

## ОТЧЕТ О ПРОДЕЛАННОЙ РАБОТЕ №2

Студент Газзаев Алан- Барсаг Валерьевич 411117, группа М4130

### Описание работы

Целью данной лабораторной работы является получение представления об возможностях применения эволюционных алгоритмов для решения различных классов задач и программных средств для их разработки.

### Оборудование и программное обеспечение

- Java JDK версии 1.8 и выше
- Watchmaker framework 0.7.1 из <https://github.com/dwdyer/watchmaker>

### Bits Count

Было подсчитано сколько итераций (поколений) потребовалось, чтобы найти оптимальное решение для строки длиной 20, 50 и 100.

Размерность	Run 1	Run 2	Run 3	Run 4	Run 5	Среднее
20	11	14	14	11	13	12,6
50	389	455	74	452	334	340,8
100	1973	2632	3228	2899	2835	2713,4

### Traveling salesman problem

Результаты при исходных параметрах:

```
[Evolution (pop: 300, gen: 100, elite: 3, Truncation Selection (50%))]  
ROUTE: Berlin -> Vienna -> Athens -> Rome -> Madrid -> Lisbon -> Dublin ->  
London -> Paris -> Luxembourg -> Brussels -> Amsterdam -> Copenhagen ->  
Stockholm -> Helsinki -> Berlin  
TOTAL DISTANCE: 10494.0km  
(Search Time: 2.741 seconds)
```

Результаты при увеличении популяции до 600

```
[Evolution (pop: 600, gen: 100, elite: 3, Truncation Selection (50%))]
```

```
ROUTE: Rome -> Madrid -> Lisbon -> Dublin -> London -> Paris -> Luxembourg ->
Brussels -> Amsterdam -> Copenhagen -> Stockholm -> Helsinki -> Berlin ->
Vienna -> Athens -> Rome
TOTAL DISTANCE: 10494.0km
(Search Time: 0.417 seconds)
```

Результаты при увеличении числа поколений до 200

```
[Evolution (pop: 300, gen: 200, elite: 3, Truncation Selection (50%))]  
ROUTE: Berlin -> Helsinki -> Stockholm -> Copenhagen -> Amsterdam -> Brussels  
-> Luxembourg -> Paris -> London -> Dublin -> Lisbon -> Madrid -> Rome ->  
Athens -> Vienna -> Berlin  
TOTAL DISTANCE: 10494.0km  
(Search Time: 0.174 seconds)
```

Результаты при увеличении популяции до 600 и числа поколений до 200

```
[Evolution (pop: 600, gen: 200, elite: 3, Truncation Selection (50%))]  
ROUTE: Madrid -> Rome -> Athens -> Vienna -> Berlin -> Helsinki -> Stockholm  
-> Copenhagen -> Amsterdam -> Brussels -> Luxembourg -> Paris -> London ->  
Dublin -> Lisbon -> Madrid  
TOTAL DISTANCE: 10494.0km  
(Search Time: 0.43 seconds)
```

Результаты при увеличении числа элитарных объектов до 12:

```
[Evolution (pop: 300, gen: 200, elite: 12, Truncation Selection (50%))]  
ROUTE: Vienna -> Athens -> Rome -> Madrid -> Lisbon -> Dublin -> London ->  
Paris -> Luxembourg -> Brussels -> Amsterdam -> Copenhagen -> Stockholm ->  
Helsinki -> Berlin -> Vienna  
TOTAL DISTANCE: 10494.0km  
(Search Time: 0.167 seconds)
```

## Mona Lisa

Решение	Итерация	Фитнесс	Кол-во полигонов и углов	Рисунок
плохое	4490	344065	14 п, 86 у	
среднее	22465	213730	30 п, 195 у	
хорошее	27408	209574	50 п, 500 у	

### Ответы на вопросы:

#### 1. Типы задач

1.1. BitsCount. Задача одной функции в дискретном (бинарном) пространстве.

1.2. Traveling Salesman Problem. Задача многокритериальной оптимизации (нужно

посетить каждый город и при этом найти кратчайший путь) в дискретном пространстве (если расстояния между городами заданы заранее).

1.3. Mona Lisa. Задача мультимодальной минимизация одной функции (несколько

комбинаций полигонов дают один результат, т. е. функция имеет несколько оптимумов) в непрерывном пространстве координат углов полигонов.

2. Списки строк кандидатов List<String> candidate и расстояние totalDistance полученное

после применения фитнес-функции getFitness.

3. Генотип - список полигонов `List<ColouredPolygon>`, фенотип -  
отрендеренное  
растровое изображение.