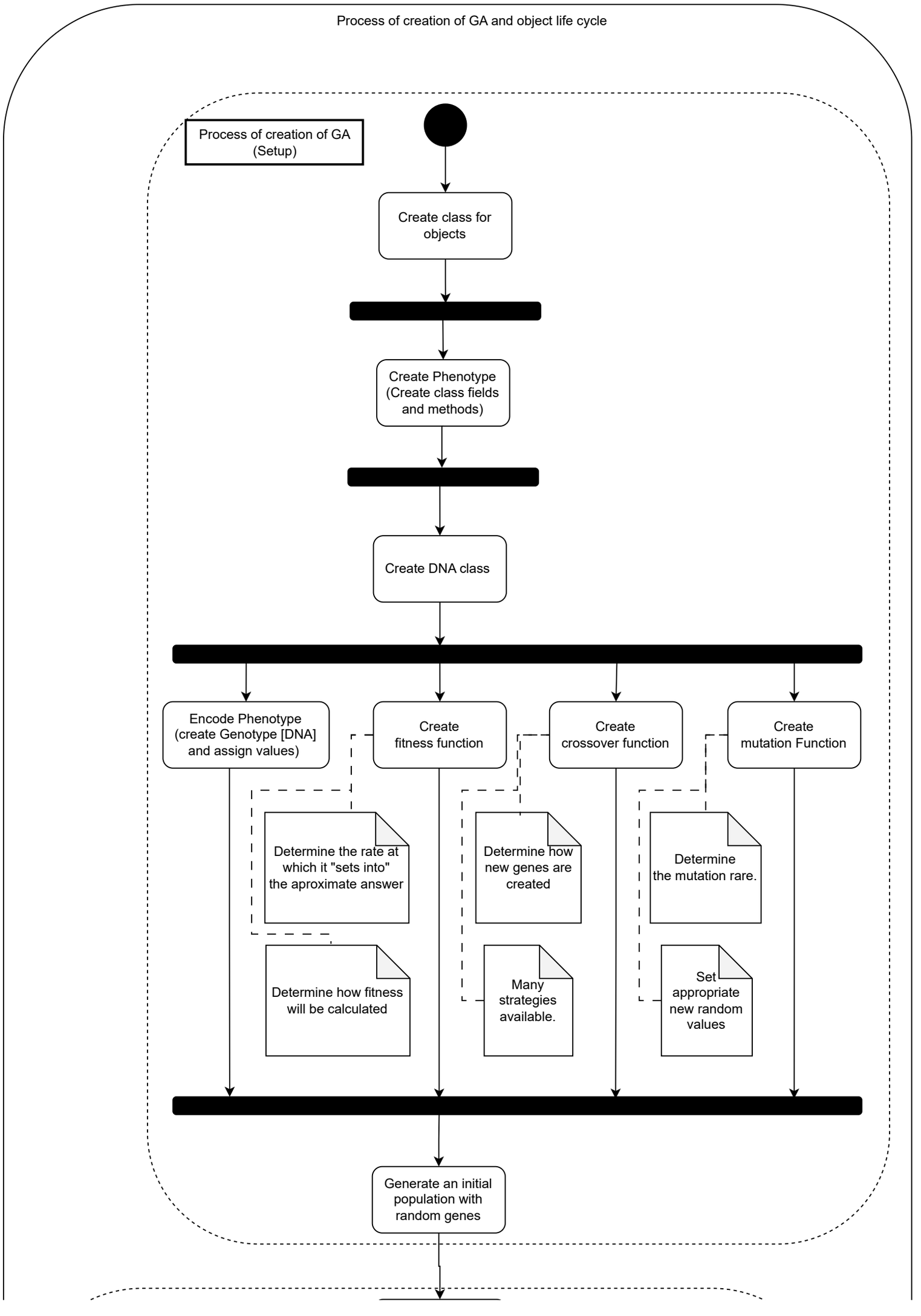
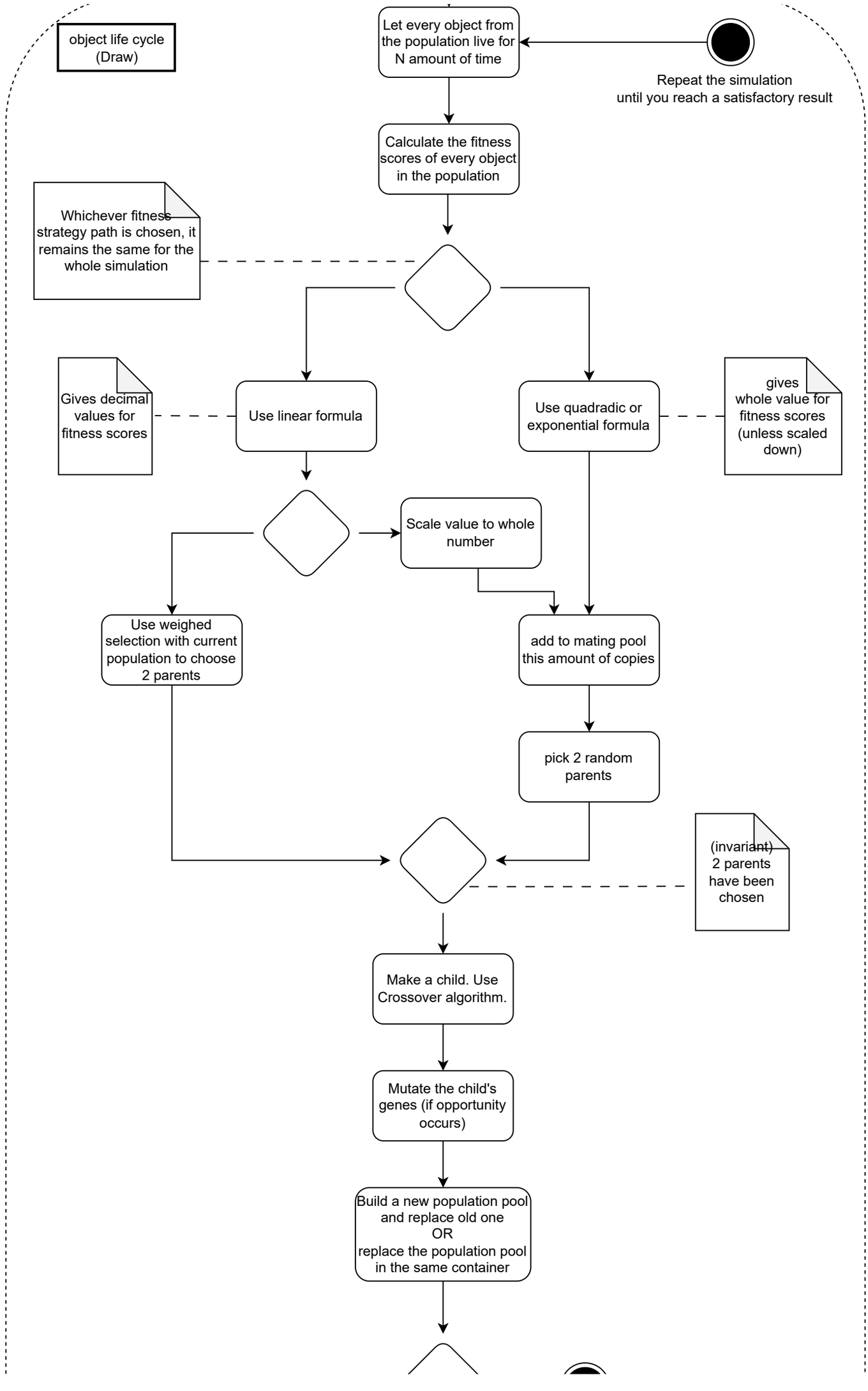
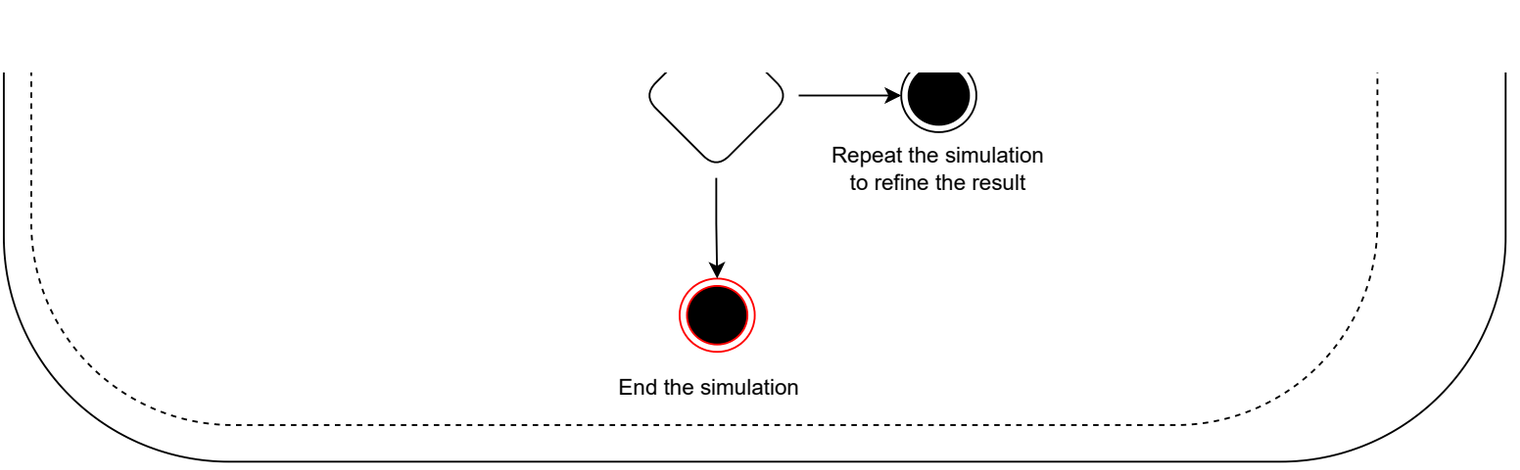

OLD MODEL
related, because the same object comprises them.
But the size of the mating pool is bigger than the population, due to the selection and reproduction steps implementation.
ShakespeareMonkey
+ DNA: Object
Population
matingPool
1
1
2..*
*
ShakespeareMonkey
1
1
DNA
ShakespeareMonkey
DNA
+ genes: Array
+ fitness: int
+ constructor(int): void
+ getPhrase: String
+ calculateFitness(String): void
+ crossover(DNA): DNA
+ mutate(float): void
NEW MODEL
ShakespeareMonkey
+ DNA: Object
ShakespeareMonkey
DNA
+ genes: Array
+ fitness: int
+ constructor(int): void
+ getPhrase: String
+ calculateFitness(String): void
+ crossover(DNA): DNA
+ mutate(float): void
population
+ maxScore: int OR float
+ target: String
+ population: Array
+ generation: int
+ bestFitness: float
+ bestPhrase: String
+ matingPool: Array
+ mutationRate: float
+ finished: bool
+ calculateFitness(): void
+ generateMatingPool(): void
+ getPhrase: String
+ generateNewGenerationAndReplaceOld(): void
+ getBest(): String
+ findMostFitObject(): void
+ trySetFinished(): void
+ isFinished(): bool
+ getGeneratio(): int
+ getAverageFitness(): float
+ getFirstFewPhrases(): String
matingPool is integrated in population.
Population
N
2..*
ShakespeareMonkey
1
1
DNA


Generate population
Setup
<<extends>>
Live
<<extends>>
Calculate fitness
<<includes>>
Fitness function
<<extends>>
Generate mating pool
<<extends>>
<<extends>>
Draw
Selection process
<<extends>>
Select parents
<<extends>>
<<extends>>
<<includes>>
crossover function
Reproduce (Make child)
<<includes>>
mutation function
<<extends>>
Replace parents population
Elements of population


Process of creation of GA and object life cycle
Process of creation of GA (Setup)
Create class for objects
Create Phenotype (Create class fields and methods)
Create DNA class
Encode Phenotype (create Genotype [DNA] and assign values)
Create fitness function
Create crossover function
Create mutation Function
Determine the rate at which it "sets into" the aproximate answer
Determine how fitness will be calculated
Determine how new genes are created
Many strategies available.
Determine the mutation rare.
Set appropriate new random values
Generate an initial population with random genes


object life cycle
(Draw)
Let every object from the population live for N amount of time
Repeat the simulation
until you reach a satisfactory result
Calculate the fitness scores of every object in the population
Whichever fitness strategy path is chosen, it remains the same for the whole simulation
Gives decimal values for fitness scores
Use linear formula
Use quadradic or exponential formula
gives whole value for fitness scores (unless scaled down)
Scale value to whole number
Use weighed selection with current population to choose 2 parents
add to mating pool this amount of copies
pick 2 random parents
(invariant)
2 parents have been chosen
Make a child. Use Crossover algorithm.
Mutate the child's genes (if opportunity occurs)
Build a new population pool and replace old one
OR
replace the population pool in the same container


Repeat the simulation
to refine the result
End the simulation