

TSP with 20 cities of random location

```
package com.company;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Random;

import static com.company.Constants.*;

public class Population {
    public ArrayList<Chromosome> chromosomes = new
    ArrayList<>(POPULATION_SIZE);

    Population() {
        for(int i = 0; i < POPULATION_SIZE; i++){
            chromosomes.add(new Chromosome());
        }
    }

    void generateNew() {

        Collections.sort(chromosomes);

        ArrayList<Chromosome> newChromosomes = new
    ArrayList<>(POPULATION_SIZE);
        int id1 = 0, id2 = 0;
        Random random = new Random();
        ArrayList<Chromosome> parents = new ArrayList<>();

        for(int i = 0; i < POPULATION_SIZE; i++){
            id1 =
    random.nextInt((int) (POPULATION_SIZE*PARENTS_PART));
            id2 =
    random.nextInt((int) (POPULATION_SIZE*PARENTS_PART));
            while(id1 == id2){
                id2 =
    random.nextInt((int) (POPULATION_SIZE*PARENTS_PART));
            }

            newChromosomes.add(new
    Chromosome(chromosomes.get(id1), chromosomes.get(id2)));
        }
        chromosomes = newChromosomes;
    }
}
```

```

double getAverageFitness(){
    double fitness = 0;
    for(Chromosome chromosome : chromosomes){
        fitness += chromosome.getFitness();
    }
    return fitness/POPULATION_SIZE;
}

double getTopFitness(){
    double fitness = Integer.MAX_VALUE;
    for(Chromosome chromosome : chromosomes){
        if(fitness > chromosome.getFitness()){
            fitness = chromosome.getFitness();
        }
    }
    return fitness;
}
}

package com.company;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;

public class Main {

    public static void main(String[] args) {
        Population population = new Population();
        Cities.init();

        File fFitness = new File("fitness.txt");
        File fTopFitness = new File("topFitness.txt");

        try {
            PrintWriter pwFitness = new
PrintWriter(fFitness);
            PrintWriter pwTopFitness = new
PrintWriter(fTopFitness);

pwFitness.println(population.getAverageFitness());

pwTopFitness.println(population.getTopFitness());
            for (int i = 0; i < 200; i++) {
                population.generateNew();
            }
        }
    }
}

```

```

pwFitness.println(population.getAverageFitness());

pwTopFitness.println(population.getTopFitness());
    }
    pwFitness.close();
    pwTopFitness.close();
} catch (FileNotFoundException e) {
    e.printStackTrace();
}
}
}

```

