**Final Report**

To develop the genetic algorithm for solving the ball out of a box problem. The main objective is to come out of the box. To achieve this, the ball can move around, alter its size and reproduce so that if not the original population, at least its generations might come out of the box. The balls might get extinct depending on their size or if they show no traits of improvement to move towards the edge of the box.

We have implemented 5 Classes: Ball class, Population class, Evolution class, Fitness class and Main class. Ball class mainly focus on defining the DNA (Coordinates, Fitness and Size). Population and Evolution class together focus on how the DNA was translated to current coordinates and the mutation process. The Population class mainly contains the whole list of balls and maintains the current co-ordinates, size and fitness of the ball. Evolution class takes care of activities like reproduction and death. Fitness class has methods to evaluate the fitness of each ball and selects the population for evolution. Main class stores the parameters for running the program.

**Implementation Strategy:**

* In the ball class we have the basic genome structure for the ball and where we use a constructor to initialize a ball.
* Constructor of the Population class is used to initialize the first generation of balls with an initial radius of 50 and starting coordinates at central point of the box.
* The evolution class takes care of the movements and mutation processes of the balls based upon the radius of the balls.
* The Fitness class calculates the fittest balls of every generation.
* The getTopKFittest method in Fitness class helps us to find the fittest balls in every generation.
* Post 100 evolutions based upon the return value of the above function the unfit balls are terminated from the existing population to continue the evolution process.

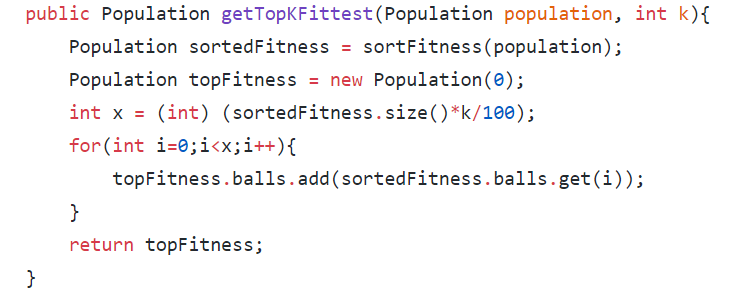
**Conclusion:**

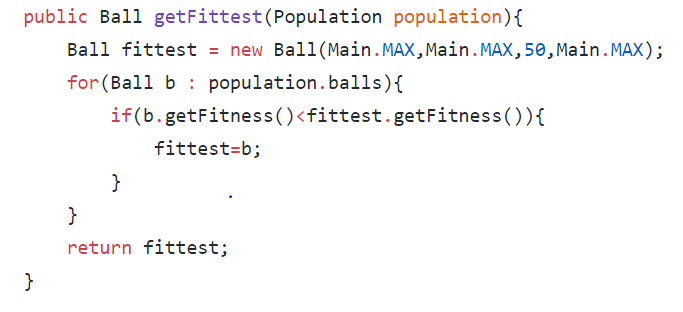
* Our problem is to find a way out of the enclosed box for all the balls present inside it by using genetic algorithm.
* We have considered the following number of balls as initial population size to calculate the effectiveness of our algorithm: (200,500,1000,1500).
* According to our tests higher the initial population greater are the chances of at least 1 ball moving out of the box.
* Over the course of iterations as the generations become fitter due to mutation processes and the chances of a ball getting out of the box are more.

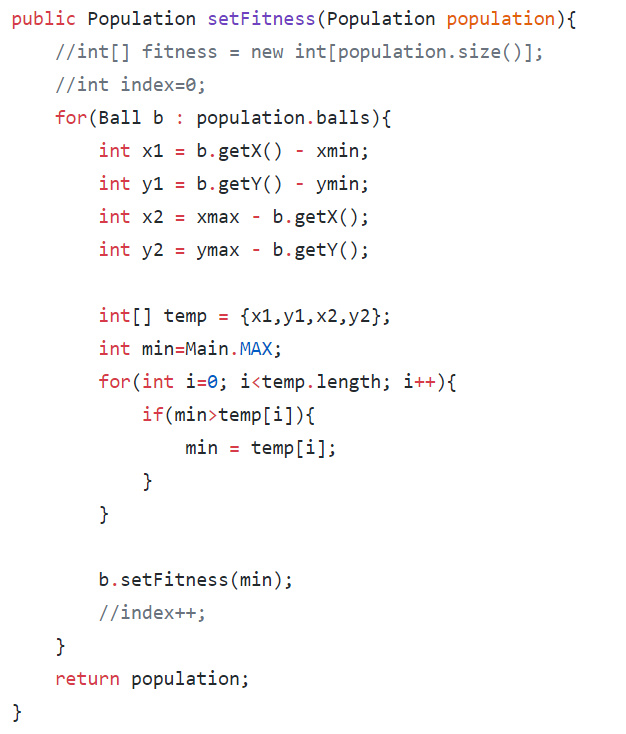
In conclusion, we have implemented genetic algorithm to find out the optimum initial population of balls required to get out of an enclosed box.

**Code Snippets:**

**Fitness Function:**

****

****

****

**Evolution: **

