

**Gebze Technical University  
Computer Engineering**

**CSE 443 - 2019 Autumn**

**HOMEWORK 2 REPORT**

**AKIN ÇAM  
151044007**

Course Assistant:  
**Ercan Aptoula**

- PART1

Not: İntellij İdea ortamında geliştirilmiştir.

Bu bölümde Generic Algoritma yöntemleri kullanılarak verilen bir fonksiyonun maksimum değeri optimize edilmiştir. Bu bölüm gerçekleştirirken template tasarım deseni kullanılmıştır. Template tasarım deseni davranışsal tasarım deseni grubunda yer alır. Bu tasarım deseni bir algoritmanın iskeletini bir yöntemde tanımlayarak alt sınıflara bazı adımları gerçeklemesine olanak sağlar. Template tasarım deseni, alt sınıfların algoritmanın yapısını değiştirmeden algoritmanın belirli adımlarını yeniden tanımlamasını sağlar. Bu sebeple Template tasarım deseni kullanılmış ve şöyle gerçekleştirilmiştir:

- GeneticAlgorithm adında bir soyut sınıf oluşturulmuştur. Bu sınıfta algoritma yöntemi tanımlanmıştır.
- Bu sınıfta geneticAlgorithmSolver ile algoritma iskeleti gerçekleştirilmiştir
- Bu method final olarak gerçekleştirilmiş ve diğer sınıflarla değiştirilmesi önlenmiştir:  
*\* initialize population with given population size*  
*\* calculate fitness each population member*  
*\* To show clearly Thread.sleep added*  
*\* Sequentially:*  
*\* Loop{*  
*\* Selection -- abstract method to change other versions*  
*\* Crossover -- abstract method to change other versions*  
*\* Mutation*  
*\* ComputeFitness*  
*\* calculateMaxFit - to find optimal solution*  
*\*}*
- Selection ve Crossover methodları algoritmanın iskelet yapısını değiştirmeden alt sınıfların gerçekleyebilmesi için soyut olarak oluşturulmuştur.
- Mutasyon InitializePopulation ve Calculate Metodları gerçekleştirilmiştir.

---

Individual Sınıfı :

- Her bir populasyon üyesini gerçeklemeye sağlayan methoddur.
- fitness ve populasyon üyesini binary olarak tutmaktadır.
- her üye 0 ile 5 arasında ve toplamı 0 ile 5 arasında olacak şekilde oluşturulur.

---

GeneticAlgorithm Sınıfı:

- Bu sınıfta ilk olarak kodu iki kere yazmayı önlemek için crossoverHelper methodu yazılmış ve code duplicate önlenmiştir.

- Crossoverda bir populyasyondaki üyeler ikili olarak eşlenir. Random bir sayı belirlenir ve o sayıdan önceki gen dizilimi değişmezken o sayıdan sonraki gen dizilimini gen diğer eşinden alır.
- Mutasyonda random seçilen bitler değiştirilmektedir.
- Fittest değişkeninde o ana kadar olan iterasyondaki max fitness değeri bulunmaktadır.

---

#### GeneticAlgorithmVersion1 Sınıfı:

- RouletteWheel methodu kullanılarak selection yapılmıştır.
- Bu methodda totalFitness değeri ile 0 arasında bir değer alınır random olarak. Daha sonra tıpkı bir çark gibi her üyenin fitness değeri toplanarak gidilir ve random değeri geçince o üye yeni listeye eklenir ve bu liste döndürülür.(Populasyon sayısı dolana kadar devam eder.)
- Daha sonra crossover da çeşitlilik için populasyon listesi karıştırılır.
- Crossoverda bir populyasyondaki üyeler ikili olarak eşlenir. Random bir sayı belirlenir ve o sayıdan önceki gen dizilimi değişmezken o sayıdan sonraki gen dizilimini gen diğer eşinden alır.

---

#### GeneticAlgorithmVersion2 Sınıfı:

- Rank selection methodu kullanılarak selection yapılmıştır.
- Bu methodda fitness değerlerine göre populyasyondakiler sıralanır.
- En küçük populasyon yapısının fitness değerine 0 dan başlayarak 1..2..3 şeklinde değiştirilir. Daha sonra toplam fitness değeri hesaplanır ve roulette wheel selection yapılır.
- Crossoverda bir populyasyondaki üyeler ikili olarak eşlenir. Random iki sayı belirlenir ve o 2 sayı arasındaki gen dizilimi değişirken diğer gen dizilimleri değişmez.

---

#### GeneticAlgorithmVersion3 Sınıfı:

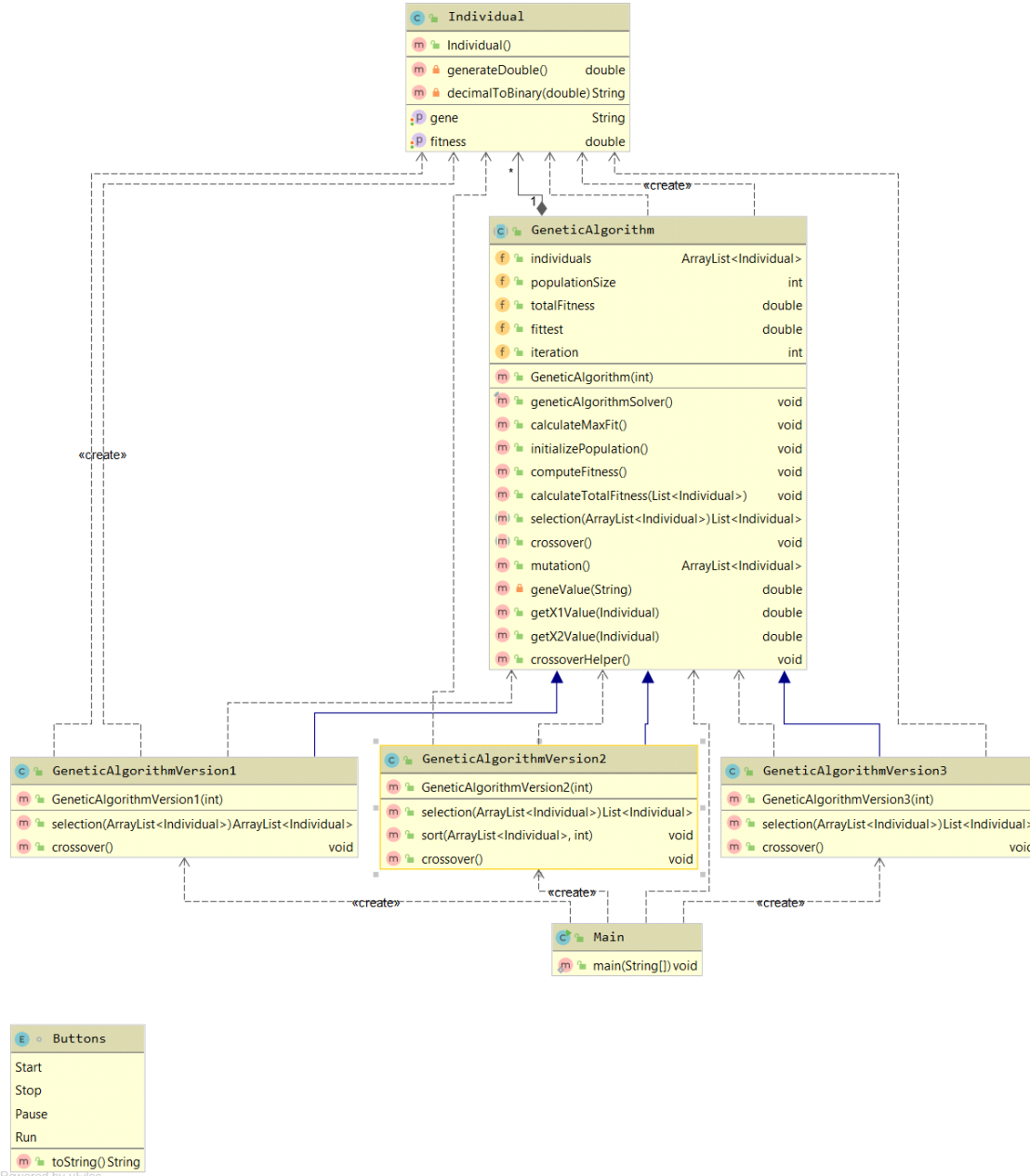
- Tournament Selection methodu kullanılarak selection yapılmıştır.
- Bu methodda bir turnuva uygulanır. Random bir sayı belirlenir ve o sayı ile alınan elemanlar arasında bir turnuva yapılır ve en iyisi seçilir. Populasyon sayısı tamamlanana kadar bu işlem devam eder.
- Crossoverda bir populyasyondaki üyeler ikili olarak eşlenir. Random bir sayı belirlenir ve o sayıdan önceki gen dizilimi değişmezken o sayıdan sonraki gen dizilimini gen diğer eşinden alır.

---

Test sonuçlarında maxfittest değeri en çok 126.04 olarak gözlemlenmiştir.

---

Sınıf Diyagramı:



Sınıf diyagramında görüldüğü gibi GeneticAlgorithm Sınıfı abstract olarak gerçekleştirilmiş. İçinde bulunan crossover ve selection metodları soyut olarak gerçekleştirilmiştir. Bu ana sınıftan extend olan ve ana algoritmanın iskeletini değiştirmeden algoritmanın belirli adımlarını değiştirmeyi sağlayan 3 sınıf gerçeklemiştir. Bu sınıflarda soyut sınıflar gerçekleştirilmiştir.

Test Sonuçları :

```

C:\Users\akinc>java -jar D:\151044007_HW2_PART1\out\artifacts\151044007_HW2_PART1_jar\151044007_HW2_PART1.jar
Version 1----->

Fittest value: 103.60382334074264
Fittest value: 103.60382336242503
Fittest value: 109.91035895439879
Fittest value: 111.60291224731655
Fittest value: 118.4081872666529
Fittest value: 124.81810605824343
Fittest value: 124.85497397524595
Fittest value: 126.02708473795336
Fittest value: 126.04024112309763
-----
MAX FITTEST: 126.04024112309763
-----

Version 2----->

Fittest value: 114.45782996447282
-----
MAX FITTEST: 114.45782996447282
-----

Version 3----->

Fittest value: 99.22883214492465
Fittest value: 115.41235039320641
Fittest value: 115.42645653733481
-----
MAX FITTEST: 115.42645653733481
-----

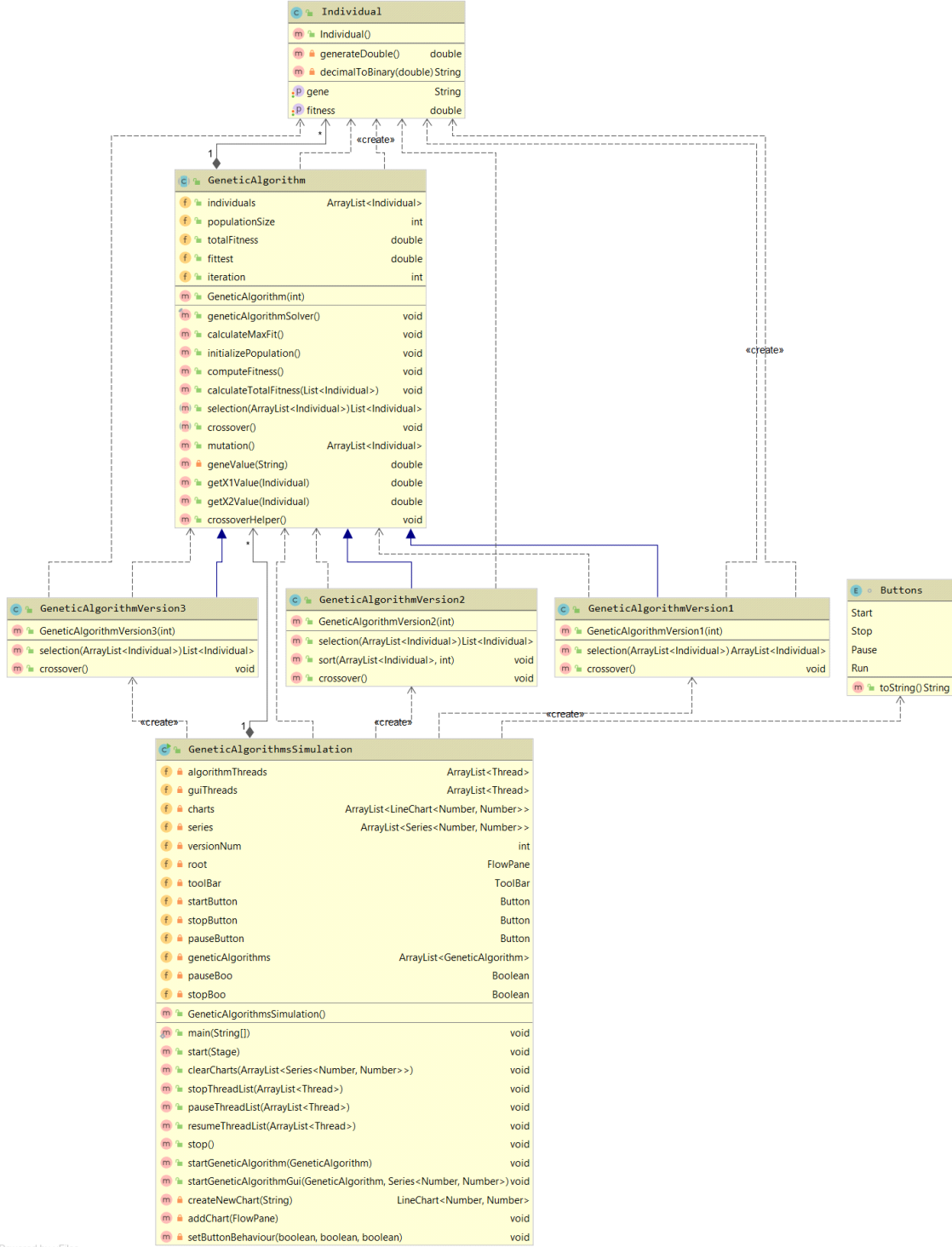
```

## • PART2

Bu bölümde Generic Algoritma yöntemlerinin grafik haline getirilmesi sağlanmıştır. Bu yapı için java fx kullanılmıştır.

3 Genetic Algoritma için 3 thread ve 3 grafik için 3 thread olmak üzere toplamda 6 thread kullanılmıştır. Ana char için Line chart, bu yapıyı gerçek zamanlı verilerle işlenmesi içi XYChart.Series<Number, Number> kullanılmıştır. 3 grafik yapısı ana Stage adlı ana konteyner a eklendikten sonra 3 genetic algoritma listeye eklenmiştir. Start Stop ve pause olmak üzere 3 buton bulunmaktadır. Start butonuyla birlikte her method çalışmaya başlar grafikler oluşturulur. Metodlarda 1 sn delay eklenmiştir adım adım gözlemlemek için. Arayüz değişiklikleri Ana thread de değil yeni bir threadde ve küçük bir task olduğu için Platform.runLater() içinde gerçekleştirilmiştir. Bu threadler bir listeye kaydedilmiştir. Pause durumunda threadler suspend olarak set edilir. Run durumunda threadler resume olarak set edilir. Stop durumunda tüm threadler durdurulur. Arayüz kısmında sonsuz döngüde bulunmasını sağlayan değişken değiştirilir ve Platform.runLater() dan çıkıp thread şın sonlanması sağlanır. Start ile yeniden oluşturulan gerçek zamanlı grafikler oluşturulur ve threadler listeye eklenir.

Sınıf Diyagramı:



Genetic Algoritma için template tasarım deseni kullanılmıştır.  
GeneticAlgorithmSimulationSınıfı bu algoritmaları kullanmaktadır. Bunlar üye olarak

bulunmaktadır. Herhangi bir deęişiklik olduęunda bu yapıları alıřtıran threadlere durum bildirilmektedir(Observer). Button enum sınıfı buton isimlerini tutmaktadır.

## Test Sonuları:

Genetic Algorithm Chart Animation

