REPORT

Bu programda virtual ve pure virtual fonksiyonlarımı aşağıdaki gibi tanımladım :

virtual List<T>& addElement(T element);

virtual List<T>& removeElement(T element) = 0;

int numberOfElement () const;

T at(int index) const;

virtual List<T>& operator +=(const List<T>& otherList) ;

virtual List<T>& operator +(const List<T>& otherList) = 0;

virtual List<T>& operator %(const List<T>& otherList) = 0;

virtual int searchElement(T element) = 0;

virtual List<T>& addElement(T element) : Bu fonksiyonu virtual yapmamın nedeni bir elemanı bir listeye normal olarak sondan ekleyebiliriz.Onun için abstract yapmadım.Fakat derived classlarda bu fonksiyon farklı çalışır.SortedList classında bir sıralama söz konusu iken,uniqueList classında bir elemanı 2 kere almamama söz konusu.Bu yüzden addElement override edilmeli.

virtual int searchElement(T element) = 0 : SortedList’te binary search varken , uniqueListte normal bir search vardır.Bu iki classın implementasyonları farklı olacağından bu fonksiyonun da base class içerisinde implement edilmesine gerek yoktur.

virtual List<T>& operator +=(const List<T>& otherList) : Bu fonksiyon virtualdır çünkü içerisinde kullanılan addElement fonksiyonu her iki class için de farklıdır.Fakat compiler late binding yapacağı için tüm classlar için doğru çalışır.O yüzden abstract olmasına gerek yoktur.

virtual List<T>& removeElement(T element) = 0 : Bu fonksiyonu abstract yapmamın nedeni ise bir elemanı kaldırırken ilk olarak o elemanın var olup olmadığı kontrol edilir.İki derived classta da searchElement implemantasyonları farklı olduğu için bu classın base classta implement edilmesine gerek yoktur.

virtual List<T>& operator +(const List<T>& otherList) = 0; : Bu fonksiyonu abstract yapmamın nedeni List classının abstract oluşu ve objesinin yapılamaması.Bu yüzden bu fonksiyon kullanılamaz.Ancak derived classlarında implement edilirse kullanılabilir.

virtual List<T>& operator %(const List<T>& otherList) = 0; : Bu fonksiyonda aynı + operatöründe olduğu gibi,List’in objesi oluşturulamayacağından Listte implement edilmesi gereksizdir.Bu yüzden abstract yaptım.Aynı şekilde derived classlarında implement edilirse kullanılabilir.

int numberOfElement () const;

T at(int index) const;

Yukarıdaki iki fonksiyon da her halde tüm derived classlarında doğru çalışacağından doğrudan implementasyonunu base classta yaptım.