

CS-TECH ASSESSMENT KILAVUZU

Yazılım detayları;

Yazılım mimarisi Diagram 1’de görselleştirilmiştir.

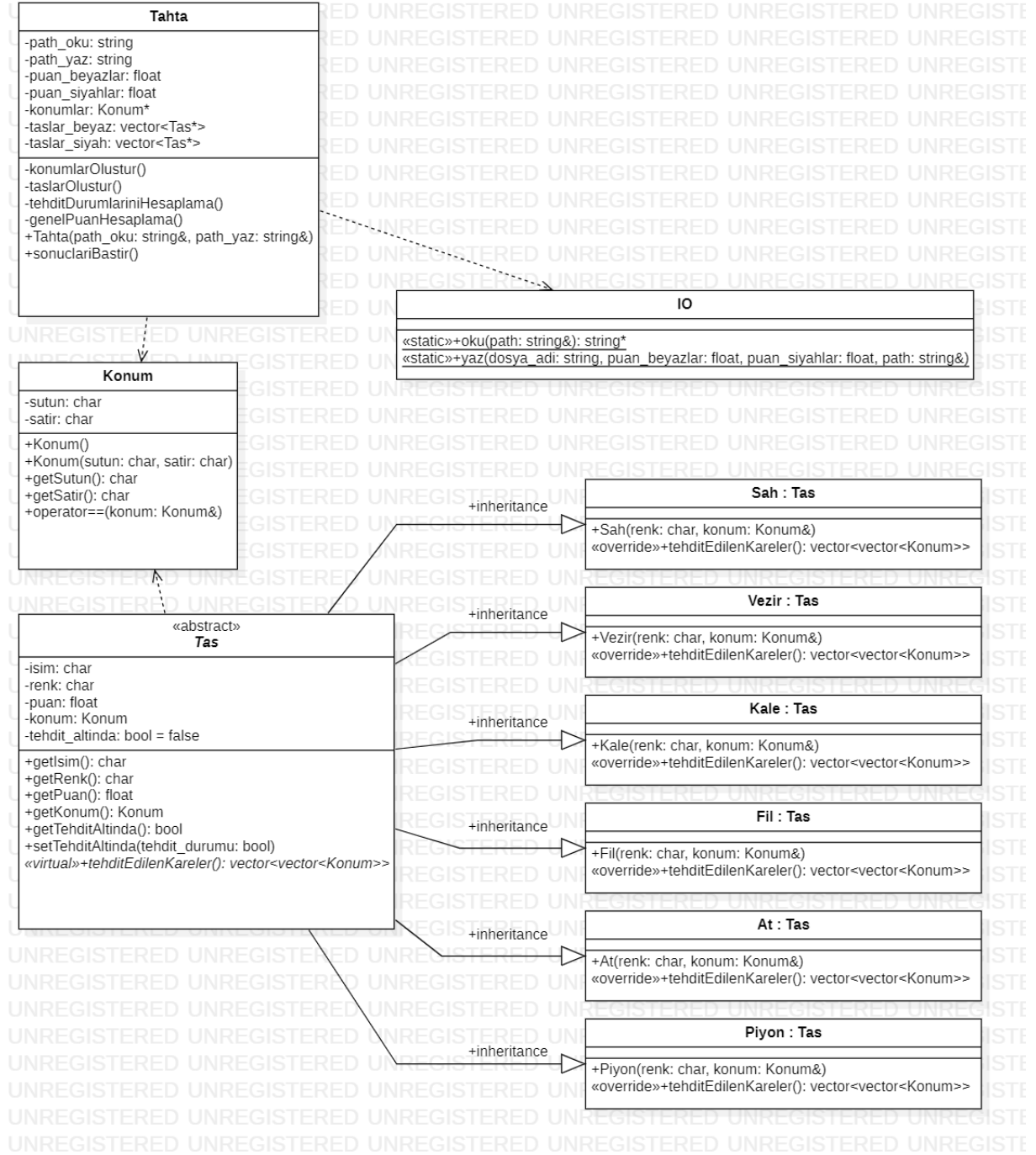


Diagram 1

Sınıflar;

IO sınıfı: Nesne oluşturmada dosyadan okuma ve dosyadan yazma işlemlerini gerçekleştiren iki adet statik fonksiyon barındırır.

Konum sınıfı: Satranç tahtasındaki kareleri temsil eden nesneler oluşturur. İki adet özelliği vardır, sütun ve satır. Yazdırma vb. işlemlerin daha kolay yapılabilmesi için sütun ve satır özellikleri "char" veri tipi olarak depolanmaktadır.

Tas sınıfı: Soyut taş sınıfı. Kendisinden kalıtım yolu ile oluşturulan Sah, Vezir, Kale, Fil, At ve Piyon sınıflarının temel özelliklerini ve fonksiyonlarını barındırır.

Tas sınıfları: "tehditEdilenKareler" fonksiyonu hariç bütün özellikleri aynı olan 6 adet taş sınıfı. Taşların her birinin tehdit ettikleri kareleri hesaplama algoritması farklılıklar içerdiği için "tehditEdilenKareler" fonksiyonu her sınıfta override edilmektedir.

Tahta sınıfı: Main.cpp içerisinde Tahta sınıfından bir nesne oluşturulur. Konumların oluşturulması, taşların oluşturulması ve konumlara atanması, taşların tehdit ettikleri karelerin hesaplanması ve genel puan hesaplaması, Tahta sınıfının kurucu fonksiyonu içerisinde gerçekleşir. İşlemler bittikten sonra Main.cpp içerisinde "sonuclarıBastir" fonksiyonu kullanılarak sonuçlar konsol ekranına detaylı bir biçimde basılır aynı zamanda "Applicant Assessment Test_4" dokümanı içerisinde istenilen formatta bir txt dosyasına yazılır.

Yaklaşım;

Satranç tahtasında taşların tehdit edilme durumlarına göre puan hesabı yapılabilmesi için, her bir tas türünden nesneler oluşturulabilecek sınıflar hazırlanmalıdır. Tas türlerinin birçok ortak noktası olması sebebi ile Tas sınıfı oluşturulmuştur. Taş adında bir taş türü olamayacağı için Taş sınıfı soyut sınıf olarak oluşturulmuştur. Taş sınıfından kalıtım yolu ile Sah, Vezir, Kale, Fil, At ve Piyon sınıfları oluşturulmuştur. Taşlar oluşturulurken kurucu fonksiyon içerisinde Konum sınıfından bir nesne referans verilerek taşlar konumlandırılır. Konumlandırma işlemi Tahta sınıfının kurucu fonksiyonu içerisinde gerçekleştirilir. Tahta sınıfı içerisinde öncelikle tahta üzerindeki 64 karenin hepsi Konum cinsinden birer nesne olarak oluşturulur. Ardından oluşturulan konumlar, taşlar oluşturulurken bulundukları karelere göre referans olarak taşlara verilir. Böylece tahta üzerindeki her bir taşın özellikleri ve bulundukları konumlar hazırlanmış olur.

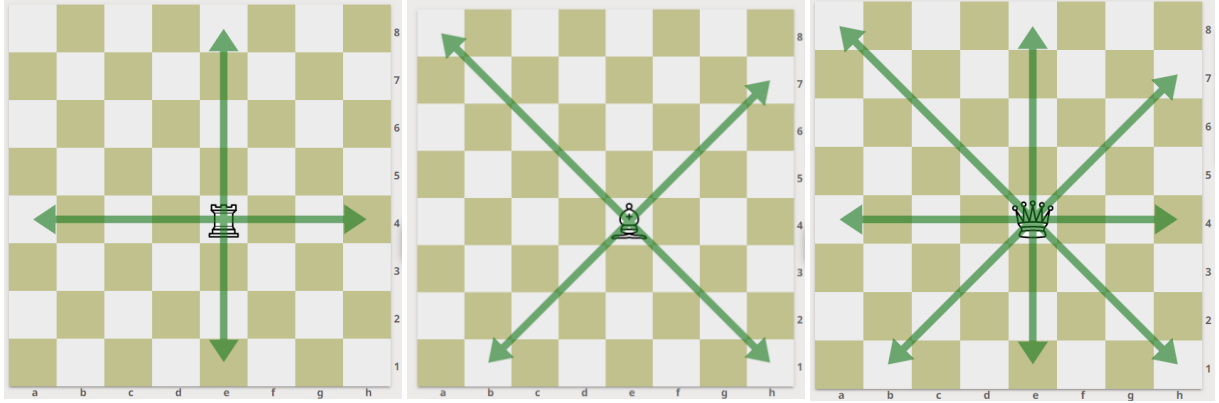
İki farklı vektör içerisinde tutulan beyaz ve siyah taşların her birinin tehdit ettikleri konumlar taşların kendi sınıfları içerisinde bulunan "tehditEdilenKareler" fonksiyonu ile bulunur. Tehdit altında olan konumlar rakip rengin taşlarının konumları ile karşılaştırılır bir eşleşme olması anında rakip taşın "tehdit_altında" özelliği "true" atanır.

Puanların hesaplanması aşamasında ise döngüler yardımı ile iki renginde bütün taşlarının puanları "getPuan" fonksiyonu ile alınır ve genel puan ile toplanır. "tehdit_altında=true" olan taşların "getPuan" fonksiyonları orijinal puanlarının yarısını döndürür. Bu sayede birden fazla taşın aynı taşı tehdit etmesi durumunda taş puanı sürekli ikiye bölünmez.

Tehdit Edilen Kareleri Bulma;

Tehdit edilen kareler bulunurken taşlar iki farklı grupta ele alınmıştır. Vezir, Kale, Fil grubu ve Sah, At, Piyon grubu.

Vezir, Kale, Fil grubunun tehdit ettiği kareler hesaplanırken bu taşlar yatayda, dikeyde ve/veya çaprazlarda tahta sınırlarına kadar tehdit edebilecekleri için döngüler kullanılmıştır. Bu grupta dikkat edilmesi gereken en önemli nokta ise hareket yolları üzerinde bulunan kendi renginden veya rakip renkten bir taşın yolu kesmesi ve o taşın sonrasının hesaba katılmaması gerektiğidir. Engel durumlarının hesaplanabilmesi için bu gruptaki taşların kendi sınıfları içerisindeki “tehditEdilenKareler” fonksiyonları, tahta boş gibi hareket edip bütün tehdit edilebilecek kareler alınır. Bu kareler Kale ve Fil için 4 farklı yönde, Vezir için 8 farklı yönde ilerlemektedir.



Bu yönler üzerindeki bütün kareler “tehditEdilenKareler” fonksiyonu ile Tahta sınıfı içerisindeki “tehditDurumlariniHesaplama” fonksiyonu içerisine gönderilir. Bu fonksiyon içerisinde, bütün taşların konumları alınır ve bir döngü yardımı ile yollardan her biri üzerinde ilerlenir. İlerleme işlemi sırasında bir taşla rastlandığında bu taş kendi renginden ise döngü sonlanır ve bir sonraki yönün vektörüne geçilir. Eğer rastlanan taş rakip renkten bir taş ise bu taşın “tehdit_altında” özelliği “true” atanır ve döngü sonlandırılır. Yönler ait vektörlerin döngüleri engel ile karşılaşıldığında sonlandırıldığı için engelden sonraki konumlar hesaba katılmamış olur.

Derleme;

Taşların okunacağı ve sonuçların yazdırılacağı txt dosyalarının yolları programa 2 farklı şekilde verilebilir.

1. Konsol üzerinde çalıştırılırken “command line argument” string cinsinden 2 farklı dosya yolu absolute path olarak verilebilir. “C:*...*\folder\board*x*.txt”, “C:*...*\folder\sonuclar.txt”. Hata ihtimali gözetilerek dosya yolunda Türkçe karakter kullanılmaması önerilir.
2. Main.cpp içerisinde okunacak dosyanın yolu ve yazılacak dosyanın yolu sırasıyla “path1” ve “path2” olarak verilmiştir. Bu string değişkenlerine istenilen absolute path verilerek program derlenebilir.

Not: “Resources” klasörü altında hazır board txt dosyaları bulunmaktadır.

Araçlar vb.;

Program JetBrains CLion IDE’si ile geliştirilmiştir. CLion derleme işlemi için CMake kullanılmaktadır.