

VERİ YAPILARI ÖDEV 1 RAPORU

ÖDEVİN AÇIKLAMASI: Ödev de bizden istenen bağlı liste kullanarak gen ,kromozom ve DNA işlemlerini yapmak.DNA.txt ve Islemler.txt dosyasını okumak.

ÖDEVİN YAPILIŞI: Ödev iki yönlü dairesel bağlı liste formatına uygun olduğunu düşündüğüm için düğümü mü bu şekilde yaptım.

Dugum.hpp kodum oluşturdum

```
#ifndef DUGUM_HPP
#define DUGUM_HPP

class Dugum {
public:
    char veri; // Genetik bilgi (karakter)
    Dugum* sonraki; // Sonraki düğüme işaretçi
    Dugum* onceki; // Önceki düğüme işaretçi

    Dugum(char veri); // Yapıcı: Düğümün verisini alır ve sonraki ve önceki düğümleri nullptr olarak başlatır
};

#endif
```

DNA.hpp kodum **Yanlarına yorum satırı olarak kısaca ne işe yaradıklarını yazdım.**

```
✓ #ifndef DNA_HPP
#define DNA_HPP

✓ #include "Liste.hpp"
#include <string> // std::string kullanımı için gerekli

✓ class DNA {
private:
    struct KromozomDugum { // DNA'yı oluşturan bağlı liste düğümü
        Liste* kromozom; // Bir kromozomu temsil eden Liste
        KromozomDugum* sonraki; // Sonraki düğüme işaretçi
        KromozomDugum(Liste* kromozom) : kromozom(kromozom), sonraki(nullptr) {}
    };

    KromozomDugum* ilk; // DNA'daki ilk kromozom düğümüne işaretçi

public:
    DNA(); // DNA yapısını başlatır (ilk düğüm nullptr olarak ayarlanır)
    ~DNA(); // Tüm düğümleri ve içlerindeki veriyi (Liste) silerek belleği temizler
    void kromozomEkle(Liste* kromozom); // Yeni bir kromozomu DNA'ya ekler
    void caprazlama(int indeks1, int indeks2); // İki kromozom arasında çaprazlama yapar
    void mutasyon(int kromozomIndeks, int genIndeks); // Belirtilen kromozomdaki bir gende mutasyon yapar
    void yazdir(); // ekrana yaz işlemi için ilgili kromozomun en küçük genini ekrana yazdırır
    // Otomatik işlemler fonksiyonu
    void otomatikIslemler(const std::string& islemlerDosyasi); // Islemler.txt dosyasını okuyarak işlemleri uygular
    Liste* kromozomGetir(int indeks); // Belirtilen sıradaki kromozomu döndürür
};

#endif
```

Ve Liste.hpp

```
#ifndef LISTE_HPP
#define LISTE_HPP

#include "Dugum.hpp"

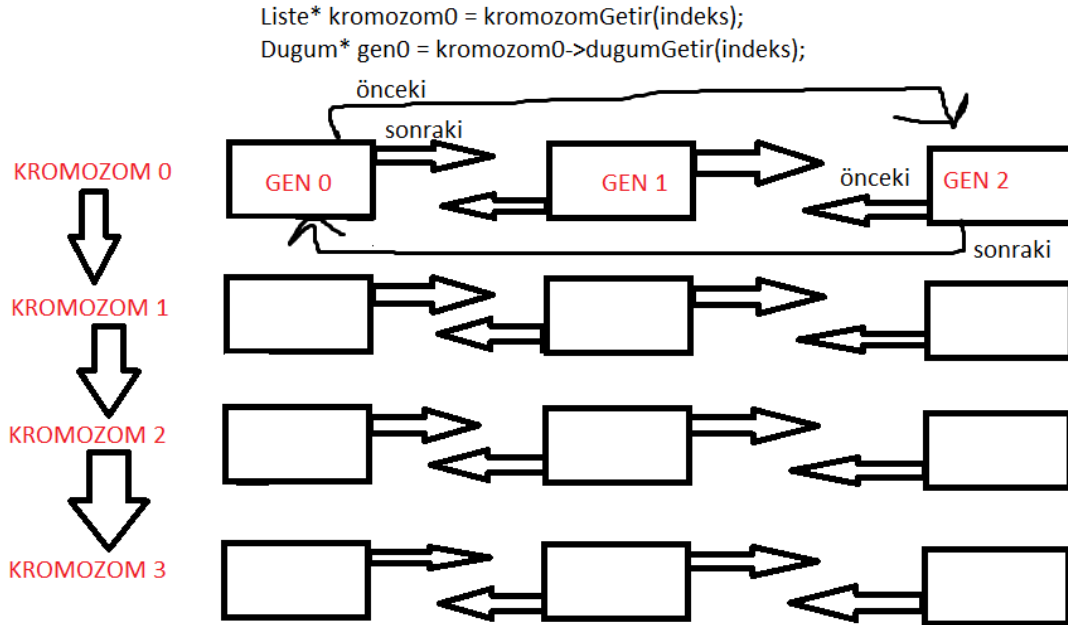
class Liste {
private:
    Dugum* ilk; // Listenin ilk düğümüne işaretçi

public:
    Liste(); // Yapıcı: Bağlı listeyi başlatır (ilk düğümü nullptr olarak ayarlar)
    ~Liste(); // Yıkıcı: Tüm düğümleri serbest bırakarak belleği temizler
    void ekle(char veri); // Yeni bir gen (düğüm) ekler (listenin sonuna ekleme yapar)
    Dugum* dugumGetir(int sıra); // Belirtilen sıradaki düğümü döndürür
    void yazdir(); // Listenin en küçük (genini) bulup ekrana yazdırır
    int uzunluk(); // caprazlama işleminde kromozomun uzunluğunu bulma ve orta noktasını bulma
};

#endif
```

Ödevimin modellenmesini yapacak olursak bu şekilde düşünebiliriz.

KROMOZOM DÜĞÜMLERİ TEK YÖNLÜ—GEN DÜĞÜMLERİ İKİ YÖNLÜ DAİRESEL BAĞLI LİSTE MODELİNE UYGUN YAPILMIŞTIR



Bizden istenen menü tasarımı ve işlemlerin hepsi istenildiği gibi yapılmıştır.

- 1 - Çaprazlama = seçilen iki kromozomun orta noktasından bölerek çaprazlama yapma.(PDF te istenilen yöntemden yapılmıştır).
- 2 - Mutasyon = seçili kromozomun genine X yazarak mutasyona uğratma
- 3 - Otomatik İşlemler = İşlemler.txt dosyasını okuyarak hızlı çaprazlama ve mutasyon yapma.
- 4 - Ekrana Yaz = tüm kromozomların en küçük genini ekrana yazma(PDF te istenilen yöntemden yapılmıştır).
- 5 - Çıkış

ÖZETLE: iki yönlü dairesel bağlı liste VE tek yönlü bağlı liste örneği model alınmış, PDF’te istenilen tüm iş kurallarına uygun şekilde yapılmıştır.

HAZIRLAYAN: BATUHAN YILDIRIM - B221210049 - 1.ÖĞRETİM A GRUBU